

Universidade do Minho
Escola de Engenharia

Pedro Nuno Lopes Fernandes

Feedback-Error Learning Control for Powered Assistive Devices

Dissertação de Mestrado

**Mestrado Integrado em Engenharia Eletrónica
Industrial de Computadores**

Trabalho efetuado sob a orientação de

Professora Doutora Cristina P. Santos, Universidade do
Minho

Doutor Juan C. Moreno, Consejo Superior de
Investigaciones Científicas

Janeiro de 2019

DECLARAÇÃO

Nome: Pedro Nuno Lopes Fernandes

Endereço eletrónico: pnfernandes@gmail.com

Título da dissertação: “Feedback-Error Learning Control for Powered Assistive Devices”

Orientadores:

Professora Doutora Cristina Santos

Doutor Juan Moreno

Ano de conclusão: 2019

Mestrado integrado em Engenharia Eletrónica Industrial de Computadores

Área de Especialização: Automação, Controlo e Robótica

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA DISSERTAÇÃO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE.

Universidade do Minho, ____/____/____

Assinatura: _____

RESUMO

Patologias da marcha podem conduzir ao desenvolvimento de uma marcha anormal, afetando a mobilidade das pessoas. Dispositivos ativos de assistência (DAA) começam a complementar a reabilitação da marcha. Particularmente, exosqueletos ou ortóteses ativas para os membros inferiores, destacam-se na área da reabilitação da marcha. O sistema de controlo motor humano tem sido usado como inspiração para o design de arquiteturas de controlo para estes DAA, pois compreendem a definição de diferentes níveis de controladores (alto, médio e baixo) organizados hierarquicamente. Especificamente, os controladores de baixo nível têm um papel importante nesta arquitetura, devendo garantir uma assistência temporalmente eficaz adaptada às necessidades do utilizador do utilizador, como a velocidade e a trajetória da marcha.

O objetivo desta dissertação é o desenvolvimento do controlo de baixo nível *Feedback-Error Learning* (FEL) em tempo real, inserido no sistema de controlo bioinspirado SmartOs. O controlo FEL foi realizado através de redes neuronais artificiais (RNA) como um controlador de realimentação positiva para adquirir o modelo inverso da planta, e um controlador Proporcional-Integral-Derivativo (PID) como controlador de realimentação negativa, para garantir estabilidade e lidar com perturbações do sistema. Uma ortótese ativa do joelho e do tornozelo foram os DAA usados e foi aplicada uma estratégia de assistência por seguimento baseado em posição. Foram efetuadas validações sem carga e com dois sujeitos a caminhar numa passadeira a 0.8, 1.0 e 1.2 km/h, com os dois DAA, separadamente, controlados pelo controlo FEL.

A RNA demorou cerca de 90 s a aprender o modelo inverso do DAA, demonstrando versatilidade e estabilidade quando foram aplicadas mudanças na magnitude e velocidade da trajetória de entrada. O controlador de realimentação negativa garantiu estabilidade e conseguiu corrigir o erro quando aplicadas perturbações externas. O controlo de FEL diminuiu o erro de posição em 15%, eliminando o desvio de fase, quando comparado com o controlador PID. Portanto, prova ser um controlo temporalmente eficaz e vantajoso para DAA. Trabalho futuro passa pela validação com mais sujeitos e diferentes estratégias de assistência.

Keywords: Reabilitação da Marcha, Dispositivos Ativos de Assistência, Estratégias de Controlo, Controladores de baixo nível, controlo FEL, controlo PID

ABSTRACT

Gait pathologies often produce abnormal gait patterns, affecting human mobility. Powered assistive devices, such as lower-limb exoskeletons and orthoses, are starting to complement gait rehabilitation, to actively aid or restore the abnormal gait pattern. The human motor control system starts to influence the design of bioinspired architectures for these devices, comprising the definition of distinct levels of controllers (high-, mid-, and low-level) distributed hierarchically. Low-level controllers play an important role in this architecture, ensuring time-effective assistance adaptive to user's needs as gait speed and trajectory.

The main goal with this dissertation is the development of a real-time Feedback-Error Learning (FEL) low-level control to be integrated into a bioinspired control architecture approached in a Stand-alone, Active Orthotic System - SmartOs. The FEL control was performed by means of an Artificial Neural Network (ANN) as a feedforward controller to acquire the inverse model of the assistive device, and a Proportional-Integral-Derivative (PID) feedback controller to guarantee stability and handle with disturbances. A Powered Knee Orthosis and Powered Ankle-Foot Orthosis were used as the assistive devices and a position-based tracking assistive strategy was applied. A validation without human load and with two subjects walking in a treadmill at 0.8, 1.0 and 1.2 km/h with the two assistive devices, controlled by the Feedback-Error Learning control, was performed.

The ANN took around 90 s to learn the inverse model of the assistive device, demonstrating versatility and steadiness when changes to the magnitude and speed of the input trajectory were applied. The feedback controller guaranteed stability and shown good reactions to the applied disturbances. The implemented FEL control was capable to decrease the angular position error by 15% and to eliminate 0.25 s of phase delay when compared to a solo PID controller. Thus, it proves to be a time-effective asset to control assistive powered devices. Future work addresses the validation with more subjects and different assistive strategies.

Keywords: Gait Rehabilitation, Powered Assistive Devices, Control Strategies, Low-level control, Feedback-Error Learning, Proportional-Integral-Derivative Control

Table of contents

Resumo	v
Abstract.....	vii
List of Figures	xi
List of Tables	xv
Acronyms	xvii
1 Introduction.....	1
1.1 Thesis Scope	1
1.2 Motivation & Problem Statement	2
1.3 Goals.....	4
1.4 Research Questions.....	5
1.5 Contribution to Knowledge	6
1.6 Thesis Outline.....	6
2 State of the Art	9
2.1 Lower-Limb Assistive Devices.....	9
2.1.1 Powered Exoskeletons	10
2.1.2 Powered Orthoses.....	12
2.2 High- and Mid-Level Assistive Strategies.....	15
2.3 Low-Level Controllers.....	18
2.3.1 Proportional-Integral-Derivative	18
2.3.2 Optimal Control.....	20
2.3.3 Robust Control	22
2.3.4 Feedback-Error Learning.....	25
2.3.5 Applications in lower-limb assistive devices	27
2.4 Conclusions	29
3 Theoretical Concepts.....	31
3.1 Feedback-Error Learning	31
3.2 Artificial Neural Network.....	34
3.2.1 The Perceptron (Single-layer Neural Network)	35
3.2.2 Multilayer Preceptor	37

3.2.3	The Multi-Layer Perceptron Algorithm	41
3.2.4	Gradient Descent Optimizer	43
4	SmartOs: Smart Control of a Stand-alone Active Orthotic System.....	45
4.1	General Overview.....	45
4.2	SmartOs Description.....	47
4.3	SmartOs Validation.....	50
4.3.1	Initial Considerations	50
4.3.2	Validation Protocol.....	51
4.3.3	Validation Results.....	53
4.4	Conclusions	55
5	Feedback-Error-Learning: low-level control	57
5.1	Methods	57
5.1.1	High-/Mid-level Control: Position-based Tracking Control: Pre-defined gait trajectory	58
5.1.2	Low-Level Control: Feedback-Error Learning Control	59
5.2	Validation Procedure.....	65
5.3	Results & Discussion.....	68
5.3.1	Real-Time Implementation	68
5.4	Conclusions	80
6	Conclusions.....	83
6.1	Future Work	86
	References	87

LIST OF FIGURES

Figure 1 - Example of Wearable Robotic Devices for the lower-limbs. A – H2 lower-limb exoskeleton [3]; B – active knee orthosis by Kavaras [91]..... 9

Figure 2 - Possible types of lower-limbs exoskeletons [17]. 11

Figure 3 - Examples of lower-limb exoskeletons (A: ReWalk™ [22], B: ATLAS [23], C: eLEGs [24], D: HAL [25], and E: H2 [3]). 11

Figure 4 -Examples of lower-limb orthoses: HIP (A: LOPES [91], B: ALEX II [27], C: 1-DOF exo by G. Aguirre-Ollinger [28]), KNEE (D: Knee by Gams [29], E: SCKAFOs [30], F: MIT knee [31]) and ANKLE (AFO by Kao [32], H: PAFOs [33], and I: AFO by Malcolm [34]). 13

Figure 5 - General control framework for powered lower limb orthotic devices. Taken from [22]. 14

Figure 6 - PID control scheme. 18

Figure 7 - LQR control scheme. 22

Figure 8 - SISO feedback loop. 23

Figure 9 - Simplified representation of the system of Fig.10. 24

Figure 10 – General FEL control loop, where: y_{meas} . is the measured position, y_{ref} . is the reference position, \dot{y}_{ref} . is the reference speed, \ddot{y}_{ref} . is the reference acceleration, e is the position error, u_{fb} is the feedback command, u_{ff} is the feedforward command; u is the total control command and y is the performed response. 26

Figure 11 - Example of robotic devices with low-level assistive strategy identified. Lower-Limb : A: [28]; B: [12]; C: [57]; D: [13]; E: [59]. Upper-Limb: F: [61]; G: [62]; H: [63]; I: [64]; J: [65].28

Figure 12 – Illustration of a feedback error controller..... 31

Figure 13 - Illustration of FEL control. A: Training Phase; B: Recall Phase; C: Feedforward in stand-alone. 33

Figure 14 – Usual neural network. 35

Figure 15 - The perceptron network. It has input nodes, preceded by a weighted sum and binary activation function. 35

Figure 16 - The Multilayer Perceptron network, consisting of multiple layers of connected neurons. 37

Figure 17 – Possible training directions in a Multilayer Perceptron. A: forwards; B: backwards	38
Figure 18 - Gradient descent error such that the weights change. Adapted from [80].....	39
Figure 19 – Visual influence of different ANN parameters. A: learning rate parameter; B: momentum and learning rate parameter.....	40
Figure 20 - Proposed algorithm for the gradient descent.....	42
Figure 21 - Operating mechanism of SmartOs system.....	46
Figure 22 - SmartOs: System Architecture.....	47
Figure 23 - Flowchart illustration of used validation protocol to ensure a real-time behavior.	52
Figure 24 – Time behavior of WMSS acquisition functions: a) InertialLAB, time division = 10 ms; b) EMG, time division = 2 ms; FSRs, time division = 10 ms; MuscLAB, time division = 10 ms.	53
Figure 25 - Time behavior of WMSS data packages procedure: a) InertialLAB, time division = 10 ms; b) EMG, time division = 10 ms; FSRs, time division = 10 ms; MuscLAB, time division = 10 ms.....	53
Figure 26 - Time behavior of LLOS functions: a) Low-level control, time division = 1 ms; b) Middle-level control, time division = 10 ms; c) Gait Event Prediction, time division = 10 ms; d) IMUs Acquisition, time division = 10 ms.	54
Figure 27 - Time behavior of LLOS data packages procedure: a) Control strategies, time division = 10 ms; b) IMUs, time division = 10 ms; Orthotic Sensors, time division = 10 ms.....	54
Figure 28 - FEL real-time control loop. I: plant as PAFO. II: plant as PKO. θ_{ref} . is the reference angular position; $\dot{\theta}_{ref}$. is the reference angular speed; $\ddot{\theta}_{ref}$. is the reference angular acceleration θ_{meas} . is the measured angular position; e is the position error; u_{fb} is the feedback command; u_{ff} is the feedforward command; u is the total control command; A is the potentiometer and B is the actuator.....	57
Figure 29 - Interaction between high-level, mid-level and low-level control hierarchy.	58
Figure 30 - Generated knee and ankle trajectory by the high-level.	59
Figure 31 - PID as feedback controller.....	60
Figure 32 - ANN as feedforward controller.....	61
Figure 33 - Flowchart with the implemented algorithm of the FEL real-time control.	62

Figure 34 - Flowchart with neural network offline training algorithm. α and β are configurable scales of the weights initialization and initial learning rate, respectively..... 64

Figure 35 - Used set-up to validate FEL control without load. 1: Power Supply; 2: Power circuit; 3: CCU; 4: LLOS; 5: Actuation device (A: PKO; B: PAFO)..... 65

Figure 36 - Used set-up for the validation with a user on a treadmill. A: PKO; B: PAFO..... 67

Figure 37 - Used reference signals for the validation procedures without load and with the user. A, B, C: PKO reference non-normalized trajectory, speed and acceleration, respectively; D, E, F: normalized input trajectory, speed and acceleration, respectively, for the PKO's ANN; G, H, I: PAFO reference non-normalized trajectory, speed and acceleration, respectively; J, K, L: normalized input trajectory, speed and acceleration, respectively, for the PAFO's ANN... 67

Figure 38 - Results of PID and FEL controller in validation without load. A, D, G, J: PKO Measured trajectory and Reference trajectory signals in the solo PID and Initial, Middle, and Final Phase of FEL, respectively; B, E, H, K: ANN and PID command signals in the solo PID and Initial, Middle, and Final Phase of FEL, respectively; C, F, I, L: PKO Position Error in the solo PID and Initial, Middle, and Final Phase of FEL..... 69

Figure 39 - Results of the FEL controller with a learned PKO IDM for the validation with load. A, D: PKO Measured trajectory and Reference trajectory signals for 0.8 km/h and 1.2 km/h, respectively; B, E: ANN and PID command signals for 0.8 km/h and 1.2 km/h, respectively; C, F: PKO Position Error for 0.8 km/h and 1.2 km/h, respectively. 71

Figure 40 - Results of the FEL controller with a learned PKO IDM for the user-orthosis interaction validation. A, D, G: PKO Measured trajectory and Reference trajectory signals for 0.8 km/h, 1.0 km/h and 1.2 km/h, respectively; B, E, H: ANN and PID command signals for 0.8 km/h, 1.0 km/h and 1.2 km/h respectively; C, F, I: PKO Position Error for 0.8 km/h, 1.0 km/h and 1.2 km/h, respectively. 72

Figure 41 - Results of FEL controller to external disturbances during the user-orthosis interaction. A: PKO Measured and Reference position signals; B: ANN and PID command signals; C: PKO Position Error..... 74

Figure 42 - Results of PID and FEL controllers in validation without load. A, D: PAFO Measured trajectory and Reference trajectory signals in the solo PID and Final Phase of FEL, respectively; B, E: ANN and PID command signals in in the solo PID and Final Phase of FEL, respectively; C, F: PAFO Position Error in the solo PID and Final Phase of FEL, respectively. 75

Figure 43 - Results of the FEL controller with the learned IDM of the PAFO in the validation without load. A, D: PAFO Measured trajectory and Reference trajectory signals for 1.0 km/h and 1.2 km/h, respectively; B, E: ANN and PID command signals for 1.0 km/h and 1.2 km/h, respectively; C, F: PKO Position Error for 1.0 km/h and 1.2 km/h, respectively. 76

Figure 44 - Results of the FEL controller with a learned IDM of the PAFO for the user-orthosis validation. A, D, G: PKO Measured trajectory and Reference trajectory signals for 0.8 km/h, 1.0 km/h and 1.2 km/h, respectively; B, E, H: ANN and PID command signals for 0.8 km/h, 1.0 km/h and 1.2 km/h respectively; C, F, I: PKO Position Error for 0.8 km/h, 1.0 km/h and 1.2 km/h, respectively. 77

Figure 45 - Results of the FEL controller to external disturbances during user-orthosis interaction. A: PAFO Measured and Reference position signals; B: ANN and PID command signals; C: PAFO Position Error..... 79

LIST OF TABLES

Table 1 - Examples of lower limb exoskeletons developed for daily-life activities.....	12
Table 2 - Examples of orthosis evaluated for daily life activities and clinical training.	13
Table 3 - Control strategies applied to powered robotic devices: concept, advantages, drawbacks, implementation, and some examples of powered devices. Information obtained from [35], [38], [8], [13], [40], [9], [10], [41], [42].	16
Table 4 - Effects of PID gains on closed loop characteristics [46].	19
Table 5 - PID control advantages and disadvantages [47].	20
Table 6 - Advantages and disadvantages of LQR optimal control [51], [51].	22
Table 7 - Advantages and disadvantages of H^∞ robust control [59].	25
Table 8 - Advantages and disadvantages of FEL control [15], [65].	27
Table 9 - Assistive devices using low-level control. *Joints: S: Shoulder; E: Elbow; W: Wrist; H: Hip; K: Knee; A: Ankle.	28
Table 10 - Applications with FEL control schemes. *Neural Network structure: RBF: Radial Base Functions; MLP: Multi-layer Preceptor.	34
Table 11 - Some examples of activation functions.	38
Table 12 - SmartOs time constraints. *position data communication: data packages with reference positions for the orthosis to	50
Table 13 – Values to tune and achieved results of the offline implemented ANN. α and β are configurable scales of the weights initialization and initial learning rate, respectively, L are the Input nodes length and M the hidden nodes length.	64
Table 14 - Achieved results for both validations with the PKO.....	73
Table 15 - continued.	74
Table 16 - Achieved results for both validations with the PAFO.....	78
Table 17	Error! Bookmark not defined.

ACRONYMS

AAN – Assisted-as-needed Strategy
ADC – Analog-to-Digital Converter
ANN – Artificial Neural Network
AAFO – Active Ankle-Foot Orthosis
ABLE – Assistive Biorobotic Low-cost Exoskeleton
ADAM – Adaptive Moment Estimation
ALEX – Active Leg Exoskeleton
CAN – Control Area Network
CCU – Central Control Unit
CPG – Central Pattern Generator
DOF – Degree of Freedom
eLEGS – exoskeleton Lower Extremity Gait System
EMG – Electromyography
FEL – Feedback-Error Learning
FES – Functional Electrical Stimulation
FSM – Finite-State-Machine
FSR – Force Sensitive Resistor
HAL – Hybrid Assistive Limb
IDM – Inverse Dynamic Model
IMU – Inertial Measurement Unit
KANN – K-Nearest-Neighbor
LLOS – Low-Level Orthotic System
LOPES – Lower Extremity Powered Exoskeleton
LQR – Linear Quadratic Regulator
MCU – Microcontroller
MEMS – Microelectromechanical Systems
MIT – Massachusetts Institute of Technology
ML – Machine Learning
MLP – Multi-Layer Preceptor

NRMSE - Normalized Root Mean Squared Error

PAFO – Powered Active-Foot Orthosis

PID – Proportional Integral Derivative

PD – Proportional Derivative

PKO – Powered Knee Orthosis

RQ – Research Question

SCI – Spinal Cord Injury

SmartOs - Smart Control of a Stand-alone Active Orthotic System

SVM – Support Vector Machine

WMSS – Wearable Multi-Modular Sensory System

1 INTRODUCTION

1.1 Thesis Scope

This dissertation presents the developed work in the scope of the fifth year of the Integrated Master in Industrial Electronics and Computer Engineering at University of Minho, Portugal, during the academic year of 2017-2018, included in the fields of robotics and real-time control systems.

The project started with an abroad traineeship in *Consejo Superior de Investigaciones Científicas (CSIC)*, Madrid, Spain, included in the ERASMUS placement program, entitled “Adaptive position control for the BioMot Exoskeleton / Power Knee 2”. With this aboard experience, it was possible to assimilate knowledge regarding Feedback-Error Learning control scheme, previously performed on a humanoid platform called “Binocchio” [1], with the goal to implement it in a powered ankle-foot orthosis, a constituent part of the BioMot exoskeleton [2]. The work ended with no satisfactory results, once it was not possible to apply the FEL control to the mentioned orthosis. Nevertheless, expertise was acquired regarding orthosis and FEL approach.

The project has been maintained at University of Minho, in the Biomedical Robotic Devices (BiRD) Laboratory, included in the Center of MicroElectroMechanical Systems (CMEMS) Research Center, now aimed to a powered knee and ankle-foot orthoses, which are modules joints of the H2 lower-limb exoskeleton [3]. This work is inserted in the Smart Control of a Stand-alone Active Orthotic System (SmartOs project), that proposes a bioinspired real-time architecture for human gait training and rehabilitation that has being been developed in our BiRD laboratory.

1.2 Motivation & Problem Statement

Modern societies are privileged with a high quality of life and high average life expectancy, mainly due to the improvement of living conditions, proper and healthy eating habits, good physical exercising, development of medicine and reliable health care systems. When given circumstances threaten people's motion and function, such as aging, injuries, diseases or environmental factors, people turn to rehabilitation facilities so they can search for therapies that can reduce or recover the motor disabilities [4]. Consequently, research for new, improved and faster rehabilitation therapies is a requirement for these populations and for researchers themselves.

A target of the rehabilitation therapies is mobility since it is one of the common and relevant human physical activities. Gait pathologies such as stroke, Parkinson's disease, cerebral palsy, spinal cord injury, muscle dystrophy, degenerative joint disease, multiple sclerosis, and rheumatoid arthritis can deeply affect human mobility [5]. Without it, daily life activities such as getting up from bed in the morning, walking to the bathroom, going for a walk or even working can be drastically affected. Furthermore, the lack of mobility can lead to a deprived social life and a decrease in communication and mental health. Our body health is also affected since our muscles can deteriorate without mobility. As mentioned in [6], human motor impairment can result in social and work (10% in working age) exclusion, early retirement, costly medical treatments, and social assistance. Therefore, rehabilitation therapies regarding the improvement of people's mobility can be a goal to robotics rehabilitation, once it can reinforce and/or restore the human mobility.

Conventional therapeutic approaches, regarding gait rehabilitation, are progressively being complemented with powered robotic assistive devices to treat physical lower limb impairments [4], [7]. These devices can operate as rehabilitative tools that relieve physical therapists from heavy work and enable constant and effective monitoring of patient's outputs in terms of joint kinematics and kinetics [8]. They also allow more didactic and relaxed therapy sessions, as well as can foster user-oriented gait training. Particularly, powered lower-limb exoskeletons and orthoses have been highlighted, as they increase human motor function, build up joint strength and rehabilitate through task-oriented training, acting in parallel with the human lower limb [4], [8], [9], [10].

These wearable robotic devices targeted for gait use are multipart equipment containing mechanical structures, embedded electronics, actuation agents, sensor and monitoring systems, safety and control assistive strategies [9], [11], [12], [13], [14]. Hence, all these different systems should be able to interact in a dedicated software architecture, particularly, sensor and actuation system, to ensure user-oriented, reliable and effective real-time control strategies when assisting the user.

Furthermore, the human motor control system starts to influence the design of bioinspired architectures for lower-limb robotic devices [4], in order to reflect not only the physical interaction between the user and the robotic devices but also with the involving environment. These assistive devices should apply control assistive strategies that ensure repetitive movements of the user's limbs, so an improvement in movement coordination occurs.

Feedback and feedforward controllers play an important role in bioinspired architectures, once they should provide accurate responses to the control assistive strategies, guaranteeing stability to the developed system. For this matter, it becomes imperative the development of robust, time-effective and real-time low-level controllers, capable of guaranteeing an effective actuation during motion.

Certain assistive devices have implemented such low-level controllers, once access to expertise and technology that enables overcoming difficulties in implementing challenges is a reality, such as deeper mathematical understanding about its system modelling [4]. Optimal and robust control are examples of used low-level controllers with these features, providing the assistive strategies with adaptiveness to different rehabilitative scenarios. However, straightforward control laws are much preferred to use in these assistive devices, due to simpler implementation and tuning methods [4]. The Proportional-Integral-Derivative (PID) and its variants are by far the most implemented low-level controller in the abovementioned assistive devices, having satisfactory results without an arduous implementation process. Nevertheless, adopting user-oriented assistance strategies merely based on PID control laws might present flaws, such as the steady-state errors and low range of operation, limiting its versatility.

Therefore, the development of low-level controllers capable of providing low steady-state errors, emphasizing adaptiveness to different rehabilitative situations without system

modeling, is a must to improve the control process and user-oriented assistance in powered assistive devices.

1.3 Goals

The main goal of this dissertation addresses the real-time development of the Feedback-Error Learning (FEL) control [15]. It has a PID feedback controller that protects the user and the assistive device against disturbances and a feedforward controller that uses an artificial neural network (ANN) to learn the inverse dynamic model (IDM) of the assistive device, allowing adaptability to the low-level controller for several gait speeds and changes in magnitude trajectory. The design of the ANN involved, an empirical exploring to find an effective configuration for correctly learning the IDM of the plant and posterior real-time implementation in a wearable processing board.

This control is embedded in SmartOs, an existing modular, bioinspired, real-time, time-effective, and ergonomic orthotic system, which acts and cooperates closely with the human being. Currently, SmartOs includes two actuation devices: Powered Knee Orthosis (PKO) and a Power Ankle-Foot Orthosis (PAFO), both modules from the H2 lower-limb exoskeleton [3]. The proposed control is validated with healthy subjects wearing these orthotic devices, separately, with trajectory tracking control assistive strategy.

To accomplish this main goal, the succeeding goals shall be addressed:

- Goal 1: To investigate in the literature the FEL control in powered lower-limb devices in order to identify remarks in its implementation. Alongside, some analysis to similar control laws, also implemented in powered assistive devices, is performed to properly understand the advantages and drawbacks of each one.
- Goal 2: To validate the time-effective use of SmartOs in gait rehabilitation. A time inspection of the multiple SmartOs modules is done, namely, the prompt execution of the wearable sensor systems, control strategies, and gait analysis algorithms. Additionally, adjustment in SmartOs modules' priority levels are performed to fulfill the temporal requirements of SmartOs.

- Goal 3: To get the assistive device IDM by exploring time-effective function approximators, namely ANN. Exploring optimization methods to enable adaptiveness to different occasion, i.e., different gait speeds and changes in magnitude of trajectory.
- Goal 4: To validate the above-mention ANN, establishing its initial set-up in offline mode.
- Goal 5: To implement and validate the FEL control with hard-real-time constraints (i.e., 1kHz) in SmartOs. An existing trajectory tracking assistive strategy will be used in both the PKO and PAFO, independently, to evaluate the performance of the FEL controller in a real-time environment, including healthy subjects.

1.4 Research Questions

With this dissertation, certain Research Questions (RQs) were identified and answered, addressing the solutions to the main challenges covered in this project:

- RQ1: Which are the FEL control advantages and limitations over other low-level controls applied to powered assistive devices? This research question is answered in Chapter 2.
- RQ2: Is SmartOs a time-effective assistive and monitoring rehabilitation system? This research is answered in Chapter 4.
- RQ3: Are ANN capable of effectively learn the IDM of a powered assistive lower-limb device, for different users, trajectories and speeds? This research is answered in Chapter 5.
- RQ4: Can FEL control provide good real-time performance to a powered assistive lower-limb device? This research question is answered in Chapter 5.

1.5 Contribution to Knowledge

The main contributions identified in the current dissertation are the following:

- An ANN with supervised learning, implemented with backpropagation, stochastic gradient descent (SGD) training algorithm and Adaptive Moment Estimation (ADAM) optimizer, for multiple platforms, with online training. The algorithm is capable of quickly learning the IDM of PKO and PAFO assistive devices.
- A real-time, low-level FEL control for a PKO and PAFO assistive devices running at 1 kHz, with low steady-state errors and adaptiveness to different rehabilitative situations without system modeling.
- Contribution to the time-effective implementation of the bioinspired, modular and effective SmartOs.

Furthermore, a scientific publication was accepted in the scope of this dissertation:

- Pedro Nuno Fernandes, Joana Figueiredo, Juan C. Moreno, Cristina P. Santos, “Feedback-Error Learning for Gait Rehabilitation Using a Powered Knee Orthosis: First Advances”, 6th Portuguese Meeting on Bioengineering (ENBENG).

1.6 Thesis Outline

This master’s dissertation is organized in the following 5 chapters.

Chapter 2 presents a state of the art on lower limb’s assistive devices highlighting their need in the field of gait rehabilitation. A definition and inspection of high-level and middle-level assistive strategies applied in these assistive devices are made. To finish, a description of low-level control and its applications on the above-mention devices are noted.

Chapter 3 addresses the theoretical concepts behind the FEL control. It analyzes its theory, some insights about ANN and its training algorithms are presented.

In chapter 4, an insight about the SmartOs architecture is made, presenting a system overview regarding its constituent software and hardware. It is pointed out the contribution made to its implementation, ending with the performed validation tests on the orthotic system.

Chapter 5 presents a discussion on the performed design and offline implementation finishing with the real-time implementation in SmartOs. The obtained results for two validations are presented as discussed.

Lastly, in chapter 6 the conclusions are presented, and the research questions are answered. The future work is also addressed.

2 STATE OF THE ART

The current chapter presents a review of the main concepts and technologies involved in this dissertation. It begins by underlining the benefits of lower-limb assistive devices for therapy and gait rehabilitation. An inspection about specific lower-limb exoskeletons is made and a more thorough approach is taken regarding orthoses, concluding with a bioinspired framework for its control. Afterwards, high-level assistive strategies are defined, analyzed and compared. Subsequently, low-level controllers applied in orthotic devices are described and inspected. Lastly, applications for the specified low-level controllers are listed.

2.1 Lower-Limb Assistive Devices

Research in the field of gait rehabilitation including new assistive and empowering lower-limb devices have been widely addressed, especially in powered lower-limb exoskeletons and orthoses [16]. These anthropomorphic and electromechanical devices, ultimately augment, reinforce or even restore an abnormal human gait, rehabilitating users affected by neurologic, motor diseases or injuries [17]. Figure 1 shows the H2 lower-limb exoskeleton and a PKO.

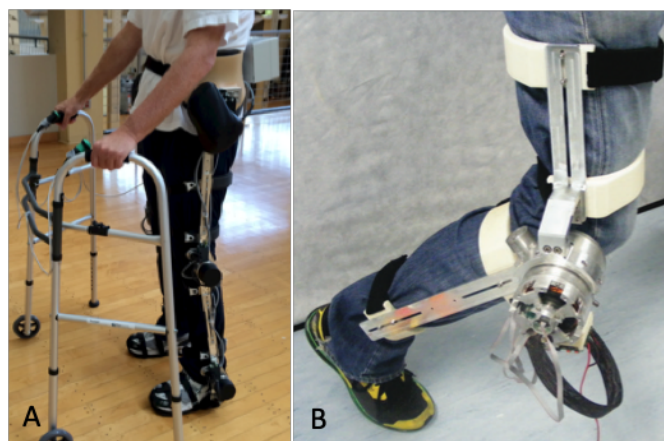


Figure 1 - Example of Wearable Robotic Devices for the lower-limbs.
A – H2 lower-limb exoskeleton [3]; B – active knee orthosis by Kavaras [91].

The examples observed in Figure 1, are lower-limb assistive devices with active/powering characteristics, i.e., it allows them to actively control the joints of the devices [18]. Powered

devices are, normally, combined with actuators and sensors that work together, on account of control architectures in which power is added at appropriate phases of the gait cycle, allowing users to walk in a self-comfortable manner. The power feature makes these assistive devices adjustable and compliant to the user, rather than a passive/unpowered behavior that is a rigid and stiff device, that can only balance the limb for a certain range of motion [19].

In gait rehabilitation and assistance, there has been a great interest in the development of rehabilitative powered lower-limb devices to relieve or overcome the drawbacks associated with traditional rehabilitation therapies. Particularly, robot-aid therapies allow: 1) less lack of repeatability through high degree of training with repetitive tasks which they can provide; 2) longer training sections, since they relieve the repetitive and heavy work that relies on the physical therapists; and 3) a better analysis of the therapeutic progress, given the possibility of not only evaluate by visual observation but also based on objective data collected in real-time from these devices [16], [17].

Powered lower-limb exoskeletons and orthoses have been highlighted as a preferred robotic assistive system, as they are applied in the real environment and not only in the clinical environment [2]. When the intention is to empower all the lower-limb joints, i.e., trunk (T), hip (H), knee (K), ankle (A) and feet (F), the used technology is a powered exoskeleton. If the purpose is to actively assist one of these lower-limb structures, an powered orthosis is applied [20], [21].

2.1.1 Powered Exoskeletons

Commonly, powered lower-limb exoskeletons are addressed as robots with the capability to provide extra motion energy, improve load carrying ability, and aid lower-limb impaired subjects to regain independent gait [17]. According to the mechanical joints covered in these devices, they can be classified as: trunk-hip-ankle-foot (THKAF), hip-knee-ankle-foot (HKAF), trunk-hip-knee (THK), hip-knee (HK), and knee-ankle-foot (KAF) exoskeletons, as disclosed in Figure 2.

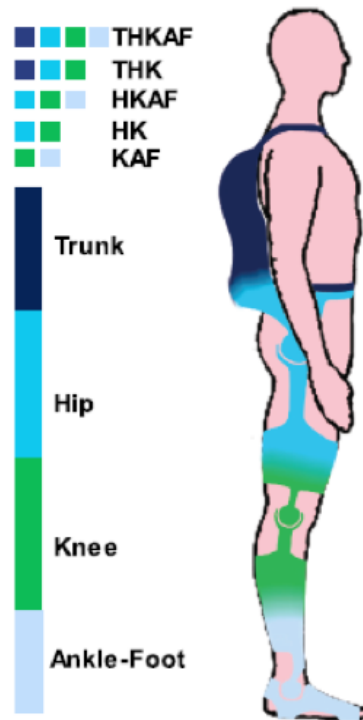


Figure 2 - Possible types of lower-limb exoskeletons [17].

Some examples of powered lower-limb exoskeletons can be found in Figure 3.



Figure 3 - Examples of lower-limb exoskeletons (A: ReWalk™ [22], B: ATLAS [23], C: eLEGs [24], D: HAL [25], and E: H2 [3]).

Table 1 presents some of the main lower-limb exoskeletons that have been evaluated for daily-life assistance, with focus on the main actuated joints and their applications.

Table 1 - Examples of lower limb exoskeletons developed for daily-life activities.

Devices	Target Joints		Applications
	Active	Passive	
ReWalk™ [22]	Hip, Knee	Ankle	- SCI Paraplegics
ATLAS [23]	Hip, Knee	Ankle	- Quadriplegic children
eLEGs [24]	Hip, Knee	Ankle	- SCI Paraplegics
HAL [25]	Hip, Knee,	Ankle	- Power augmentation - Impaired subjects
H2 [26]	Hip, Knee, Hip	-	- Stroke survivors

As suggested by Table 1, a great number of lower-limb exoskeletons apply powered movement actuation at distinct joints, being hip and knee the most addressed active joints. Moreover, the reviewed devices have been intended for gait assistance and rehabilitation, being designed for pathological patients with gait disorders (due to neurologic injuries and/or motor disabilities).

2.1.2 Powered Orthoses

Powered lower-limb orthoses are designed for a range of applications, focusing on restoring an abnormal gait pattern affected by specific leg injuries or impairments. They are divided into hip, knee, and ankle systems.

Some of the main orthoses that have been evaluated are presented in Figure 4. These orthoses are detailed in Table 2 regarding the actuated target joints and the main application considering the scope of assistance and the target end-users.

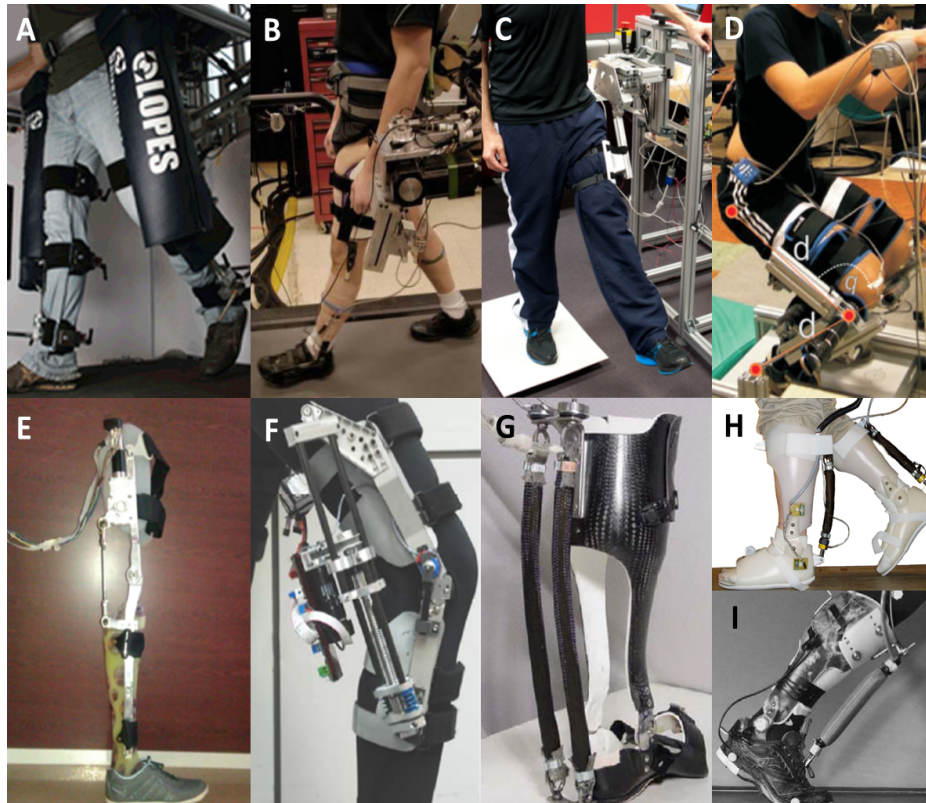


Figure 4 -Examples of lower-limb orthoses: HIP (A: LOPES [91], B: ALEX II [27], C: 1-DOF exo by G. Aguirre-Ollinger [28]), KNEE (D: Knee by Gams [29], E: SCKAFOs [30], F: MIT knee [31]) and ANKLE (AFO by Kao [32], H: PAFOs [33], and I: AFO by Malcolm [34]).

Table 2 - Examples of orthosis evaluated for daily life activities and clinical training.

Devices	Actuated Joint	Assistance	Subjects
ALEX II [27]	Hip	- Treadmill walking.	- Stroke survivors.
1-DOF exo by G. Aguirre-Ollinger [28]	Hip	- Swing motion.	- Healthy.
Knee by Gams [29]	Knee	- Squatting motion augmentation.	- Healthy.
SCKAFOs [30],	Knee	- Swing motion.	- Affected by poliomyelitis.
MIT knee [31]	Knee	- Running.	- Healthy.
AFO by Kao [32]	Ankle	-Treadmill walking.	- Subjects with ankle impairments.
PAFOs [33]	Ankle	- Treadmill walking.	- Younger and older (sarcopenia).
AFO by Malcolm [34]	Ankle	- Treadmill- walking to running transition;	- Healthy

Table 2 shows that active orthoses for hip, knee and ankle joints have been designed to assist and rehabilitate. It also suggests that these devices can be applied to many applications, particularly, patients with neurologic disorders (e.g. stroke survivors, and affected by the poliomyelitis virus), deficits at the human joints (e.g., *genu recurvatum*, and drop-foot), and muscle weakness, or healthy subjects when performing daily-life-activities (climbing stairs, squatting with or without heavy loads, stand-to-sit and sit-to-stance, and running).

In addition to the advancements of the mechanical design, sensors and actuators of these powered lower-limb devices, assistive strategies have to be thought as a key feature. Additionally, control approaches capable of providing adequate responses should be a concern, namely, reflecting the physical interaction between the user, the robotic devices, and the involving environment. When dealing with such devices, the human motor control should be considered in a continuous assessment. For this matter, a generalized control framework has been proposed by Tucker *et. al* [22] as a template for the next generation of multifunctional controllers for active lower-limb exoskeletons and orthoses. This model is illustrated in Figure 5.

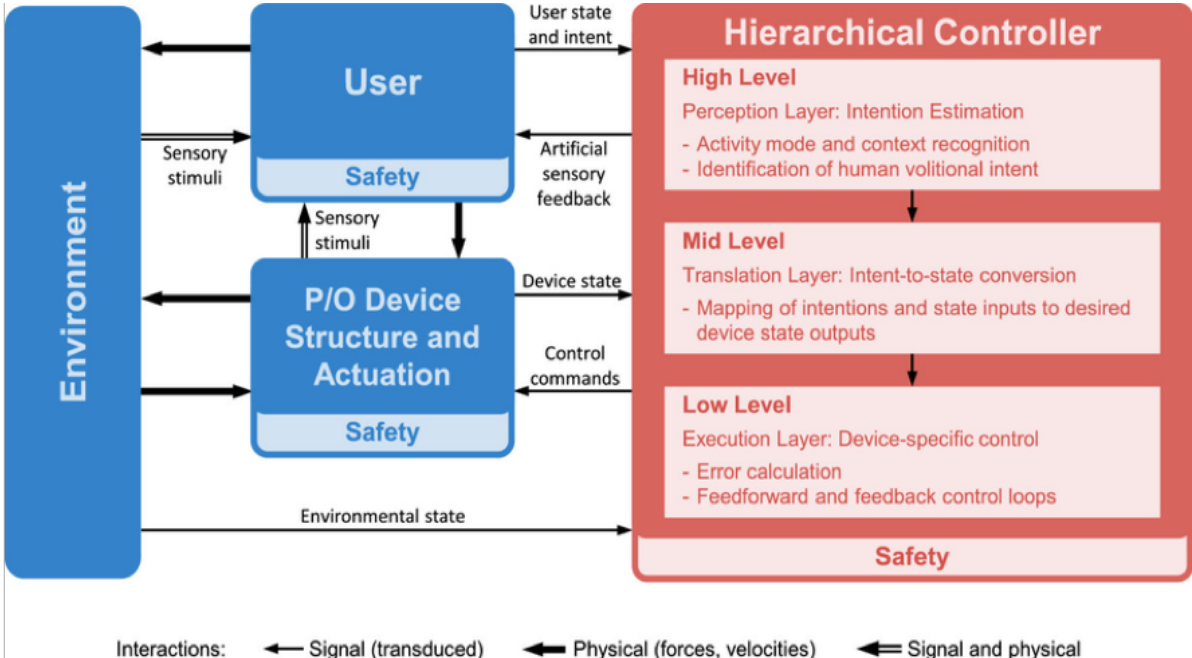


Figure 5 - General control framework for powered lower limb orthotic devices. Taken from [22].

This generalized framework includes four major subsystems: the hierarchical control structure, the user, the navigated environment, and the powered device itself. Safety mechanisms are included throughout the architecture, in order to emphasize the human-robot interaction. The proposed hierarchical control architecture can be divided into three main layers: high-, mid-, and low-level controllers.

The high-level controller implements a perception layer. At this layer, the user's state and intentions can be inferred to adjust the motion assistance. Typically, it comprises the implementation of motion intentions estimation, activity mode recognition and gait trajectory generation [22].

The mid-level controller implements the translation layer. The user's locomotion intentions from the high-level controller are passed to this layer and are translated into device states. At this level, the gait pattern (gait phases and events) is disclosed, and the control law is applied [22].

Lastly, the low-level controller tracks the desired device states provided by the mid-level layer. Within feedback and/or feedforward control, the error with respect to the current state is computed, and commands are sent to the actuators [36]. This level is commonly implemented through position and torque control where a position or torque reference is provided to the controller, respectively [36].

2.2 High- and Mid-Level Assistive Strategies

In the high layer of assistance, it is expected that the state and intentions of the user can be inferred to adjust the adequate motion assistance to provide.

With the intention of the user identified, it is up to the mid-level to convert this intention state to a desired device state output. This transition layer implements controllers that are capable of mapping the input state received from the upper layer to an output state that the lower layer can execute to a specific device. Moreover, it implements motion control strategies for assistive devices, that can be generally divided into tracking control and assisted-as-needed (AAN) control [22], [23].

The tracking control integrates mechanisms based on predefined trajectories, such as the position and torque control, where the patient receives constant assistance, while the legs

being guided on a fixed, rigid trajectory. The position control aims to predefine the target joint trajectory through the gait analysis of limbs movement and/or joints kinematics [37]. Recent studies suggest parameterizing the joints trajectories in accordance with the users' body conditions and movement phases to improve the flexibility and comfort [38]. Similarly, the torque control assists the lower-limb according to a predefined torque.

Alternatively, the AAN control defines the actuator state in function of the user's impairment and training progress, encouraging a voluntary participation by the user, being assisted only as needed and how much is needed and, consequently, they are encouraged to interact with the devices [23][39].

For this purpose, (adaptive) impedance control, model-based control, fuzzy-control, myoelectric control, among other strategies, can be adopted in the attempt that the powered device supplies AAN [39], [8], [13]. Table 3 summarizes the concept, advantages, disadvantages, control strategies and used assistive device inserted in both categories: tracking and AAN control.

Table 3 - Control strategies applied to powered robotic devices: concept, advantages, drawbacks, implementation, and some examples of powered devices. Information obtained from [35], [38], [8], [13], [40], [9], [10], [41], [42].

Strategy	Concept	Advantages	Drawback	Controllers	Devices
Position- and Torque-based Tracking Control	Impose repetitive joint trajectories based on pre-recorded data.	<ul style="list-style-type: none"> - Easy implementation; - May be used in patients with severe pathology or paraplegic. 	<ul style="list-style-type: none"> - Gait may be not natural; - May decrease the motor learning; - May reduce subject's active participation and motivation. 	<ul style="list-style-type: none"> - PID position and torque controller. - PD position and torque controller. 	<ul style="list-style-type: none"> e.g., ATLAS, eLEGS, HAL, Mindwalker, SCKAFOS, PKO
Impedance control / Adaptive impedance control	Modify the impedance parameters of human limbs by setting the device's parameters.	<ul style="list-style-type: none"> - Impedance selected according to patient's state (e.g., disability level); - Increase the leg's natural frequency - Considers human-robot interaction. 	<ul style="list-style-type: none"> - Inter/intra-subject difference affects the biomechanical modelling (masses and dimensions of subject's body segments, and passive torque); - Controller requires real-time computation, since the total torque required is constantly changing. 	<ul style="list-style-type: none"> - FSM; - Radial basis function neural network; - Multi-Layer perceptron neural network; - PD position controller. 	<ul style="list-style-type: none"> e.g. Hussain, knee orthosis by Daachi

Table 3 - continued

Strategy	Concept	Advantages	Drawback	Controllers	Devices
Model-based control	Action is computed on the basis of human-robot model. Considers gravity compensation, zero moment point (ZMP) and balance criteria.	- Assists according to modelled human-robot interaction.	- Efficacy depends on the accuracy of human-robot model. - Requires series of sensors to recognize kinematics and dynamics variable.	- ZMPs combined with 2 neural network to decrease the time-consuming computation. - Human body model.	e.g., ABLE, HAL, WPAL, Hip joint by Y. Yu
Adaptive oscillators-based control	Recursively follow user's intended movements, relying on the periodicity of the gait pattern.	- Realizes a model-free control able to synchronize with joint angle, and to provide continuous gait phase.	- Limited to subjects who can deliver periodic and stable bio-signals (healthy or subjects with residual motion capability)	- CPGs based on PD position control.	e.g, Lower-limb exo by N. Tagliamonte, ALEXII, LOPES, 1-DOF hip by G. Aguirre-Ollinger
Fuzzy controllers	Adaptive models that recursively follow user's intended movements, relying on a fuzzy-logic layer.	- Considers the user's movement intention.	- Requires many variables to be tuned manually according to the specific motion tasks and individuals. - Time consuming.	- Fuzzy-logic.	e.g, Exo by Sogang University
Pre-defined action based on gait pattern	Recurrent gait phase transitions are used to regulate the controller's action, resulting in a flexible interaction.	- Desired assistance are determined by referring to walking data.	- Do not consider the user's movement intention; - Needs to be tuned with different walking conditions.	- FSM - KNN and SVM.	e.g., MIT Exoskeleton, MIT AAF0, Soft Exosuit, Hip exo by C. Lewis
Myoelectric control	Actuator torque is based on EMG signals obtained from desired muscles.	- Takes in consideration the user's disabilities through muscle activity; - Tracks patient's rehabilitation progress;	- Muscular activity is very different between subjects and depending on the tasks; - Requires repeated calibration - Requires signal processing	- PID controller	e.g. AAF0 by P. Kao, AAF by Ferris

2.3 Low-Level Controllers

Low-level controllers are inserted in the execution layer, closer to the device. An error calculation is performed with the current state provided by the device sensors and the reference state received from the upper layer (high-/mid-level), feeding a feedback and/or feedforward controller. Subsequently, a command is sent to the device actuator to track the desired state [36]. This level is commonly implemented through position and torque control, where a position or torque reference is provided to the controller, respectively [36].

Each of the low-level strategies analyzed in the current subchapter have been implemented in assistive robotic systems, in particular, exoskeletons and orthoses that are specified in Table 4.

2.3.1 Proportional-Integral-Derivative

The PID control law provides a generic and efficient solution to a wide range of real-world control problems being the most common form of feedback control [43], [44], [45]. In process control today, more than 95% of the control loops are of PID type, being the Proportional-Integral (PI) control the most used [45].

In Figure 6 it can be seen a generic control loop using a PID controller.

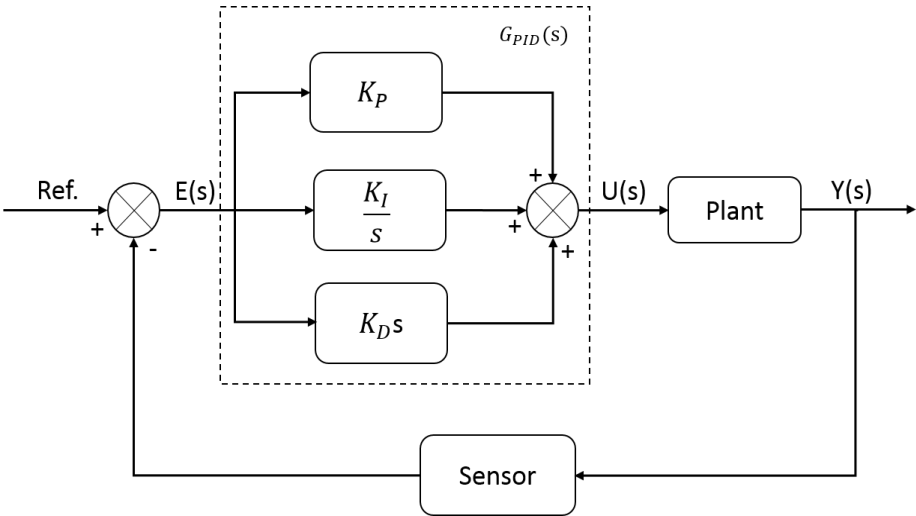


Figure 6 - PID control scheme.

The transfer function of a PID controller can be written as indicated in Equation (1):

$$G_{PID}(s) = \frac{U(s)}{E(s)} = K_P + K_I \frac{1}{s} + K_D s \quad (1)$$

where $U(s)$ is the control signal acting on the error signal $E(s)$, K_P is the proportional gain, K_I is the integral gain, K_D is the derivative gain, and s is the argument of the Laplace transform. The proportional term provides an overall control action proportional to the error signal through the all-pass gain factor. The integral term reduces steady-state errors through low-frequency compensation, having a gradual action over the integral of the error. The derivative term improves transient response through high-frequency compensation, taking an anticipatory action over the derivative of the error.

A good proportional gain value leads the output variable ($Y(s)$) of the plant to a fast-transient response, fast-tracking the reference value, and small overshoot without bringing the system to noticeable oscillations and/or instability. The integral term is well chosen when the offset value of $Y(s)$ is removed and the rising time of the output variable is decreased, without increasing its overshoot, oscillation, and stability time. Lastly, the derivative gain is properly selected when the process was speed-up and stabilized, reducing overshoot, error, and oscillation values, avoiding instability [46]. Having these features in mind, it is possible to tune the PID controller considering the whole system as a black-box and choosing the best gain values by analyzing the response of the output variable, by trial-and-error. Table 4 reviews the effects of independent changes of the PID gains, stating their effects on closed-loop responses.

Table 4 - Effects of PID gains on closed loop characteristics [46].

	Rise Time	Overshoot	Settling Time	Steady-State Error	Stability
Increasing K_p	Decrease	Increase	Minor Increase	Decrease	Degrade
Increasing K_i	Small Decrease	Increase	Increase	Large Decrease	Degrade
Increasing K_d	Small Decrease	Decrease	Decrease	Minor Change	Improve

Some tuning techniques were tailored to the PID control, although iterative tuning can also be applied. The most well-known tuning methods are those developed by Ziegler and Nichols [43]. These methods are based on the characterization of process dynamics by a few

parameters and simple equations for the controller parameters. These methods are widely referenced, because they give moderately good tuning only in restricted situations [45].

The real challenge when implementing the PID controller is to balance the impact of all three gains to the whole system. If the system parameters cannot be precisely tuned, the systems response may not resist the uncertainties and disturbances, and thus present a control with low robustness. Although, when an ideal combination between them is achieved, this controller has a durable, steady and good responses. Nevertheless, if the evolving environment suffers changes, the controller could not be capable to adapt, being necessary to re-tune the PID parameters, making it a drawback comparing with another controllers. Table 5 lists the PID control advantages and disadvantages.

Table 5 - PID control advantages and disadvantages [47].

Advantages	Disadvantages
Feasibility and easy to implement.	Balance all gains can be arduous.
System model is not necessary.	System may not resist to uncertainties and disturbances.
Variety of tuning methods (black box).	Changes in the environment affect control response.

2.3.2 Optimal Control

Optimal control theory consists on designing controllers that optimize some desirable characteristics in order to achieve an optimal state of the system when exposed to disturbances [48], [49]. Therefore, the optimal control law is based on the idea of minimizing a J criterion on evolution variables of the system to be controlled. Two useful criteria are the minimum linear quadratic cost and the minimum time criterion of establishment [50].

2.3.2.1 Linear Quadratic Regulator (LQR)

The LQR, is based on the idea of minimizing the integral of the heavy sum of squares of the state variables and command variables. The cost function to diminish is presented in Equation (2):

$$J(x(t), u(t)) = \int_0^{\infty} [x^T(t)Qx(t) + u^T(t)Ru(t)]dt \quad , Q \geq 0, R > 0 \quad (2)$$

where $x(t)$ are the state variables of the system to be controlled, and $u(t)$ is the command variable of the controller. Q is the matrix responsible to weight the size of the states. If Q has high values, then small deviations of the state significantly increase the cost function (tight control with high gain). The matrix R weights the control values. Contrary to what happens with Q , if R has high values, then the control action will be small [51]. Varying the values of these matrices, may be interesting to differently adjust the weight of each state value and control values.

To perform the LQR as a linear feedback controller, it is necessary to find the value of the feedback vector K_{opt} of the complete feedback that causes a satisfying system response behavior. The optimal control law is state in Equation (3):

$$u_{opt}(t) = -K_{opt}x(t) \quad (3)$$

and considering the linear and time invariant state-space model representation in Equation (4) and (5):

$$x'(t) = Ax(t) + Bu(t), x(t_0) = x_0 \quad (4)$$

$$y(t) = Cx(t) \quad (5)$$

where $x'(t)$ are the first order derivatives of the state variable, A state-space matrix of the system, B the input matrix, and C the output matrix. In order to find the K_{opt} that minimizes the cost function J , the Riccati's equation is often used [49] (Equation (6)):

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \quad (6)$$

where finding the value of P allow us to discover the optimal values of K , as noted in Equation (7):

$$K_{opt} = R^{-1}B^T P \quad (7)$$

Therefore, the values of Q and R must be tuned, in order to minimize the above-mention cost function. If Q contains values close to 0, the state variables have low or none contribution to the control, meaning that their excursions are not relevant, while the control tries to force the state variables to values near the reference. Regarding the matrix R , when it has low levels,

the energy spent to bring the state variables close to the reference will be close to infinite and very fast, spending a lot of control energy. Consequently, minimizing the cost function J results in moving the state variables near the reference values with as little control energy and state deviations as possible, with the balance between control energy and state deviations specified via the Q and R matrices, respectively [51]. When K_{opt} has been found, the control scheme represented in Figure 7 can be applied.

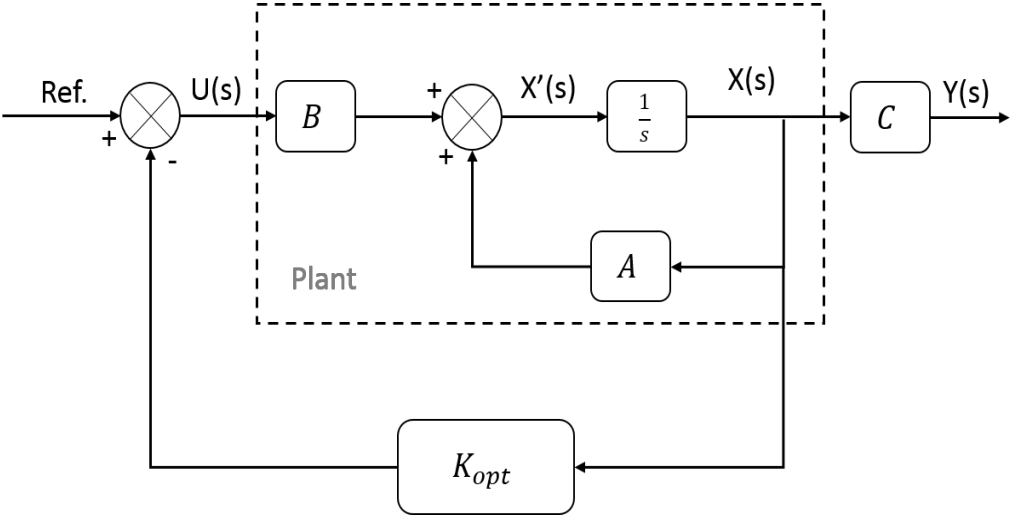


Figure 7 - LQR control scheme.

Hence, the advantages of the LQR controller can be summarized in the following Table 5.

Table 6 - Advantages and disadvantages of LQR optimal control [51], [51].

Advantages	Disadvantages
Stability is guaranteed if all state variables are available and if the system model is known	System model must be available as well as the state variables;
Simple control scheme	Obtaining the solution for the Ricatti 's equation can be difficult.

2.3.3 Robust Control

The robust control methods aim to bound uncertainties, so their results meet the control system requirements in all cases, being prepared to operate in a worst-case scenario [52].

There are a variety of techniques that have been developed for robust control, such as: 1) H_2 and H_∞ that uses Hankel norms to measure control system properties; 2) parameter estimation, a method that establishes boundaries in the frequency domain that cannot be crossed to maintain stability [53]; 3) fuzzy control, that is based upon the construction of fuzzy sets to describe the uncertainty inherent in all variables and a method of combining these variables called fuzzy logic [54]. The H_∞ control is going to be the chosen robust control technique to analyze once was the only one found in the literature that was applied to orthoses.

2.3.3.1 H_∞ Robust Control

H_∞ robust control considers the minimization of the ∞ -norm of the sensitivity function of a single-input-single-output (SISO) linear feedback system [55]. Consider the following SISO feedback system in Figure 8 - SISO feedback loop.:

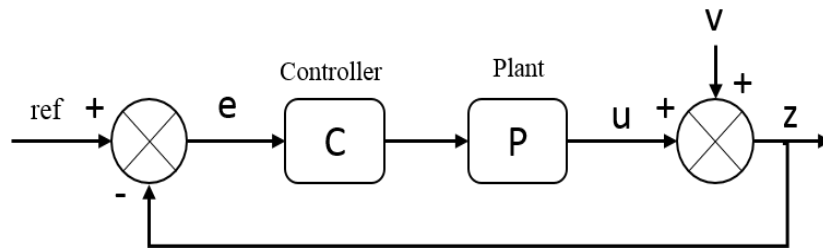


Figure 8 - SISO feedback loop.

where u is the control signal acting on the error signal e , z is the output variable and v represents the disturbance signal that affects the plant. The controller and the plant have C and P as transfer function, respectively. The sensitivity function of the feedback system in Figure 8 - SISO feedback loop. characterizes the sensitivity of the output variable z to disturbances only ($ref=0$), and can be written as follow:

$$Z = V - PCZ \Leftrightarrow Z = \frac{1}{1 + PC} V \quad (8)$$

$$\Leftrightarrow Z = SV \xrightarrow{\text{sensitivity}} S = \frac{1}{1 + PC} \quad (9)$$

Therefore, S represents the sensitivity function of the feedback loop – Figure 9

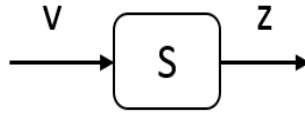


Figure 9 - Simplified representation of the system of Fig.10.

Preferably it is intended for $S = 0$. The idea of the H_∞ robust control is to optimize a controller that makes the closed-loop system stable and minimizes the peak value of the sensitivity function [52] described in Equation (10):

$$\| S \|_\infty = \max |S(j\omega)| \quad (10)$$

In order to achieve good results, an optimization considering the worst case should be done, once it purposes to minimize the effect of the worst disturbance value over the system. Therefore, the peak value of the sensitivity is defined by the square of the 2-norms on the input and output signals [55]. It follows that H_∞ robust control is concerned with the minimization of system norms as shown in Equation (11):

$$\| S \|_\infty = \frac{\| z \|_2^2}{\| v \|_2^2}, \quad \| v \|_2 < \infty \quad (11)$$

The complementary sensitivity function is also needed, being stated in Equation (12):

$$T = \frac{PC}{1 + PC} \quad (12)$$

As above-mentioned, the H_∞ robust control goal is to design a controller C , which minimizes the influence of disturbances signals on the output variable. Thus, minimizing the closed loop norm $\| S \|$. This objective is accomplished by bounding the values performance value $\sigma(S)$, and robustness value $\sigma(T)$. For that, the norm presented in Equation (13) must be minimized:

$$C_{min} = \min \| N \|, N = \begin{bmatrix} W_S S \\ W_T T \end{bmatrix} \quad (13)$$

where the values of W_s are the performance weighting function and the values W_t are the robustness weighting function to be inferred by the control designer. These values can be discovered by applying the relations shown in Equation (14) and (15):

$$|S(j\omega)| < \frac{1}{W_s(j\omega)} \quad (14)$$

$$|T(j\omega)| < \frac{1}{W_t(j\omega)} \quad (15)$$

W_s and W_t are to restrict the magnitude of the sensitivity and complementary sensitivity function, respectively. This technique to design the controller is denominated Loop Shaping. With this technique, the parameter W_s and W_t are altered in a way so that the frequency response of the system stays stable within the desired limits. There are several methods to tune the parameters of the Loop Shaping such as: Glover-McFarlane H_∞ loop-shaping procedure [56], quantitative feedback theory [53], Automatic Weight Selection [57].

Table 7 lists H_∞ robust control advantages and disadvantages over the classical ones.

Table 7 - Advantages and disadvantages of H_∞ robust control [59].

Advantages	Disadvantages
Applicable to problems involving multivariable systems with cross-coupling between channels.	Deeper mathematical understanding.
Minimizes disturbances impact over the system.	Good information about the system model.
	Tuning the controller.

2.3.4 Feedback-Error Learning

FEL control was proposed by Kawato [15], that was inspired in the “architecture” of the human motor control. FEL tries to describe how the human central nervous system acquires the internal models of the human body that are responsible to the motor function, for robots that had to perform random trajectories [15], [58]. FEL includes a feedforward controller that has the IDM of the target plant (acquired with learning methods resembling the human cerebellum) to learn motor commands to produce trajectories. Additionally, FEL has a feedback controller to provide stability during the IDM learning, like the sensory systems that the human motor system has [15], [59], [60]. When the learning process is finished by means

of the feedback error, the learned IDM is capable to perform the desired trajectory, reducing the feedback from the sensory system. Kawato understood that this architecture can be used for inverse dynamics modeling for robotic systems [15].

The inverse dynamics modeling architecture of FEL is shown in Figure 10.

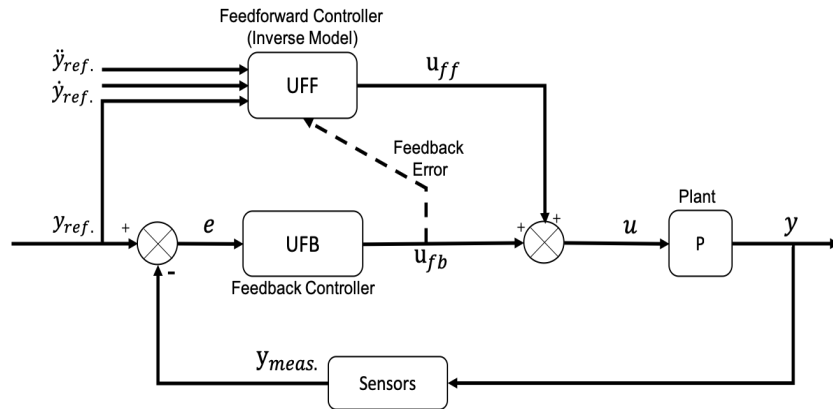


Figure 10 – General FEL control loop, where: $y_{meas.}$ is the measured position, $y_{ref.}$ is the reference position, $\dot{y}_{ref.}$ is the reference speed, $\ddot{y}_{ref.}$ is the reference acceleration, e is the position error, u_{fb} is the feedback command, u_{ff} is the feedforward command; u is the total control command and y is the performed response.

As observed in Figure 10, the feedforward controller UFF maps reference inputs into feedforward command signals, i.e., $u_{ff} = UFF(y_{ref.}, \dot{y}_{ref.}, \ddot{y}_{ref.})$, compensating the insight of the systems dynamics, obtaining high tracking accuracy [61]. The singularity of the proposed FEL controller is the presence of a feedforward controller capable of learning the IDM of the controlled plant [15]. This feedforward controller can be implemented using regression techniques, establishing a function approximator. During control, the feedforward controller is adapted, such that it learns the IDM plant model. It was verified by [61] and [15] that when the output of the feedback controller is used as a learning signal, the input-output relation of the feedforward controller converges to the IDM of the plant. Kawato used as function approximator, the ANN with supervised learning [58].

The feedback controller UFB , aside from providing the error signal to learn the IDM of the plant, it determines the tracking performance while the IDM is being learned. Eventually, the feedback controller is responsible to compensate for the existing disturbances, when the learning phase for the IDM is learned.

The FEL controller can be thought as a two-phase controller: (1) when is acquiring the plant's IDM – training phase; (2) when has the plant IDM – recall phase. During the learning phase, the u_{ff} signal has a small contribution to total control command u , compared to the

one provided by the feedback controller (u_{fb}), once the feedforward controller is learning the plant's IDM. However, as the training proceeds, the feedforward controller starts to learn the IDM and its control signal becomes more predominant to the total control output. The value of the feedback command u_{fb} is an indication to the feedforward controller of the mismatch between the true plant's dynamics and the learned IDM [62]. The controller shifts to the recall phase when the feedback controller has a small contribution to the total control command and the feedforward controller is the major grantor of the control, meaning that the feedforward has acquired the plant's IDM. In the recall phase, the feedback controller becomes subjected to eliminate random disturbances. With the two controllers combined, the overall performance of the FEL control is increased [63], [64].

Table 8 summarizes the advantages and disadvantages of the FEL control scheme.

Table 8 - Advantages and disadvantages of FEL control [15], [65].

Advantages	Disadvantages
It is not required an explicit model of the controlled system.	Training phase can take long periods to learn the IDM or never learning the model at all.
Allows the feedback controller to control the system while the IDM is learned in real-time.	Adjusting the design parameters of the implemented regression technique can be an arduous assignment.
Overall performance of the controller has adaptive characteristics.	

2.3.5 Applications in lower-limb assistive devices

Table 9 refers examples to upper and lower-limb assistive devices found in the literature. It summarizes the joints to rehabilitate, the used low-level control, and the number and health condition of the subjects involved in the validation process. Figure 11 displays the devices referred in Table 9.

Table 9 - Assistive devices using low-level control. *Joints: S: Shoulder; E: Elbow; W: Wrist; H: Hip; K: Knee; A: Ankle.

Device	Joints*	Controller	Validation (subjects)
Aguirre-Ollinger [28]	K	LQ torque	2 healthy
HAL [25]	H, K, A	PD position	1 SCI
Gomes et al [66]	H, K, A	Robust H^∞	simulation
Nakul Gopalan [67]	(biped model)	FEL position	simulation
H2 [3]	H, K, A	PID position/torque	3 stroke
MINDWAL-KER [68]	H, K, A	PD position	4 SCI
PKO [42]	K	PID torque	5 healthy
L-Exos [69]	S, E	Slide-mode PD	9 stroke
Rehab-Robot [70]	S, E, W	PID feedback	1 healthy
Resquín, F. [71]	S, E, W	FEL position	12 healthy: 4 brain injured
ARMIN III [72]	S, E	PD position	1 healthy (5 undergoing stroke)
MGA [73]	S, E, W	PD position	1 healthy

By analyzing Table 9, some conclusions can be pointed out. Firstly, it is possible to infer that position-based tracking control is preferred over the torque-based tracking control. Secondly, most of the shown systems used PID or PD low-level controls and the FEL control was never applied to a lower-limb orthotic device. Nevertheless, robust control such as H^∞ and LQR optimal control are also approached in the literature.

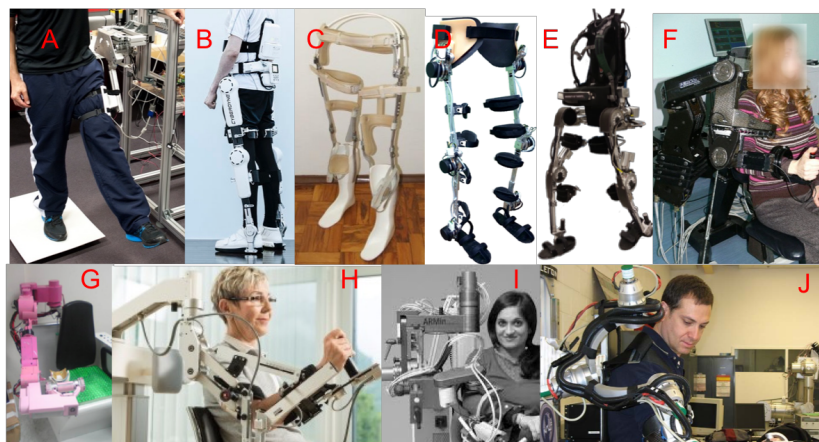


Figure 11 - Example of robotic devices with low-level assistive strategy identified. Lower-Limb : A: [28]; B: [12]; C: [57]; D: [13]; E: [59]. Upper-Limb: F: [61]; G: [62]; H: [63]; I: [64]; J: [65].

2.4 Conclusions

In this chapter, it was inspected and analyzed high and mid-level assistive strategies. The position/torque-based tracking control aims to predefine the target joint trajectory through the gait analysis of limbs movement and/or joints kinematics. Regarding the AAN assistive strategies, these are designed to ensure that the robotic devices only supply as much effort as the patient needs to accomplish training tasks by assessing his/her performance in real-time. These strategies may benefit from a bioinspired hierarchical framework that is based on the human motor control: high-level as an intention's estimator; mid-level translates the intention into the desired state; and low-level tracks the reference state.

Subsequently, low-level assistive strategies that were applied in assistive robotic devices were described and reviewed. Firstly, the PID control was designated as being the most common form of feedback controller, due to its mathematical straightforwardness, simple implementation and tuning, and satisfactory results. Nevertheless, the PID control still has low robustness to disturbances that can affect the controller. Consequently, optimal control was considered as a control that minimizes its performance to an optimal state of the system, assuring resistance against disturbances. In this category, LQR controller diminishes the minimum linear quadratic cost function. However, despite its satisfactory results against disturbances that can affect the system, is a controller that requires deep mathematical knowledge to be implemented. Following, the robust control was stated as a strategy that aims to bound uncertainties, so their results meet the control system requirements in all cases, being prepared to operate in a worst-case scenario. The H^∞ robust control method has the goal of designing a controller which minimizes the influence of disturbances over the considered output of the target system. Although it has a great response against disturbances, it involves considerable mathematical understanding. For last, FEL control was proposed as a strategy that is composed by a feedback controller and a feedforward controller capable of learning the IDM of the system to be controller. It employs regression techniques, preventing knowledge about the controlled model. Being a control that has a learning state, its main drawback is the possibility to take long time periods to settle. Nonetheless, the overall performance of FEL involves adaptive characteristics being suitable to operate under different conditions.

Overall, the chosen low-level controller to implement in SmartOs was the Feedback-Error Learning control. The benefit of not requiring an explicit model of the controlled plant to the feedforward controller and to have a dedicated feedback controller to handle with disturbances are important to assets of FEL. Moreover, its adaptiveness and possible implementation in real-time conditions, withal a better performance than with a stand-alone feedback controller are attractive advantages to implement it in an assistive device.

3 THEORETICAL CONCEPTS

The present chapter intends to clarify some theoretical concepts that are used in this dissertation. The FEL control strategy is detailed, as well as the ANN although not in detail. The preceptor and multilayer perceptron ANN structure and a possible training algorithm to learn the plant’s IDM are herein detailed.

3.1 Feedback-Error Learning

The FEL is a type of control suggested by Kawato [15]. It was thought as a controller that tries to simulate how the central nervous system acquires the internal models of the body (plant in a generic control system) [15]. A FEL controller comprises a feedforward controller that has the IDM of the plant and tracks the reference trajectory and a feedback controller responsible to handle disturbances. Figure 12 illustrates the dynamics of a feedback controller in charge of drawing a circle in this example. The brain represents the controller, the reference movement is a circle, and the plant is the hand with the pencil that receives motor command to draw the desired movement, measured by the eye.

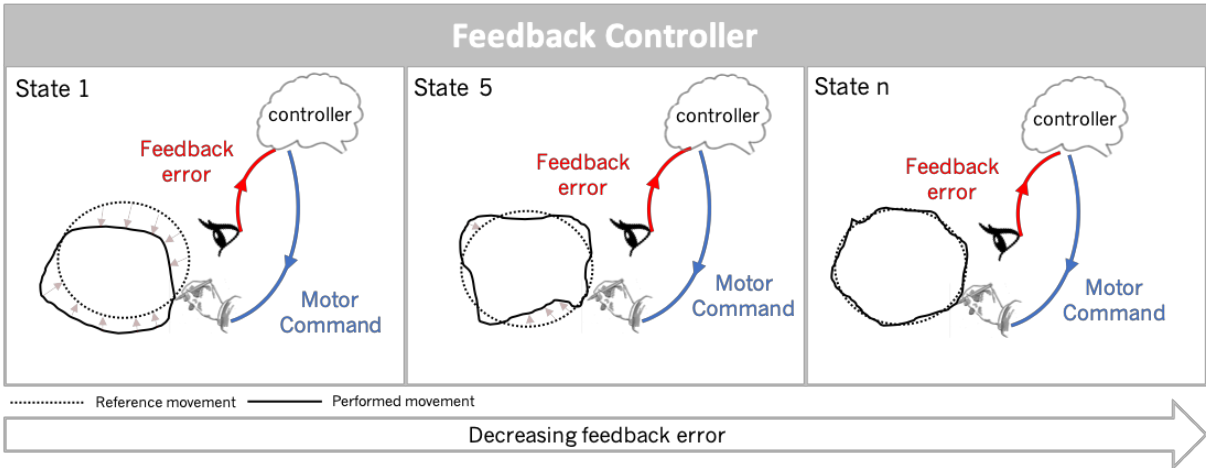


Figure 12 – Illustration of a feedback error controller.

To perform the circle, the controller has to perceive the error between the reference movement and the performed movement (feedback error). Due to a feedback control law, a motor command is executed, trying to decrease the error between the reference and the achieved movement. This feedback loop is executed at each state until the error reaches a stabilized small value (state n).

The feedforward controller is added to the feedback control loop, receiving as input the reference movement to perform a feedforward command that makes the output of the system follow the desired trajectory (draw the circle). The total control command applied to the plant is the sum of the feedback command and the feedforward command, that decreases the feedback error and, consequently, the feedback command. The feedforward controller maps the reference movements into the right motor commands to perform the desired movement. This represents the IDM of the plant. So, the feedforward controller has to learn the IDM of the plant, to apply the right motor commands to the control problem. To accomplish that, the controller has two phases. A training phase to learn the IDM of the plant, receiving the feedback command and the reference movement as inputs and performing a predictive feedforward command that decreases the feedback error and the feedback contribution to the total control command, shown in Figure 13.A. The next phase is achieved when the feedback command has low contribution to the total control command and the feedforward command has the higher contribution. This is the recall phase, that is when the feedforward controller has successfully learned the plant's IDM and only receives as input the reference movement and outputs the right feedforward command, illustrated in Figure 13.B.

If we disconnect the feedback controller from the control scheme, the feedforward controller is capable to perform the control by itself, in consideration that the IDM has a low error – Figure 13.C. However, if disturbances appear, the feedforward controller alone is not capable of correcting them, or even if the reference movement changes (like drawing a smaller circle), since the IDM is not able to provide the correct set of motor commands, and another IDM has to be found. Therefore, the feedforward controller enters again in training phase.

This described behavior represents the FEL controller. A feedforward controller that has the IDM of the plant and performs the right control command to perform the reference

movement, and a feedback controller responsible to provide stability during the learning phase of the feedforward controller and handle with disturbances.

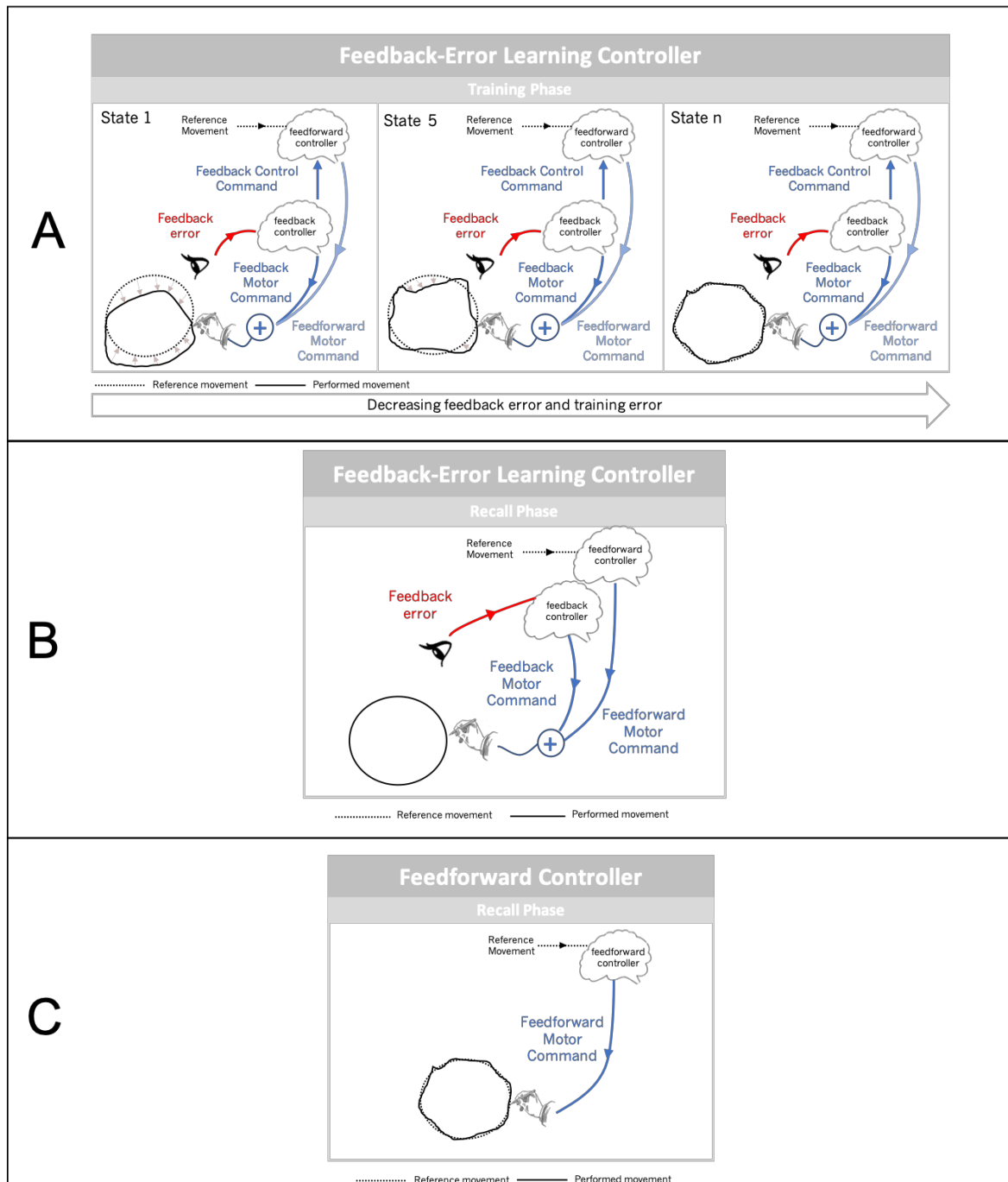


Figure 13 - Illustration of FEL control. A: Training Phase; B: Recall Phase; C: Feedforward in stand-alone.

To learn the plant's IDM, the FEL controller uses regression techniques. Kawato [15], [58] proposed ANN. Although, Radial Basis Functions (RBF), Multilayer Perceptron (MLP) and Cerebellar Model Articulation Controller (CMAC) are other possible structures that ANN can adopt to perform FEL control, since are the most used and indicated to perform function

approximations [74]. Table 10 specifies applications in real scenarios approaching the FEL control in robotic devices, detailing some features of the used ANN.

Table 10 - Applications with FEL control schemes. *Neural Network structure: RBF: Radial Base Functions; MLP: Multi-layer Preceptor.

Name	ANN*	Learning Method	Application
Passold, F et all [75]	RBF	Gradient descent	Scara Robot
Zhao-Hui Jiang [76]	MLP	Gradient descent	AdeptOne XL manipulator
Andon V. Topalov [77]	MLP	Gradient descent	CRS CataLyst-5 industrial manipulator
Francisco Resquín [71]	MLP	Gradient descent	Right Upper-Limb Exoskeleton

3.2 Artificial Neural Network

ANN are computed architectures inspired by biological brains [78], [79]. They are comprised of interconnected neurons/nodes, normally connected on one direction, establishing feedforward links, proceeding from input to output nodes. Every ANN has an input layer and an output layer and in between it can exist hidden layer(s) [80].

Each node has an activation function that is performed on the input data, producing an activation which is passed to the adjacent node(s). Associated to these nodes' connections, are weights that constraint how input data are linked with the activation. These set of weights on an ANN determines its performance through a set of input data and are established during a training phase, where the ANN learns how to tune these weights into a target (received) learning value. This type of training is called supervised learning, as it receives inputs to map to a specific learning signal [80]. Figure 14 illustrates a possible ANN.

When the training phase ends, the ANN is used to classify/predict on new input data. It passes through the input nodes activation functions and weights, forwarding through the ANN, and the ultimate activation of the output nodes until a classification/prediction is made.

Based on their functional characteristic, ANN can have multiple types of structures. As the developed dissertation has its focus on function approximators, the most popular types of adopted structures for this application are: Radial Basis Function (RBF), Cerebellar Model Articulation Controller (CMAC) and Multilayer Perceptron (MLP) [74]. Since the MLP has been reported in several applications of FEL control (**Error! Reference source not found.**) and w

idely applied, is going to be here described. To do so, first, a review of The Perceptron is needed, which is the basic structure of the MLP.

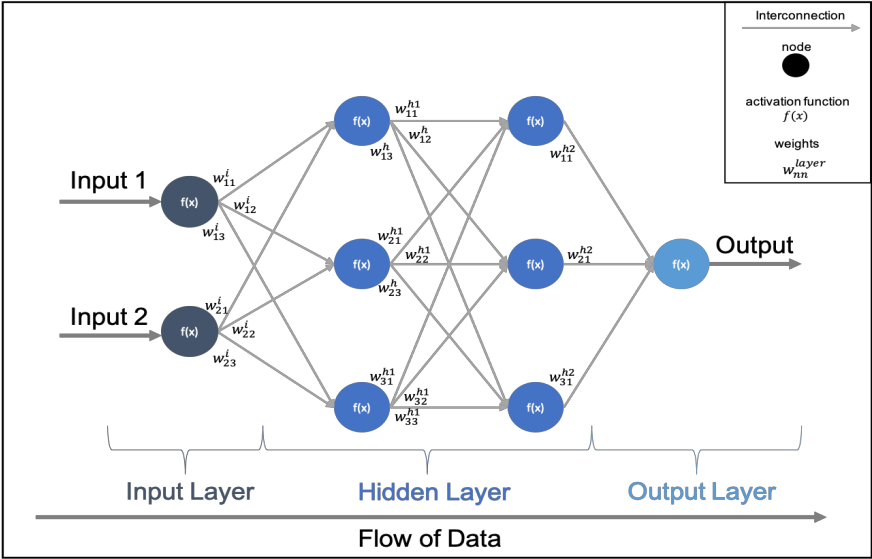


Figure 14 – Usual neural network.

3.2.1 The Perceptron (Single-layer Neural Network)

The perceptron is a type of ANN created by Rosenblatt [81], [79]. This type of ANN only has two layers: 1) the input layer where all inputs are weighed and summed and 2) the output layer that has a binary activation function for its output with a pre-defined threshold that if it

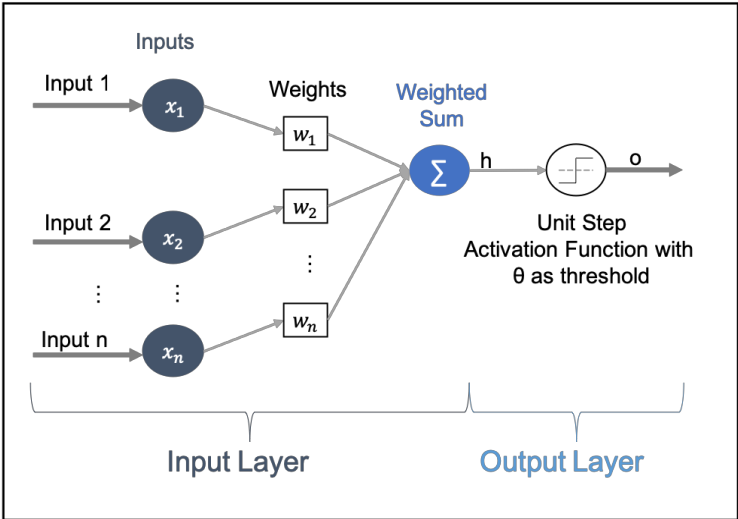


Figure 15 - The perceptron network. It has input nodes, preceded by a weighted sum and binary activation function.

is reached or passed, the output is 1, otherwise it is 0, classifying its output value. Figure 15 illustrates this behavior.

The perceptron training process is iterative and involves the repeated submission of the inputs and related outputs to the ANN (training set). The random assignment of small values to the pre-defined threshold of the output layer as well as for weights of the input layer that composes the ANN should be done first. Then, the following training algorithm is performed:

1. An input is fed forward through the ANN;
2. If the output is correct, then no changes are performed to the thresholds of the binary activation (θ) and the input weights (w_x), and we are back to the 1st step.
3. Otherwise:
 - a. If output was 1 but should be 0, then the following correction is done:

$$\begin{cases} \theta = \theta + 1 \\ w_x = w_x - 1, \text{ if } input_x = 1 \end{cases} \quad (18)$$

- b. If output was 0 but should be 1, the correction is done the following way:

$$\begin{cases} \theta = \theta - 1 \\ w_x = w_x + 1, \text{ if } input_x = 1 \end{cases} \quad (19)$$

All of the training set needs to be presented to the ANN several times to improve its tune to the best possible outcome. In the first iterations through the training set, the ANN might correctly adjust and improve its performance. The next time its efficiency might not increase and eventually it will stop improving. So, it is important to establish a maximum number of iterations whereupon the ANN can train. If it achieves a training state where all the inputs are correct, then it is also time to stop.

The recall phase happens when the ANN has all its weights tuned and it is ready to classify. The perceptron recall ANN algorithm works as follows:

1. Input nodes receive values;
2. The weighted sum is performed, producing the activation h :

$$h = \sum_{i=1}^n w_i x_i \quad (16)$$

3. Classification is performed, based on the unit step activation function $g(h)$:

$$o = g(h) = \begin{cases} 1, & \text{if } h > \theta \\ 0, & \text{if } h \leq \theta \end{cases} \quad (17)$$

where θ is the tuned threshold value.

The perceptron is simply separating the input into 2 categories, those that cause a fire (1), and those that do not (0). This is a very limitative behavior in terms of regression, once it can only solve linear problems and estimate linear models [80]. To solve non-linear problems the multilayer perceptron appears.

3.2.2 Multilayer Perceptron

An ANN that is multilayered, and feedforward includes: 1) an input layer of nodes, 2) one or more intermediate (hidden) layers of nodes, and 3) an output layer of nodes. Depending on the problem, the output layer can involve one or more nodes. In case of regression, it is common to be a single output node that predicts a value, but in case of classification it is normal to have as much output nodes as classes (the predicted class with the highest activation it is probably the correct one). Figure 16 demonstrates an example of a multi-layer perceptron.

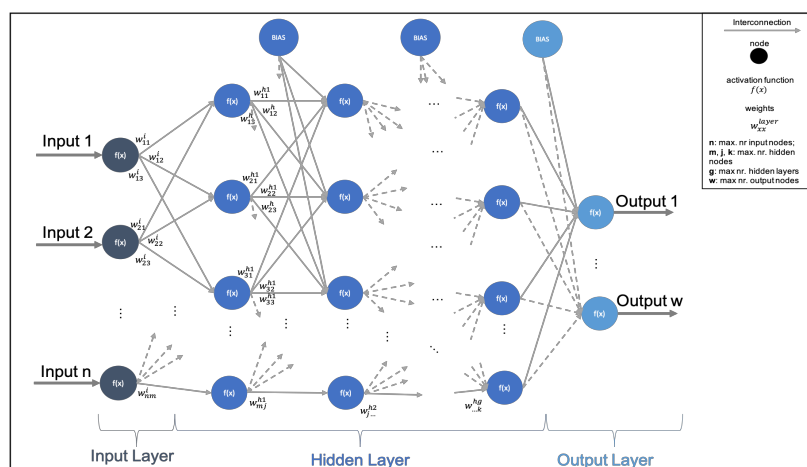


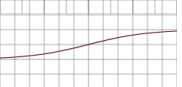
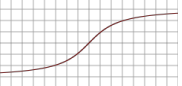
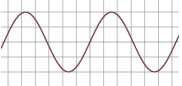


Figure 16 - The Multilayer Perceptron network, consisting of multiple layers of connected neurons.

Figure 16, also present the bias nodes that are added to increase the flexibility of the model to fit the input data. Specifically, it allows the ANN to fire neurons when all input features are equal to 0. They are added to the hidden and output layer [80].

Training the MLP is done in two parts, the forward and backward. The first part aims to predict an output for a given input, based on the current weights' configuration – going forwards through the ANN, as illustrated in Figure 17 – Possible training directions in a Multilayer Perceptron. A. Fed forwarding the ANN with an input is a similar process as the one done in the preceptor. An input neuron receives a value, forwarding it to the adjacent nodes. Each neuron at the hidden layers does a weighted sum with the received values, in accordance with the respective weight value, applies its activation function, and outputs the result to the adjacent nodes. This process is repeated until it reached the output layer. It should be noted that there are an enormous variety of activation functions that could be used on the output of the ANN nodes. Table 11 summarizes some activation functions that could be employed.

Table 11 - Some examples of activation functions.

Function	Equation	Range	Plot
Identity	$f(x) = x$	$]-\infty, +\infty[$	
Binary Step	$f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$	$\{0,1\}$	
Logistic	$f(x) = \frac{1}{1 + e^{-x}}$	$[0,1]$	
Hyperbolic tangent	$f(x) = \tanh(x)$	$[-1,1]$	
Sinusoid	$f(x) = \sin(x)$	$[-1,1]$	

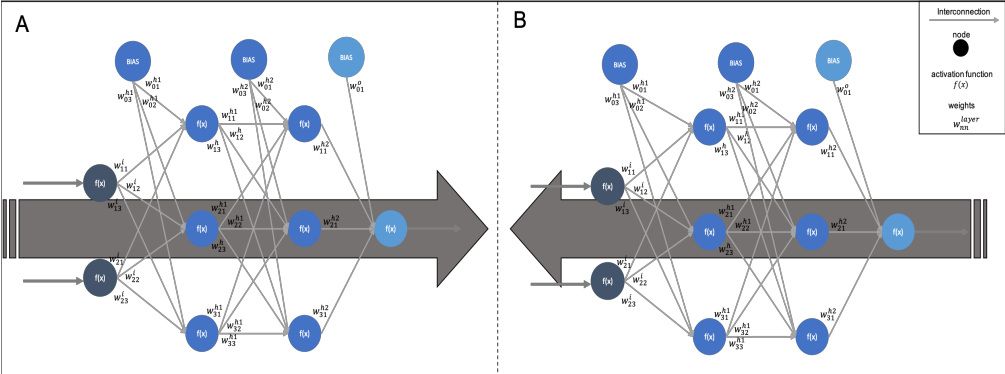


Figure 17 – Possible training directions in a Multilayer Perceptron. A: forwards; B: backwards

Now that we have the outputs of the ANN, we can compare them to the learning signal (targets) and compute the error. This is done in the second part of the training process, where the weights are updated according to the error difference between the predicted output and the learning signal (target output) – going backwards through the ANN as Figure 17.B shows.

The common method to train a MLP ANN is the Backpropagation algorithm. Hereby, the error is sent backwards through the ANN, applying gradient descend techniques [80], in order to find the set of ANN’s weights that minimize the error between the predicted and the target output. In other words, the weights have to be adjusted so the neurons fire only when the targets allows them and working out which weights caused the error. The used error function in backpropagation can be the sum of squares, which finds the difference between the predicted output y , and the target output t , for every output neuron N , as Equation (18) states (mean square).

$$E(t, y) = \frac{1}{2} \sum_{k=1}^N (y_k - t_k)^2 \tag{18}$$

Differentiating Equation (18), the error’s gradient and the increasing/decreasing direction is found. Since the purpose is to minimize the error, the direction along which the error decreases is followed, i.e., where its gradient is descendent. Figure 18 illustrates this behavior.

Thus, the ANN’s weights are updated in accordance with the error gradient descent with respect to the weights, as described in Equation (19).

$$w_{ij}^t = w_{ij}^{t-1} - \frac{\delta E}{\delta w_{ij}} \tag{19}$$

where E is the error value, w , the weights, t indicates the current update, $t - 1$ the previous one and ij is the performed connection that identifies the neurons.

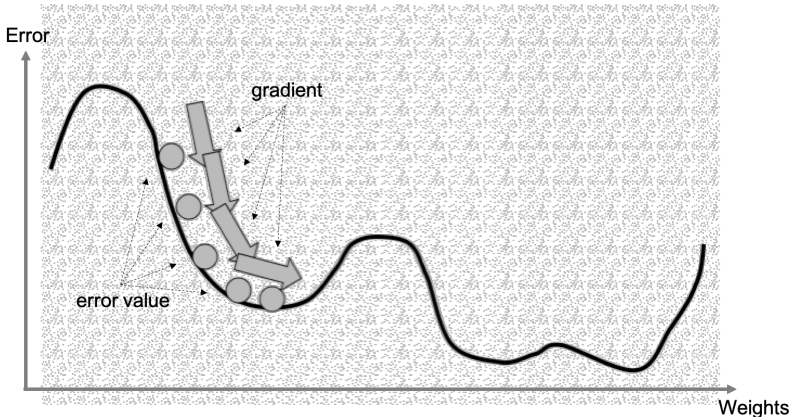


Figure 18 - Gradient descent error such that the weights change. Adapted from [80].

Observing Figure 18, we realize that the gradient descent algorithm could be stuck in a local minimum, because the error gradient leads us there, misleading us from finding a lower error. Thus, it is important to introduce some parameters to Equation (19) that change how the weights are updated, as presented in Equation (20).

$$w_{ij}^t = w_{ij}^{t-1} - \eta \frac{\delta E}{\delta w_{ij}} - \alpha \Delta w_{ij}^{t-1} \tag{20}$$

First, it is introduced the learning rate η , that controls how much influence the error gradient has in the weights update. Figure 19.A clarifies the learning rate importance.

If the learning rate has large values, then the weights update is also high, and the ANN converges faster to the minimum. However, it could lead the ANN to overshoot, missing the minimum error. On the other hand, small values of learning rate can cause small but precise weight updates, causing the ANN to spend many iterations until reaching converge and risking being trapped in a local minimum. So, the learning rate parameter should be carefully tuned to avoid these situations.

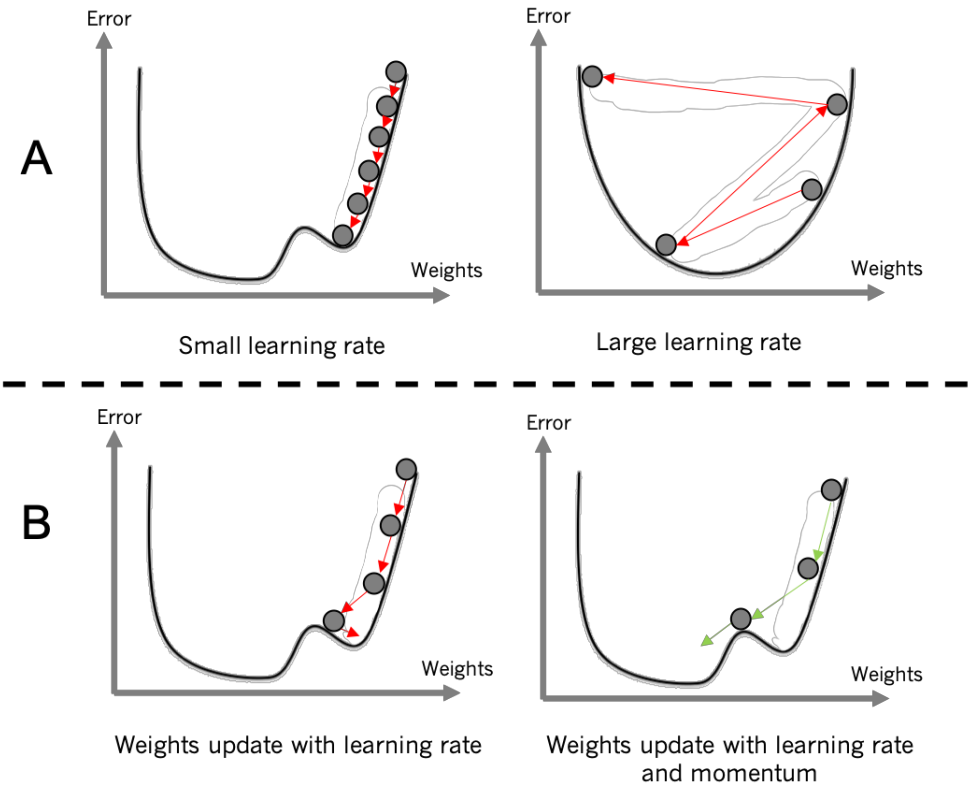


Figure 19 – Visual influence of different ANN parameters. A: learning rate parameter; B: momentum and learning rate parameter.

The momentum parameter α , appears alongside with the previous weight update $\alpha\Delta w_{ij}^{t-1}$, adding a percentage of the last update to the current one. In Figure 19.B an illustration of its behavior is shown.

Essentially, the momentum parameter improves the gradient descent to escape local minima, adding some contribution from the previous weight update. We can also use small learning rate values, making possible a more stable convergence of the error.

Concluding, the backpropagation algorithm calculates the errors' gradient with respect to the weights, so the weights can be changed according to the decrease of the error (gradient descent). To do so, a differentiation of the error function with respect to the weights is done. This is applied backwards through the ANN, starting at the outputs and ending up back at the inputs. In order to the MLP ANN successfully converges during the training phase, it is important to find the right balance of values for the learning rate and momentum parameters.

3.2.3 The Multi-Layer Perceptron Algorithm

Before presenting the proposed an algorithm for the MLP ANN, some considerations must be done first. First, it is assumed that the ANN has 3 layers with L input nodes (plus the bias), M hidden nodes (plus the bias) and N output nodes, respectively. To index the nodes in each layer, it is used i, j, k for the input, hidden and output, respectively. The input-hidden weights are labeled with v and have the hyperbolic tangent as activation function, labeled as g_j . The hidden-output weights are labeled with w and use the identity as the activation function, labeled as g_k . The gradient descent algorithm is stochastic, i.e., for each learning value t_k , an update to the weight's parameter is performed and inputs are presented in random order. This deviates the algorithm from redundancy over the dataset, converging faster to a local minimum and enabling an online implementation [82].

Figure 20 shows proposed algorithm for the gradient descent. It starts with the initialization of the ANN's weights. During training, it receives data and forwards it through the ANN to make a prediction (y_k) – Forward Phase. This is compared with the target value to generate an error value (δ_k). This error is backward it through the ANN, by applying the backpropagation algorithm so the weights are updated in the gradient descent – Backwards Phase. At specific iterations the performance of the ANN is evaluated to decide if it can stop

the training and enter in the recall mode, where only predictions are performed according the received input.

Initialization:

- All weights must be initialized to small random values;

Training is repeated for each input x_i , and learning value t_k :

Forward Phase

- Compute the activation a_j of each neuron j in the hidden layer(s):

$$h(j) = \sum_{i=0}^L v_j x_i \quad (21)$$

$$a_j = g(h(j)) = \tanh(h(j)) \quad (22)$$

- Compute the activation a_k of each ANN k in the hidden layer(s) producing an output y_k (predicted output):

$$h(k) = \sum_{j=0}^M w_k a_j \quad (23)$$

$$y_k(k) = a_k = g(h(k)) = h(k) \quad (24)$$

Backwards Phase

- Compute the error at the output using:

$$\delta_k(k) = y_k(k) - t_k \quad (25)$$

- Compute the error in the hidden layer using:

$$\delta_j(j) = (1 - y_k(j))^2 \sum_{k=0}^N w_k \delta_k(k) \quad (26)$$

- Update the output layer weights using:

$$w_k = w_k + \eta \delta_k(k) a_k + \alpha \Delta w_k^{t-1} \quad (27)$$

- Update the hidden layer weights using:

$$w_j = w_j + \eta \delta_j(j) x_i + \alpha \Delta w_j^{t-1} \quad (28)$$

Recall:

Figure 20 - Proposed algorithm for the gradient descent.

3.2.4 Gradient Descent Optimizer

The gradient descent algorithm can offer some challenges to a good ANN convergence, such as: 1) with frequent weight updates, error can overshoot, complicating convergence to an exact minimum; 2) Setting the learning rate parameter to a good value can be an arduous task; 3) Only one learning rate and momentum is applied to all weights, restricting the same weight change to all ANN making convergence less efficient; To overcome these limitations, some gradient descent optimization algorithms have been developed, that can enhance the algorithm convergence performance. In [82] a good overview of many optimizers is performed. The Adaptive Moment Estimation (ADAM) [83] optimizer was chosen to be herein described. The ADAM optimizer adds the following equations to the algorithm:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \end{aligned} \quad (29)$$

where t is the current update, v_t is an average of exponential decay of past squared gradients g_t^2 and m_t is an average of exponential decay of past gradients g_t , identical to momentum. m_t parameter makes the algorithm to search for flat minima in the error surface and v_t enables to reduce the initial learning rate in order to update the weights in a more precise way. β_1 and β_2 are constants decay rates. To avoid m_t and v_t from bias towards zero value, Equation (30) is also added.

$$\begin{aligned} \hat{m}_t &= \frac{m_t}{1 - \beta_1^t} \\ \hat{v}_t &= \frac{v_t}{1 - \beta_2^t} \end{aligned} \quad (30)$$

With this, ADAM updates the ANN weights (here generic represented as w) in accordance with Equation (32):

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t + \varepsilon}} \hat{m}_t \quad (32)$$

where the constant ε is added to avoid zero division.

Wrapping up, the ADAM algorithm has adaptive learning rates per weights parameters, that separately adapts as learning unfolds, instead of having a global learning rate [83]. With

this, versatility is given to the ANN, increasing the speed of convergence and decreasing the learning error.

4 SMARTOS: SMART CONTROL OF A STAND-ALONE ACTIVE ORTHOTIC SYSTEM

The current chapter addresses the SmartOs, a system that is in progress and being developed by several students. First, a general overview about the system is presented, stating its operating mechanism. Second, a general system description is made, describing the SmartOs modules and their time constraints. Finally, the contribution of this dissertation to SmartOs is presented, which focused on the validation of the temporal requirements of SmartOs' modules. A validation protocol was developed and implemented to ensure SmartOs time-effectiveness.

4.1 General Overview

SmartOs is a modular wearable robotic gait technology, that acts and cooperates closely with the human being, by means of a powered lower-limb orthosis, synergistically linked to a wearable laboratory for motion analysis. It intends to provide not only customized rehabilitative tasks tailored to the patient needs, but also to foster objective diagnosis of the patient's motor conditions. SmartOs was thought as a safe and reliable system with contributions not only from engineers but with all significance parts, from physical medicine and rehabilitation physicians to the patients, being currently in first-phase of validation, in *Hospital de Braga* with stroke survivors. Its applications can go from motor rehabilitation in hospitals, to rehabilitation centers, nursing homes or other clinical units.

To accomplish these goals, SmartOs comprises a/an:

- **Wearable Motion Lab**, aiming to monitor in real-time the patient's motor status (i.e., the user's motor ability, and spasticity level) and the human-orthosis interaction. This wearable motion lab includes ergonomic, stand-alone, wearable sensory devices. Wearable biofeedback mechanisms (e.g., time-discrete feedback provided through a vibrotactile system - Waistband) to encourage the user's active

participation and to seamless the bidirectional interaction between the user and the orthotic system.

- **Smart hierarchical control architecture**, bioinspired on the principles and organization of the human motion-control system [22]. Assistive commands according to the user's motion capacity and disability level (monitored by the wearable motion lab) are generated by this architecture, which then are interpreted by **adaptive learning technologies**. Thus, it is possible to: i) timely recognize and predict the intended motion; ii) to detect incipient falls and the user's impaired level; iii) to segment gait cycle; iv) to automatically tailor the dynamic behavior of the human-orthosis interface.
- **Assist-as-need personalized technological solution** for the robotic-based gait training. The orthotic system delivers a task-oriented and repetitive gait training according to the user's needs, fostering a recovery of the functional lower-limb motor abilities, thus improving an abnormal gait.

Therefore, SmartOs experiences the same requirements established for ambulatory rehabilitative orthotic systems: wearable, adjustable, modular, flexible, effective actuation, and time-power effectiveness technology. Figure 21 illustrates the SmartOs operating mechanism.

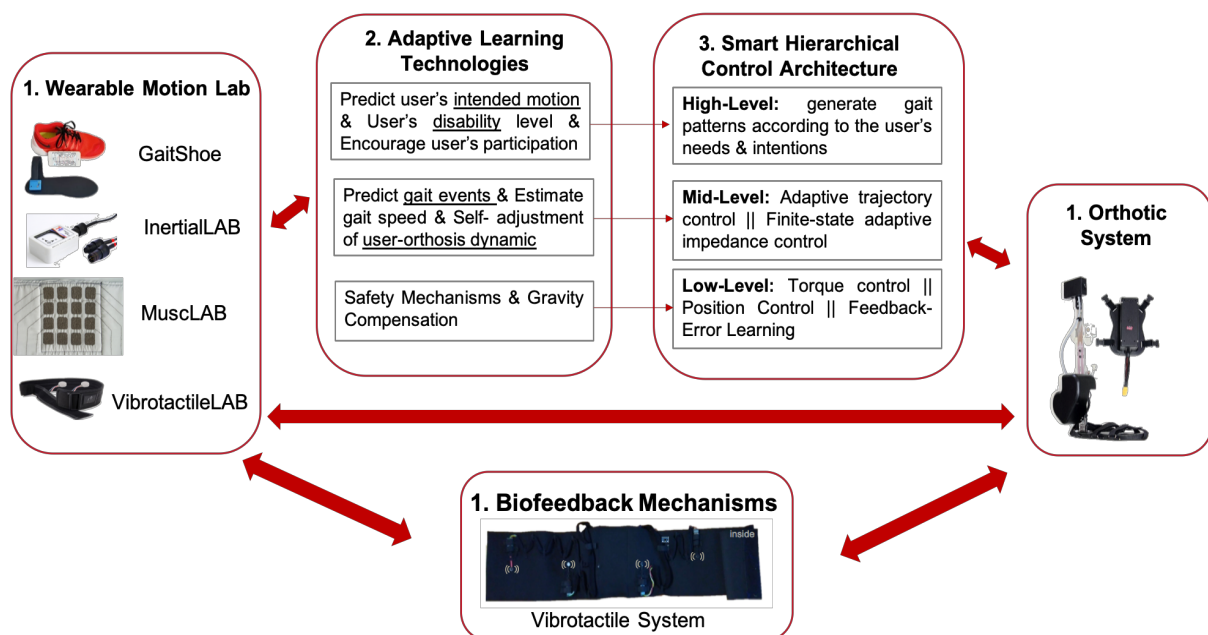


Figure 21 - Operating mechanism of SmartOs system.

SmartOs integrates work developed by several Masters students, namely: design and architecture implementation; graphical user interfaces for the systems set-up and display/log of the recorded data; technologies to monitor and analyze the gait behavior; development of assistive motion strategies. As referred in Chapter 1, one goal of this thesis is to incorporate a low-level assistive strategy in SmartOs. To accomplish this purpose, it was necessary to properly validate the system as a whole, i.e., to guarantee that real-time operation requirements are fulfilled.

4.2 SmartOs Description

The current sub-chapter is focused on the design and technical description of SmartOs. With respect to this, an overview is presented in Figure 22.

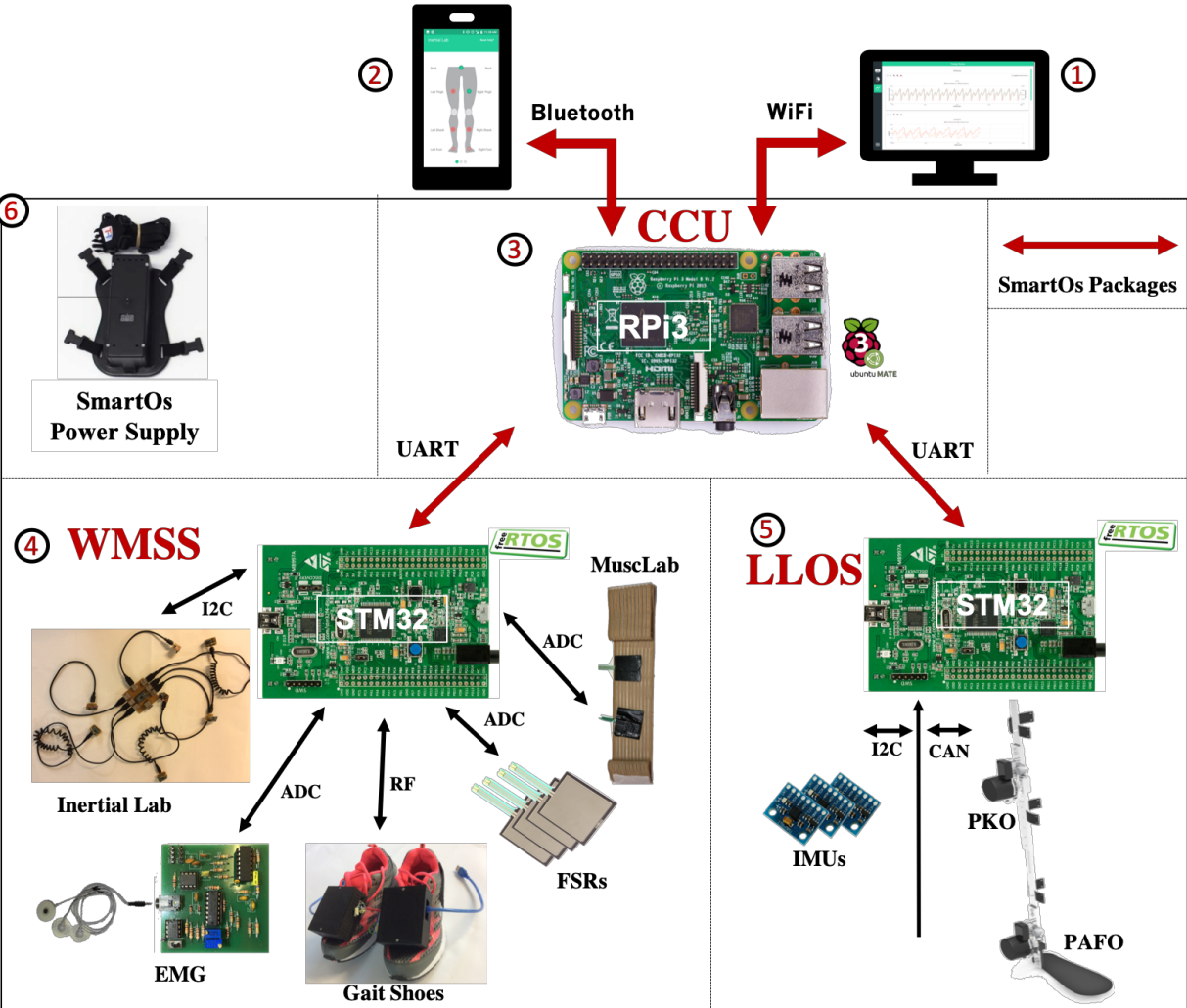


Figure 22 - SmartOs: System Architecture.

In Figure 22, it is disclosed the system main components, used communication interfaces and the system architecture. Considering effectiveness in the actuation, real-time constraints, performance, modularity, portability, and safety, a decentralized architecture was used, being composed by central control unit, for high-level reasoning methods, and several control units, to handle mid/low-level control of the actuation devices and sensory systems. As depicted in Figure 22, the architecture of SmartOs is mainly composed by 6 modules:

1. **Desktop graphical user application** for real-time monitoring of therapies and backup of collected data in database;
2. **Android OS mobile application**, for management and set-up of all modules defined in SmartOs. This application allows configurability to the system for different subjects and therapies;
3. **Central Control Unit (CCU)** of SmartOs, which interfaces all SmartOs with the applications (external devices). This unit is composed by a Raspberry Pi 3 [84], running the Ubuntu Mate OS [85], and implements the high-level reasoning methods (smart control strategies based on hierarchical control framework), as well as the communication with external devices, such as, Wearable Multi-Modular Sensor System and Low-Level Orthoses System, and two graphical interfaces, i.e., a desktop and mobile application;
4. **Wearable Multi-Modular Sensor System (WMSS)**, which interfaces: i) an InertialLAB with Inertial Measurement Units (IMU) at each lower-limb segment, responsible to estimate the lower-limb joint angles, using Inter-Integrated Circuit (I2C) protocol to transfer data; ii) wired multi-channel Electromyography (EMG) acquisition system through Analog to Digital Converters (ADC) to get the muscle activation; iii) two wireless Gait Shoes through radio frequency (RF) to acquire gait events and speed; iv) four Force Sensor Resistors (FSR) for gait validation, through ADCs; v) MuscLAB to estimate muscle activity through FSRs. This unit is centered in a STM32F4-Discovery board [86], running FreeRTOS OS [87], and it is capable of providing an accurate and time-effective acquisition of the sensors, running the necessary algorithms and an efficiently external communicate;
5. **Low-Level Orthoses System (LLOS)**, designed to interface two wearable H2 joints (ankle-foot and knee) through Connection Area Network (CAN) protocol, as well as

wearable inertial sensors, in particular, three IMUs through I2C. This unit is centered in a STM32F4-Discovery board as well, running FreeRTOS OS, and it will be responsible for: i) collecting the information from the orthotic devices and sensors (and redirect it to the CCU); ii) run dedicated mid/low-level controllers of motion assistance strategies based on the sensory feedback provided and reference set-points received externally; iii) run mechanisms to lighten the weight of the powered devices (using the hip angle estimated through one IMU attached to the thigh). Moreover, LLOS implements an efficient external communication system;

6. **Power supply system**, based on a LifePO4 battery and additional electronic, able to power up all components of SmartOs.

Moreover, Figure 22 also highlights the main communication structures of SmartOs. Regarding the communication between the CCU with LLOS and WMSS, a serial interface is implemented. In fact, Future Technology Devices International converters are used, as a 'bridge' between the Universal Asynchronous Receiver-Transmitter (UART) and Universal Serial Bus (USB) protocols. Furthermore, the communication between CCU and the mobile and desktop application are wirelessly performed, by means of Bluetooth and WiFi protocols, respectively. Both protocols are built-in the Raspberry Pi 3, reducing the hardware resources for the CCU.

The communication with/from CCU is only possible through specific SmartOs packages. These packages were defined to exchange data between the modules, with efficiency and in a simple way. Were defined three types of SmartOs packages: i) control packages to set-up the different SmartOs modules; ii) data packages containing values from the sensors/controls/algorithms of the different modules; iii) status packages briefing about the executed commands.

Lastly, to ensure that all of SmartOs time requirements were met, multiple resources of the selected development boards were used, for instance, timers, threads, I2C, UART, CAN, ADC, etc.

Table 12 reviews the time demands for WMSS, LLOS, and CCU.

Table 12 - SmartOs time constraints. *position data communication: data packages with reference positions for the orthosis to

SmartOs Modules	Routines of the SmartOs modules	Used Resource	Required Frequency
WMSS	WMSS → CCU (Data Packages)	FreeRTOS thread	100 Hz
	InertialLAB IMUs acquisition and joint angle estimation	Timer ISR	100 Hz
	EMGs acquisition	Timer ISR	500 Hz
	FSRs acquisition	Timer ISR	100 Hz
	MusCLAB. Acquisition and muscle activation	Timer ISR	100 Hz
LLOS	LLOS → CCU (Data Packages)	FreeRTOS thread	100 Hz
	Low-level control loop	Timer ISR	1 kHz
	Mid-level control loop	Timer ISR	100 Hz
	Gait Event Prediction (GEP)	Timer ISR	100 HZ
	IMUs Acquisition and knee/ankle joint angle estimation	Timer ISR	100 Hz
CCU	CCU → LLOS (Data Packages) *	Linux timer	{0.29, 0.32, 0.36, 0.41} Hz
	CCU ↔ Mobile App (Control/Status Packages)	Bluetooth Socket	Asynchronous
	CCU ↔ Desktop App (Data Packages)	WiFi Socket	100 Hz

4.3 SmartOs Validation

4.3.1 Initial Considerations

The main contribution of the developed work in this dissertation regarding SmartOs was the integration of a low-level control. In order to add this control to the LLOS, a time requirement check of the whole system was imperative, once each individual module was already validated as stand-alone, both in terms of expected results and execution times. This validation is crucial to ensure a real-time and effective operation of SmartOs as a gait rehabilitation and monitorization system. The time-effectiveness was validated considering the signal sampling requirements, control loop frequencies, actuation response of orthoses, and the efficiency of development boards to deal with implemented algorithms.

4.3.2 Validation Protocol

A validation protocol was implemented, in order to guarantee the time effectiveness of SmartOs. Hereafter, a thorough validation respecting the time requirements of SmartOs were accomplished given the relevance of the real-time of the SmartOs application. For each SmartOs module. Figure 23 shown the taken approach to validate the time requirements of WMSS and LLOS modules.

First, the WMSS was validated enabling all available routines depicted in Table 12, for this module. To inspect if the required frequency was met, an available port of the module's development board was set to every routine. That port was toggled between high (1) and low (0) every time the routine was run, generating a square wave that was displayed in a digital oscilloscope. The high and low time of the displayed square wave must match with the required frequency assigned in Table 12. Otherwise, the priority of the inspected routine timer/task should be adjusted. Next the LLOS was validated, following the same protocol, finishing the validation with the two modules running all its routines, i.e., the WMSS and LLOS. Each SmartOs module, has its own timer responsible to handle with its routines. For instance, LLOS' control has a timer responsible to decide if it is time to run low-level control, or mid-level control or to build a SmartOs data package (monitoring the numbers of times that the timer has overflowed).

All SmartOs modules follow the explained logic, so to validate if each routine was in accordance with the stipulated time constraint, the algorithm in Figure 23 was followed.

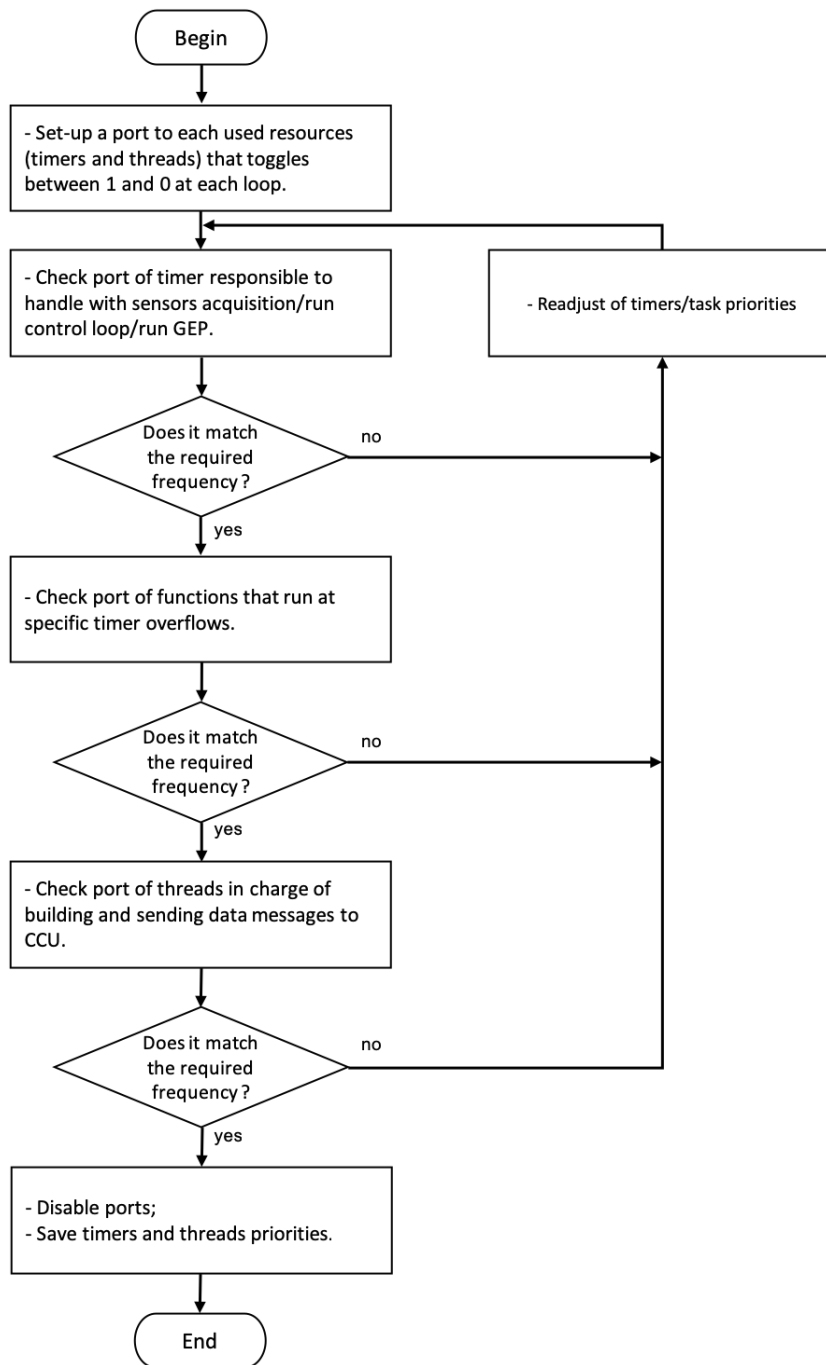


Figure 23 - Flowchart illustration of used validation protocol to ensure a real-time behavior.

4.3.3 Validation Results

Concerning the validation of the timer Interrupt Service Routine (ISR) of WMSS, Figure 24 shows the obtained results. The following frequency results were obtained: a) InertialLAB = 100 Hz (expected 100Hz); b) EMG = 500 Hz (expected 500 Hz); c) FSRs = 100 Hz (expected 100 Hz); d) MuscLAB = 100 Hz (expected 100 Hz).

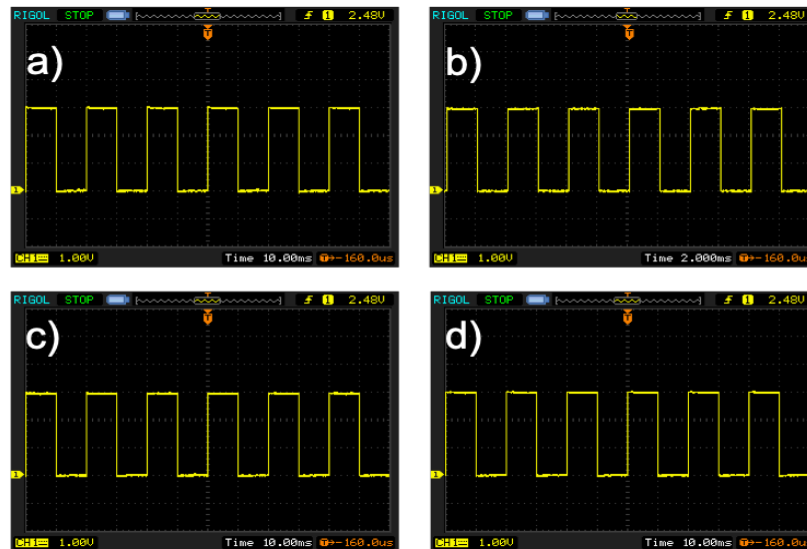


Figure 24 – Time behavior of WMSS acquisition functions: a) InertialLAB, time division = 10 ms; b) EMG, time division = 2 ms; FSRs, time division = 10 ms; MuscLAB, time division = 10 ms.

Regarding the SmartOs data packages exchange between WMSS and CCU data packages the temporal results are observed in Figure 25.

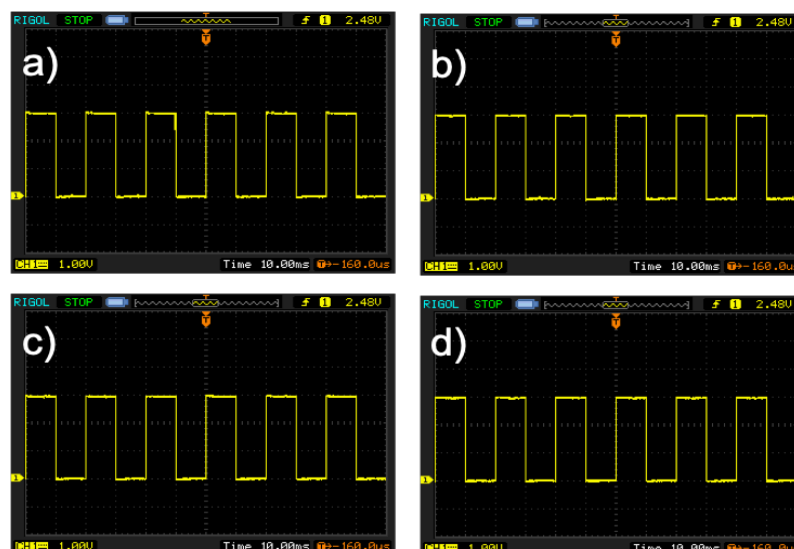


Figure 25 - Time behavior of WMSS data packages procedure: a) InertialLAB, time division = 10 ms; b) EMG, time division = 10 ms; FSRs, time division = 10 ms; MuscLAB, time division = 10 ms

The following frequency results were obtained: a) InertialLab = 100 Hz (expected 100Hz); b) EMG = 100 Hz (expected 100 Hz); c) FSRs = 100 Hz (expected 100 Hz); d) MuscLab = 100 Hz (expected 100 Hz).

Figure 26 shows the results for the timer ISR of LLOS. The following frequency results were obtained: a) Low-level control = 1 kHz (expected 1 kHz); b) Mid-level control = 100 Hz (expected 100 Hz); c) Gait Event Prediction (GEP) = 100 Hz (expected 100 Hz); d) IMUs Acquisition = 100 Hz (expected 100 Hz).

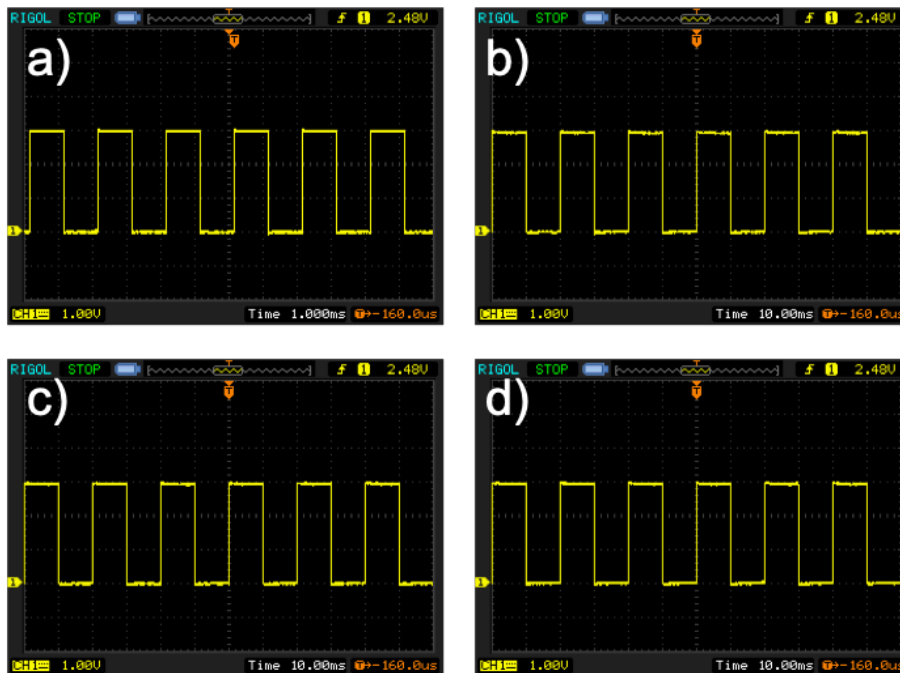


Figure 26 - Time behavior of LLOS functions: a) Low-level control, time division = 1 ms; b) Middle-level control, time division = 10 ms; c) Gait Event Prediction, time division = 10 ms; d) IMUs Acquisition, time division = 10 ms.

Figure 27 shows the obtained results for the SmartOs data packages exchange between LLOS and CCU.

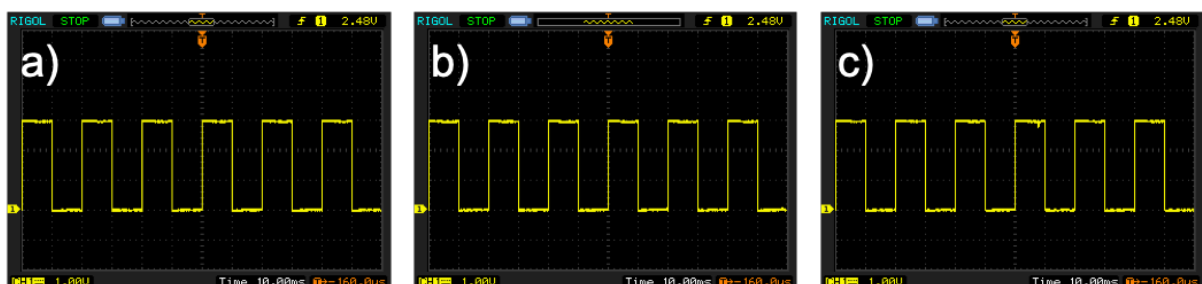


Figure 27 - Time behavior of LLOS data packages procedure: a) Control strategies, time division = 10 ms; b) IMUs, time division = 10 ms; Orthotic Sensors, time division = 10 ms.

The following frequency results were obtained: a) Control Strategies = 100 Hz (expected 100Hz); b) IMUs = 100 Hz (expected 100 Hz); c) Orthotic sensors = 100 Hz (expected 100 Hz).

After several adjustments of timers and threads priorities, all obtained results met the temporal requirements, ensuring a real-time feature for SmartOs. Thus, the integration of new methods in SmartOs, both for WMSS and LLOS, must not jeopardize the implemented architecture. A time requirement identification for the additional modules should be performed in order to run the above validation protocol, so the effectiveness of SmartOs could be maintained.

4.4 Conclusions

Within this chapter, SmartOs was stated as a modular wearable robotic gait system, capable of providing personalized rehabilitative tasks to the user necessity and an objective diagnosis of the user's motor conditions. Its applications can go from motor rehabilitation in hospitals, to rehabilitation centers, nursing homes or other clinical units.

A technical description of the SmartOs system was covered. The system was presented, in terms of its modules (LLOS, MWSS, CCU) and external devices, such as mobile application, and desktop application. The presented description intends to provide a general overview of the developed system, highlining its time requirements for a real-time behavior.

Lastly, a validation of the temporal requirements of SmartOs was presented, considering the system as a whole. It has been considered: signal sampling requirements; control loop frequencies and efficiency of development boards to deal with implemented algorithms. This validation was crucial to ensure a real-time and effective operation of SmartOs as a gait rehabilitation and monitorization system. With the obtained results, it was validated the real-time performance of SmartOs.

5 FEEDBACK-ERROR-LEARNING: LOW-LEVEL CONTROL

This chapter presents the methods, validation and results of the real-time implementation of the FEL controller in SmartOs. The used assistive strategy is explained, with detail to the used high- and mid-level controls. The methods to implement in real-time the FEL controller and validation procedures are specified. Results of the FEL controller implementation are shown and discussed.

5.1 Methods

The real-time FEL controller was implemented in SmartOs, more specifically, in the LLOS module. Therefore, all the developed algorithms for the real-time implementation were done in C language in the STM32F4 discovery development board, using its hardware abstraction

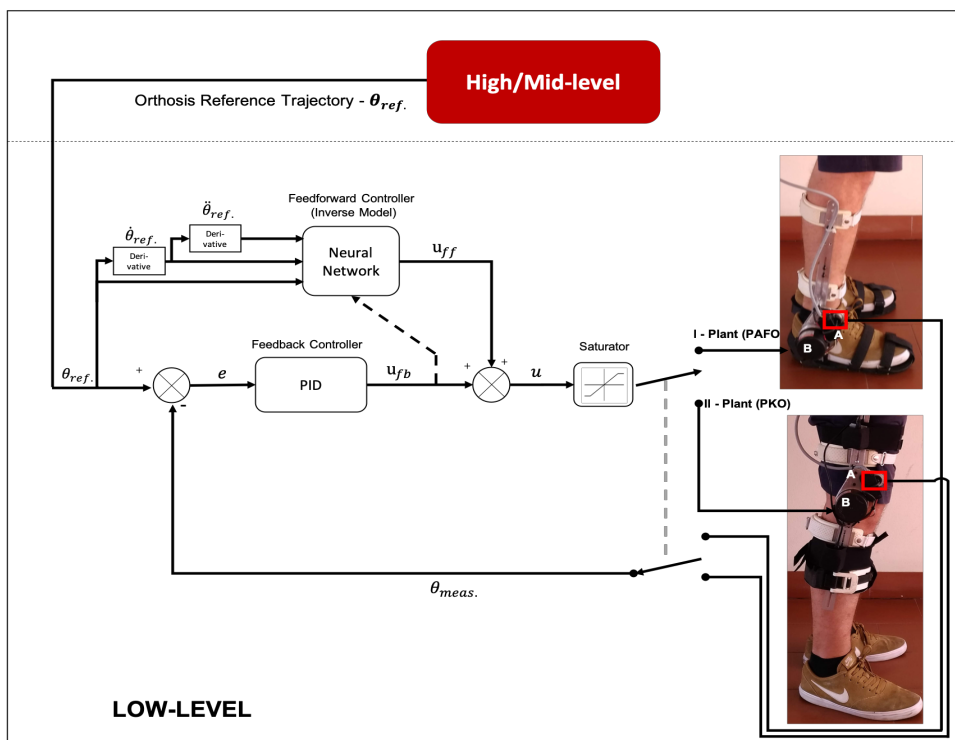


Figure 28 - FEL real-time control loop. I: plant as PAFO. II: plant as PKO. $\theta_{ref.}$ is the reference angular position; $\dot{\theta}_{ref.}$ is the reference angular speed; $\ddot{\theta}_{ref.}$ is the reference angular acceleration $\theta_{meas.}$ is the measured angular position; e is the position error; u_{fb} is the feedback command; u_{ff} is the feedforward command; u is the total control command; A is the potentiometer and B is the actuator.

layer (HAL) libraries and following the modular and time-effectiveness architecture of SmartOs. Figure 28 shows the implemented control loop.

The PKO and PAFO were the assistive devices to be controlled. For the high-level and mid-level, a position-based tracking control assistive strategy was selected to set the user-oriented trajectory and the orthosis reference trajectory, respectively.

5.1.1 High-/Mid-level Control: Position-based Tracking Control

The position-based tracking control presented in Figure 29 generates reference joint angular trajectories and it is designed for therapies that ensure repetitive movements of the user's limbs, being suitable to improve movement coordination.

The high-level of the position-based tracking control uses a regression model with splines to produce user-oriented trajectories (θ_{user}) with 48 points, parameterized in accordance with the user's gait speed and height.

The mid-level applies Equation (33) to output the orthosis reference trajectory ($\theta_{ref.}$), specified by the user-oriented reference trajectory and the Number of Control loops (NOC). Equation (34) describes the NOC that ensures a speed-dependent gait trajectory, found empirically to set the necessary number of control loops that a single reference trajectory has to be available for the low-level control, so the selected gait speed is assured.

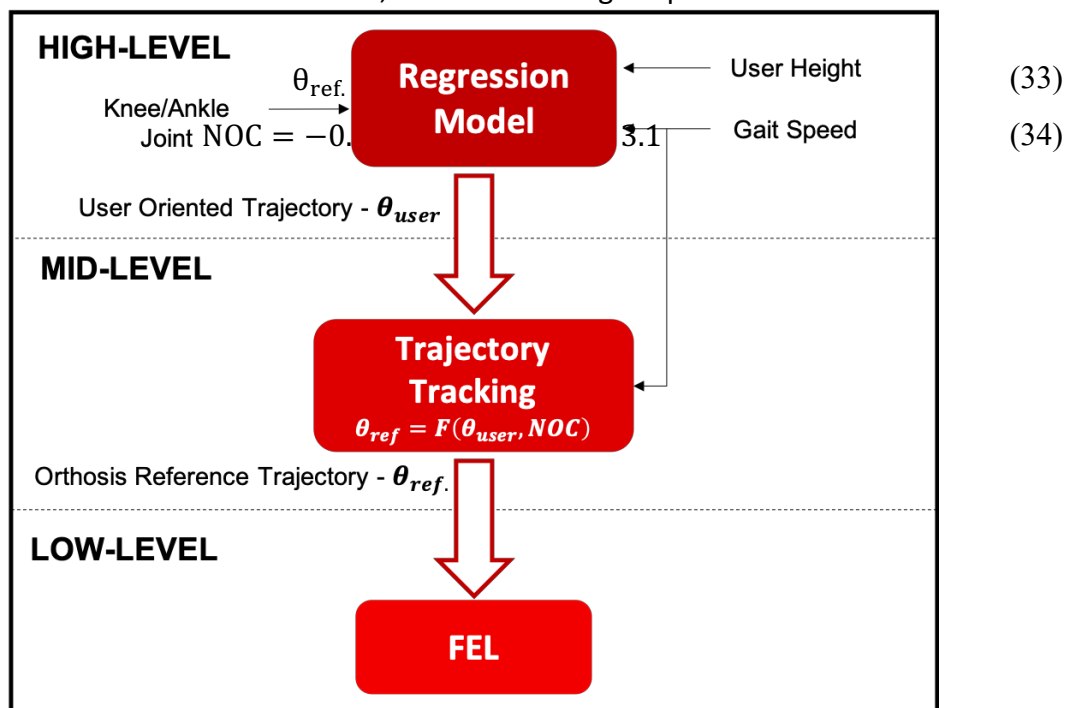


Figure 29 - Interaction between high-level, mid-level and low-level control hierarchy.

In accordance with the selected gait speed, user's height, and lower limb joint (i.e., knee or ankle), the high-level and mid-level supply the low-level FEL control with the correct trajectory to follow. Depending on the selected gait speed, the orthosis reference trajectories are generated at a frequency that varies from 10.5 Hz to 19.6 Hz, which corresponds to a selected gait speed of 0.5 km/h to 1.6 km/h, respectively. Figure 30 illustrates 1 period of the supplied trajectory for the PKO and PAFO, respectively, for a gait speed of 1 km/h.

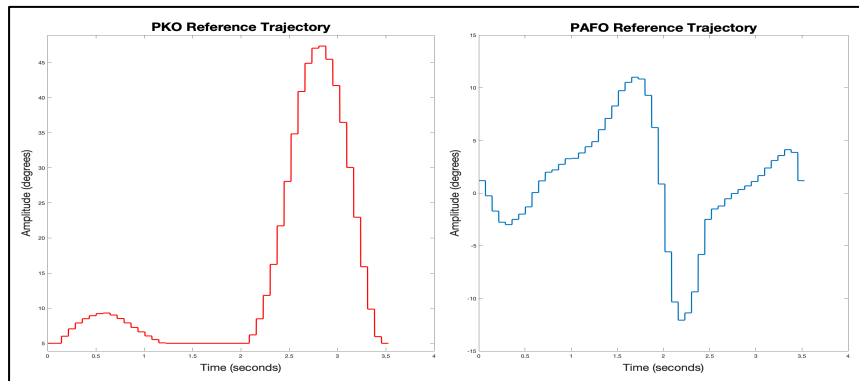


Figure 30 - Generated knee and ankle trajectory by the high-level.

5.1.2 Low-Level Control: Feedback-Error Learning Control

The FEL controller was implemented at the low-level of the hierarchical control architecture for controlling the PKO and PAFO, independently. As can be seen in Figure 28, the FEL control scheme implies a feedback and a feedforward controller. Thus, both the feedback controller and the feedforward controller had to be tuned, so the orthosis could properly be commanded, at a chosen frequency of 1 kHz.

5.1.2.1 Feedback Controller

The interaction between the orthosis and the user's limb should be as smooth as possible, avoiding abrupt movements that can cause discomfort and/or instability to the user. For this reason, the applied control should avoid oscillations and overshoots in the actuator's response. This behavior was achieved through the correct tuning of a PID controller by means of the Ziegler-Nichols method, detailed in [88]. Equation (35) implements a PID controller:

$$u_{fb} = K_p e_k + K_i \sum_{n=1}^k e_n \Delta t + K_d \frac{e_k - e_{k-1}}{\Delta t} \quad (35)$$

where, e_k and e_{k-1} correspond to the current and previous error between the reference and measured angular position, respectively. Concerning the practical application in a rehabilitation scenario, we used $K_p=90$ and $K_I=K_D=1.5$ as PID gains. Figure 31 illustrates the PID controller for the PAFO and PKO, when controlled with position-based tracking control strategy.

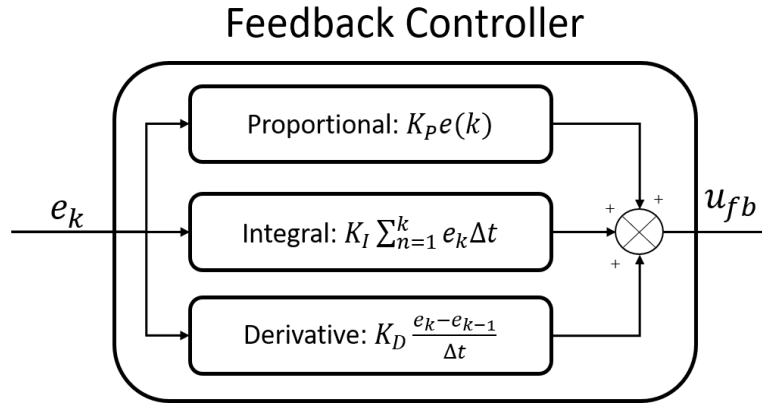


Figure 31 - PID as feedback controller.

5.1.2.2 Feedforward controller

The feedforward controller aims to learn the IDM of the plant, namely, the PKO or PAFO. A supervised training with ANN was used per orthosis due to its proper estimation performance, good generalization, and capabilities to map non-linearities [74]. The ANN has 3 neurons in the input layer since the ANN receives the reference position, speed and acceleration as inputs; 4 and 5 neurons in the hidden layer for the ANN of the PKO and the PAFO, and the output layer has 1 neuron since only a feedforward command is needed as output. In the hidden layer, the hyperbolic tangent activation function is used, due to its output's values in $[-1;1]$ range and for the output layer the identity function is applied since it generates values in $[-\infty; +\infty]$ range.

The MLP was the adopted structure for the ANN since it has been reported in several applications of FEL control and widely applied. The approach taken for the weights update was the backpropagation algorithm [89], that is based on the minimization of the feedback command, which serves as the cost function for the algorithm. To implement it, the ADAM method was used, which is an SGD approach for an efficient stochastic optimization that only

requires first-order gradients with little memory requirement [83]. This stochastic method uses adaptive learning rates per weight connection, provides adaptability to the ANN training, and decreases the learning time.

The weights values are updated on each training iteration and the ANN stops training when the feedback controller has a contribution to the total control of 5% or below.

All the ANN software was implemented from scratch, with the help of books [80], that theorized the ANN behavior, the SGD and backpropagation algorithm on a MLP structure, and with the help of literature that developed several optimization algorithms that accelerated the SGD training method, like ADAM [83]. Figure 32 shows the used architecture for the feedforward controller.

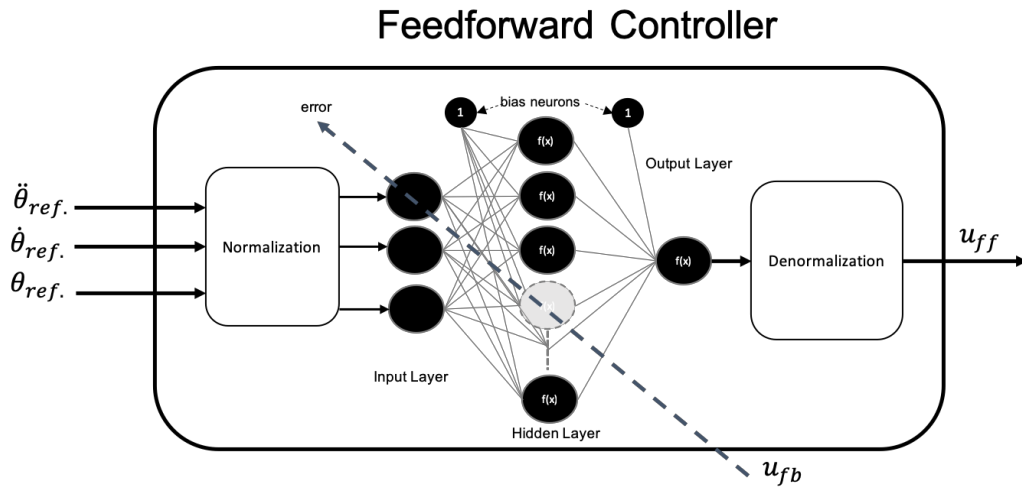


Figure 32 - ANN as feedforward controller.

The input values for the ANN were the reference angular position, which were derived to obtain a reference speed and reference acceleration. A normalization method was used to scale the input data between $[-1;1]$, given by in Equation (36).

$$y_{normalized} = 2 \frac{y_{input} - y_{min}}{y_{max} - y_{min}} - 1 \quad (36)$$

This method provides versatility to the training and testing, once it assures a trajectory magnitude limits and speed independent control [90]. With this, the inputs are always scaled between $[-1;1]$ and the ANN is indifferent to modifications applied to the input trajectory. This feature is an important advantage once it decreases training time and enables an adaptive feature to the controller. Although this method enables slight variations in the magnitude of the input trajectory, it was needed to have an ANN per orthosis given the angular differences

between knee and ankle joints along the gait cycle. With this normalization method, the predicted output needs to be denormalized to the correct magnitude of control values, which were previously established to a minimum and maximum of -2500 and 2500, respectively.

5.1.2.3 Real-time Algorithm

Figure 33 displays a flowchart with the developed algorithm of the real-time FEL controller in SmartOs.

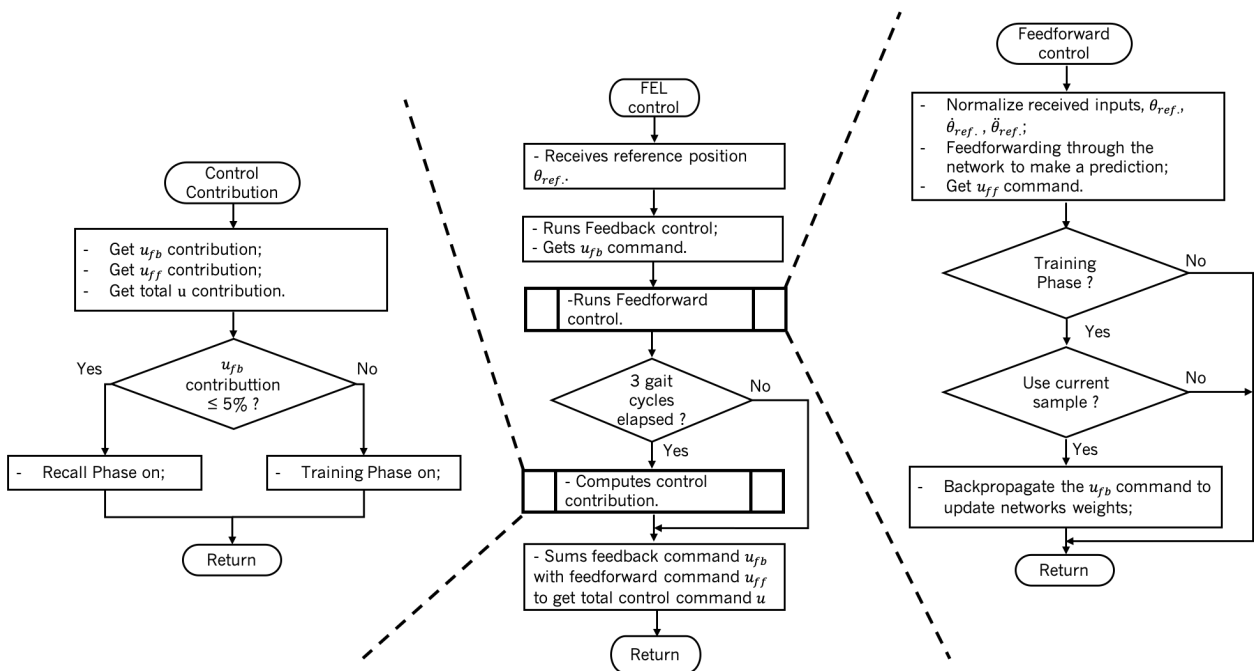


Figure 33 - Flowchart with the implemented algorithm of the FEL real-time control.

As is possible to see in Figure 33, at each control iteration, the PID controller performs a feedback command (u_{fb}) based on the error between the reference angular position ($\theta_{ref.}$) and the measured angular position given by a potentiometer embedded on the orthosis ($\theta_{meas.}$). The ANN receives the computed feedback command and forwards the normalized input references signals ($\theta_{ref.}, \dot{\theta}_{ref.}, \ddot{\theta}_{ref.}$) through its layers, getting a feedforward command prediction (u_{ff}).

Now, if the ANN is in the training phase, it is trying to learn the plant's IDM. It receives as error value the feedback command (u_{fb}), runs a stochastic function that outputs a value between [-1; 1] and if it is positive it computes its gradient and backpropagates it through the ANN, in order to update its parameters, otherwise it ignores the sample in order to randomize

the received inputs. On the other hand, if the controller is in the recall phase, it has already learned the plant's IDM and only performs the feedforward prediction.

Next, the two control signals (u_{fb} and u_{ff}) are summed, to generate a total command signal (u) that is applied to the orthosis. A saturator limits this command to a minimum value of -2500 and a maximum value of 2500, to protect the operability of the orthosis.

When three gait cycles are performed by the orthosis, the controller estimates the contribution executed by each controller to the total control command, computing the following equation:

$$u_{fb}(\%) = 100 \frac{\overline{u_{ff}^2}}{u} \quad (37)$$

where $u_{fb}(\%)$ is the feedback command contribution, $\overline{u_{ff}^2}$ is the mean squared feedforward contribution, and u is the total control command. If the feedback controller has only 5% (or lower) of contribution to the total control command, the ANN has learned the IDM of the plant and the controller is in recall phase. Otherwise, the ANN is acquiring the plant's IDM, being in training phase.

It is important to notice that the FEL controller could not be implemented in offline because we do not have the PKO and PAFO model. So, the taken approach to find the initial conditions for the ANN of the PKO and the PAFO was to tune, using MATLAB, an ANN to learn the dynamic of the PID feedback controller since we believe that the feedforward controller will have a similar range of operation, either in amplitude and frequency. Posteriorly, the found values for the initial conditions were used to setup the ANN to be trained in real-time training.

Thus, the taken approach to find the best number of hidden nodes, initial weights values and initial learning rate, in the lowest time possible and with an empirically established target validation error of 5% is presented in Figure 34. The used position, speed, and acceleration reference signals and PID commands to perform the offline tuning were previously acquired from 2 subjects, for the gait speeds of 0.8 km/h, 1.0 km/h, 1.2 km/h, as they are the target speeds to be used with end-users. Each training set has one gait cycle per speed for each subject, adding up to 6 gait cycles.

Hereby, Table 13 summarizes the tuned initial conditions for the ANN of the PAFO and for the ANN for the PKO.

Table 13 – Values to tune and achieved results of the offline implemented ANN. α and β are configurable scales of the weights initialization and initial learning rate, respectively, L are the Input nodes length and M the hidden nodes length.

	Assistive Device	Hidden Nodes	α and β Values	Weights Initialization	Learning Rate
Training	PKO and PAFO	3-20	{1, 10, 100}	$-\frac{1}{\alpha\sqrt{L}} \leq w_{hid} \leq \frac{1}{\alpha\sqrt{L}}$	$0.001 * \beta$
				$-\frac{1}{\alpha\sqrt{M}} \leq w_{out} \leq \frac{1}{\alpha\sqrt{M}}$	
Results	PKO	4	10 and 10	$-\frac{1}{10\sqrt{3}} \leq w_{hid} \leq \frac{1}{10\sqrt{3}}$ $-\frac{1}{10\sqrt{4}} \leq w_{out} \leq \frac{1}{10\sqrt{4}}$	0.0001
	PAFO	5	10 and 100	$-\frac{1}{10\sqrt{3}} \leq w_{hid} \leq \frac{1}{10\sqrt{3}}$ $-\frac{1}{10\sqrt{5}} \leq w_{out} \leq \frac{1}{10\sqrt{5}}$	0.00001

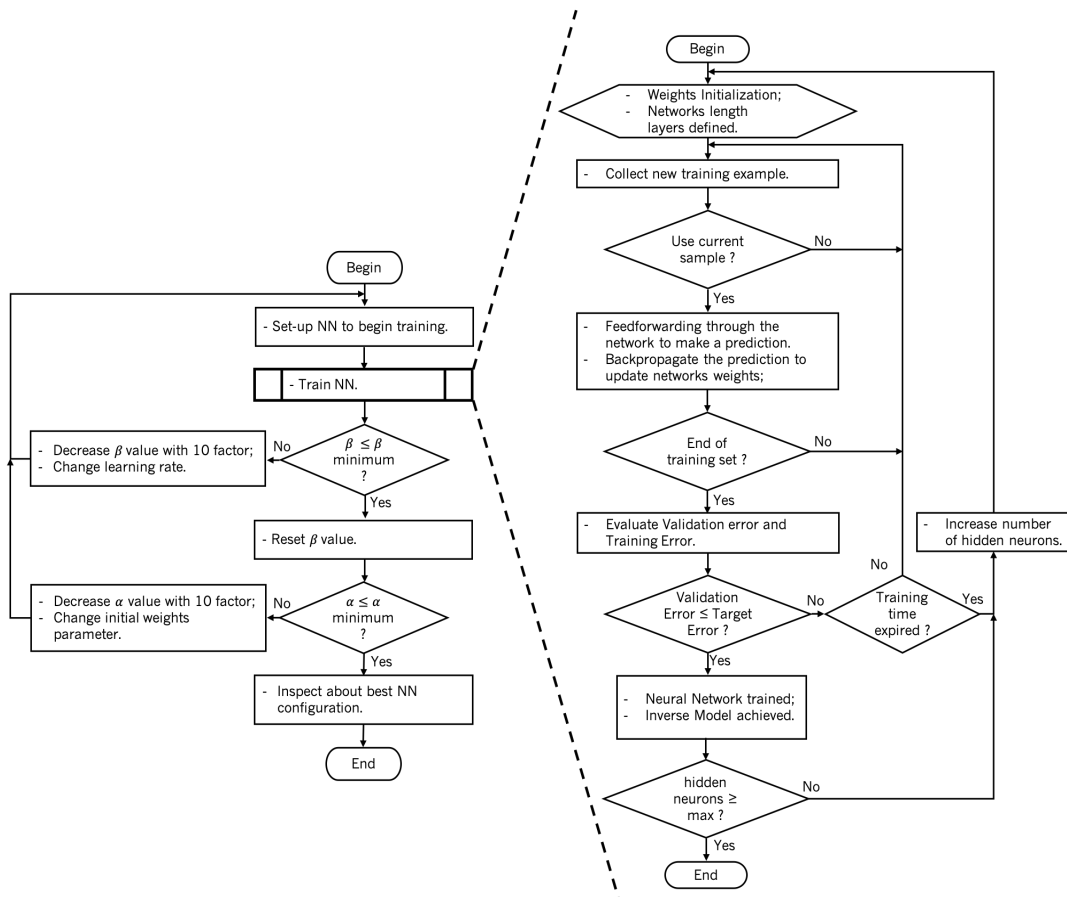


Figure 34 - Flowchart with neural network offline training algorithm. α and β are configurable scales of the weights initialization and initial learning rate, respectively.

5.2 Validation Procedure

To validate the real-time FEL controller three procedures were considered. The first one covers a validation of the PKO without load and then the validation of the PAFO without load, to investigate the behavior of the FEL controller for each device, separately. Figure 35 shows the used setup for this validation procedure. The measured and reference angular position was inspected, as the performed feedback command u_{fb} and feedforward command u_{ff} and the angular position error was inspected. The used gait speeds to move the PAFO and PKO for this validation were 0.8, 1.0, and 1.2 km/h.

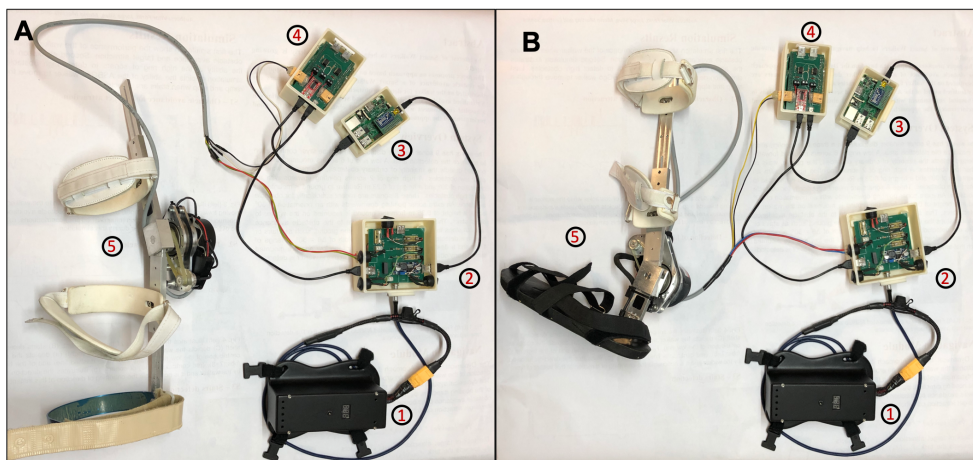


Figure 35 - Used set-up to validate FEL control without load. 1: Power Supply; 2: Power circuit; 3: CCU; 4: LLOS; 5: Actuation device (A: PKO; B: PAFO).

Some cautions with the execution of the FEL validation without load were taken. During the online training phase of the feedforward controller, the ANN is still trying to learn the IDM of the PKO or PAFO. Thus, the feedforward command predictions that are being summed with the feedback command and applied to the orthosis could lead the system to produce undesirable movements, damaging the device. Therefore, it was decided to first perform the validations using a modified input knee/ankle reference position, without significantly changing its shape.

For the PKO validation without load, it was decided to add an offset of 15° to the normal knee reference position, so the joint does not perform movements closer to its mechanical limits, which are 0° for the lower and 90° for the higher limit. Figure 37.A/B/C shows the used reference trajectory, speed and acceleration for the PKO and Figure 37.D/E/F shows the used

normalized input trajectory, speed and acceleration for the ANN of the PKO. A trained ANN with the modified knee reference was achieved for a gait speed of 1.0 km/h. Subsequently, the ANN was tested with the normal knee trajectory with 3° of offset for gait speeds of 0.8, 1.0 and 1.2 km/h. This procedure is possible due to the normalization method of the input references and the denormalization of the predicted feedforward command value of the ANN. With this, the inputs are always scaled between [-1;1] and the outputs are comprised between the established minimum and maximum command values (-2500 and 2500), and the ANN is indifferent to the trajectory offset value as Figure 37 shows. This feature is an important advantage to clinical environments once it decreases the error risk, training time, and enabling an adaptive feature to the controller.

The validation without load of the PAFO employed a different procedure since the range of motion of the PAFO is more limited, being -20° for the lower limit and 20° for the higher limit. So, to work far from these limits, the normal ankle reference trajectory was modified with an attenuation gain of 40% and 4° of offset. With this modified trajectory, an IDM for the PAFO was learned for a gait speed of 0.8 km/h. The learned IDM was re-tuned to the normal ankle trajectory for the gait speeds of 0.8, 1.0 and 1.2 km/h. Figure 37.G/H/I shows the used reference trajectory, speed and acceleration for the PAFO and Figure 37.J/K/L shows the used normalized input trajectory, speed and acceleration for the ANN of the PAFO. With Figure 37 it is possible to observe that the ANN is indifferent to modifications applied to the input trajectory, both in magnitude and attenuation.

The second validation procedure was performed to inspect about the user-orthosis interaction and includes two healthy users (a male and a female) with demographic characteristics of 25.5 ± 0.7071 years old, height of 1.69 ± 0.0919 m, and weight of 64.50 ± 14.84 kg. They were asked to walk at {0.8; 1.0; 1.2} (km/h) in a level-ground on a treadmill, first with the PKO and then with the PAFO configured with the trajectory tracking assistive strategy, each performing 3 trials with duration of 1 minute each. Figure 37 shows the users performing the described validation.

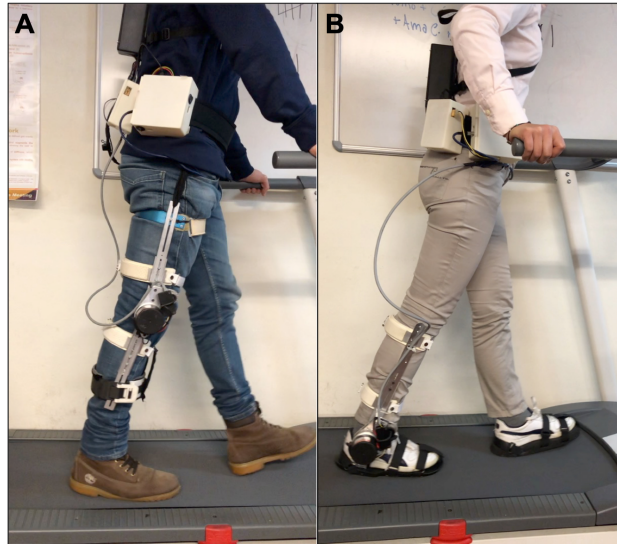


Figure 36 - Used set-up for the validation with a user on a treadmill. A: PKO; B: PAFO.

It was used the trained ANN with the plant's IDM learned in the validation without load. For the PKO the used reference trajectory was a normal knee trajectory with 3° of offset, and for the PAFO was the normal ankle trajectory, as Figure 37.A/G shows.

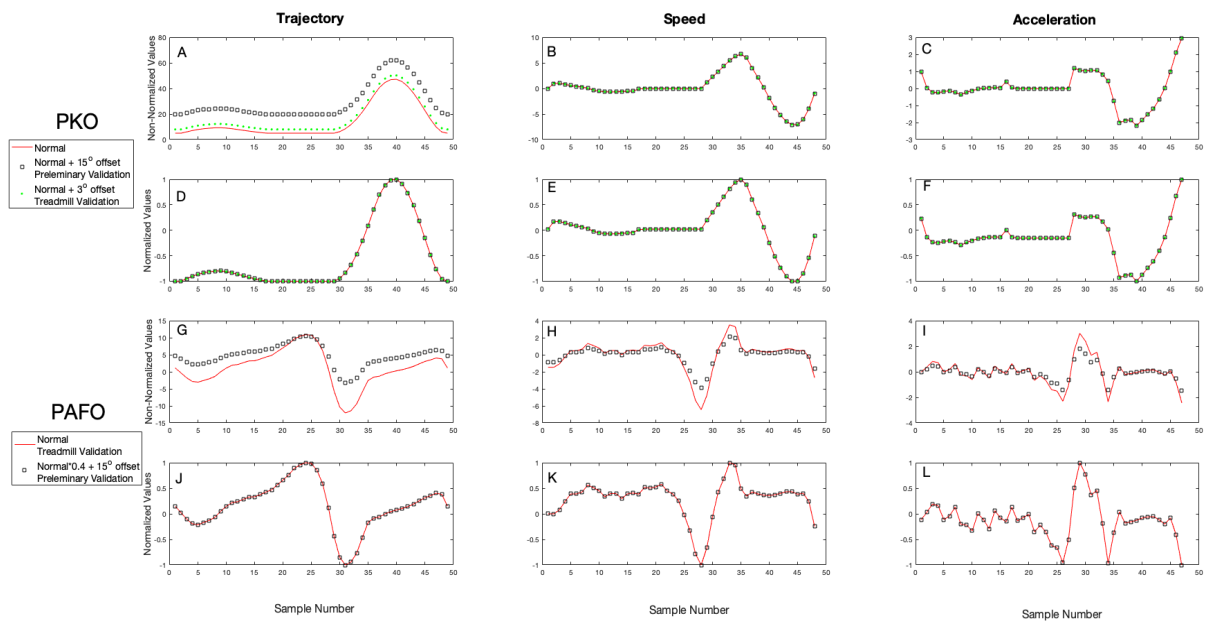


Figure 37 - Used reference signals for the validation procedures without load and with the user. A, B, C: PKO reference non-normalized trajectory, speed and acceleration, respectively; D, E, F: normalized input trajectory, speed and acceleration, respectively, for the PKO's ANN; G, H, I: PAFO reference non-normalized trajectory, speed and acceleration, respectively; J, K, L: normalized input trajectory, speed and acceleration, respectively, for the PAFO's ANN.

For the third validation procedure, it was analyzed the effect of external perturbations to the user-orthosis interaction. Under the same conditions as the ones taken for the validation of the user-orthosis interaction, the users were asked to counteract the PKO/PAFO, in order to evaluate how the controller would react to such disturbance, which may occur during a gait therapy.

5.3 Results & Discussion

5.3.1 Real-Time Implementation

In order to validate the FEL controller, the following signals were analyzed: (1) PKO/PAFO trajectory measured by a potentiometer and reference trajectory; (2) PID and ANN command and (3) Evaluation of PKO/PAFO position error. All signals were sampled with 100 Hz of frequency.

5.3.1.1 Powered Knee Orthosis

In SmartOs, the existing low-level control was the PID feedback controller, so the following FEL controller results are compared with this strategy.

Figure 38 presents the obtained results for the validation without load of the real-time FEL control with the PKO as plant, a selected speed of 1 km/h during three distinct gait cycle periods: Initial Phase; Middle Phase and Final Phase.

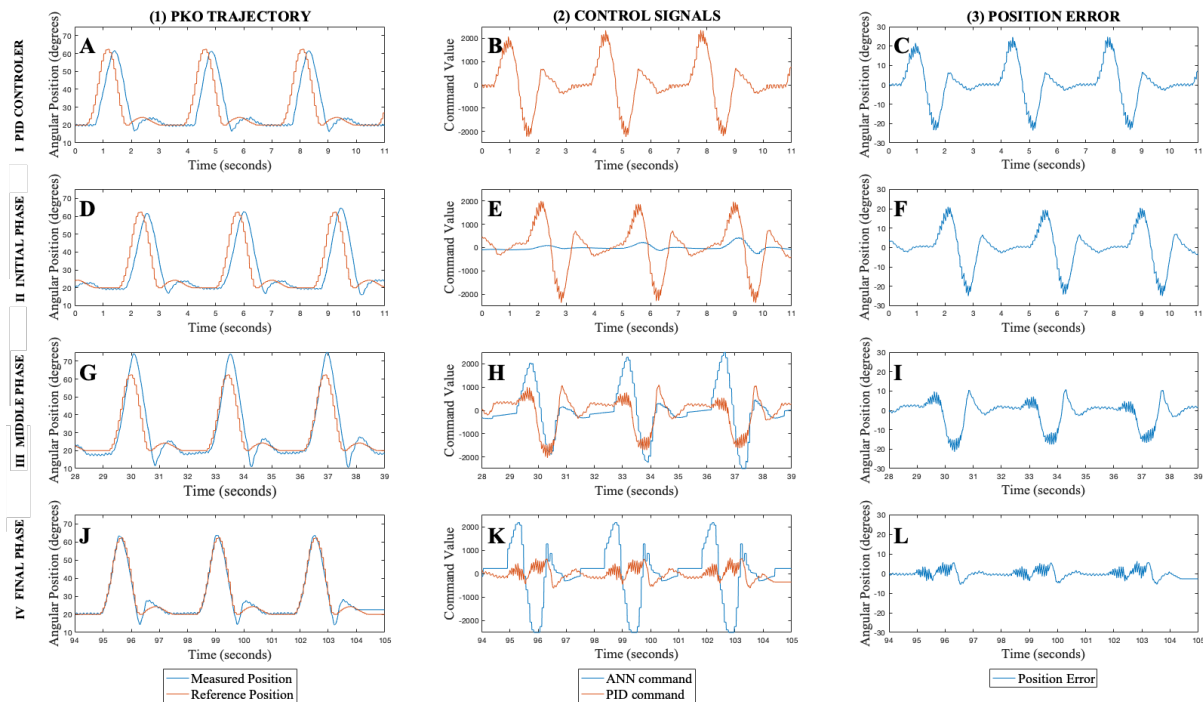


Figure 38 - Results of PID and FEL controller in validation without load. A, D, G, J: PKO Measured trajectory and Reference trajectory signals in the solo PID and Initial, Middle, and Final Phase of FEL, respectively; B, E, H, K: ANN and PID command signals in the solo PID and Initial, Middle, and Final Phase of FEL, respectively; C, F, I, L: PKO Position Error in the solo PID and Initial, Middle, and Final Phase of FEL.

The first row of Figure 38 presents the PID controller while the remaining signals illustrate the results of the FEL controller. Both controllers used as reference position trajectory a normal knee trajectory with an offset of 15° .

Concerning the PID controller, a delay phase of 210 ms is observed (Figure 38.A) between the reference and measured trajectory, which causes a position error that varies from -20° to 20° (Figure 38C). Figure 38.B displays the produced feedback command by the PID controller.

Figure 38.D/E/F shows the FEL controller in its Initial Phase (first 11 seconds). During this phase, the feedforward controller is starting to tune its ANN and its contribution to the total control signal is not significant, being the feedback PID controller the one that tracks the reference trajectory. It is possible to see in Figure 38.D that the measured trajectory has approximately 210 ms of phase delay from the generated reference signal. This outcome is very similar to the one seen in Figure 38.A since commands come mainly from the PID controller because the feedforward controller is in its initial learning phase, having a much lower influence in the total control signal (Figure 38.D). Hence, the error signal in Figure 38.E has, again, a variance between -20° and 20° .

In the Middle Phase (after 40 s), the FEL controller initiates to learn the IDM of the PKO and the measured position starts to decrease the phase difference with the generated reference signal, achieving 6 ms of phase delay, as can be seen in Figure 38.G. However, in order to correct its phase delay, the controller produces signals that causes the PKO to perform a trajectory which is 10° higher than what it should be, as shown in Figure 38.G. This is due to the fact that the ANN has not learned the IDM with the best performance yet. In Figure 38.H, it is possible to realize the increase in the ANN command signal and that the PID command signal has decreased comparatively to the initial phase. The error signal given in Figure 38.I has improved varying from 10 to -20 degrees of angular position.

For the Final Phase (past 90 s), it is verified that the FEL controller has successfully aligned the PKO trajectory with the reference trajectory and corrected the amplitude divergence, by observing Figure 38.J. In this phase, the IDM of the plant has been adequately obtained by the ANN and the FEL controller entered in the recall phase. The feedback controller only contributes with 4.4% to the total control command – Figure 38.K. Concerning the error value, it decreased about 75%, comparing Figure 38.F and Figure 38.L.

From these findings, it was verified that the ANN is capable of achieving the IDM of the PKO, taking on the control and discharging the feedback controller from tracking the position references. A time-effective response with lower position error and phase delay was accomplished, comparatively to a solo PID feedback controller implementation. It is important to mention that with the offset used in the reference position signal, the device was not damaged, if the normal knee reference trajectory had been used, during the middle phase, the PKO could have exceeded its normal range of motion, because the increase of 10° of amplitude would put the device out of its lower operating range.

Hereafter, with the current setup found for the IDM of the PKO, the FEL controller was tested for speeds of 0.8 and 1.2 km/h and as reference position a normal knee trajectory with an offset of 3° . Figure 39 displays the obtained results.

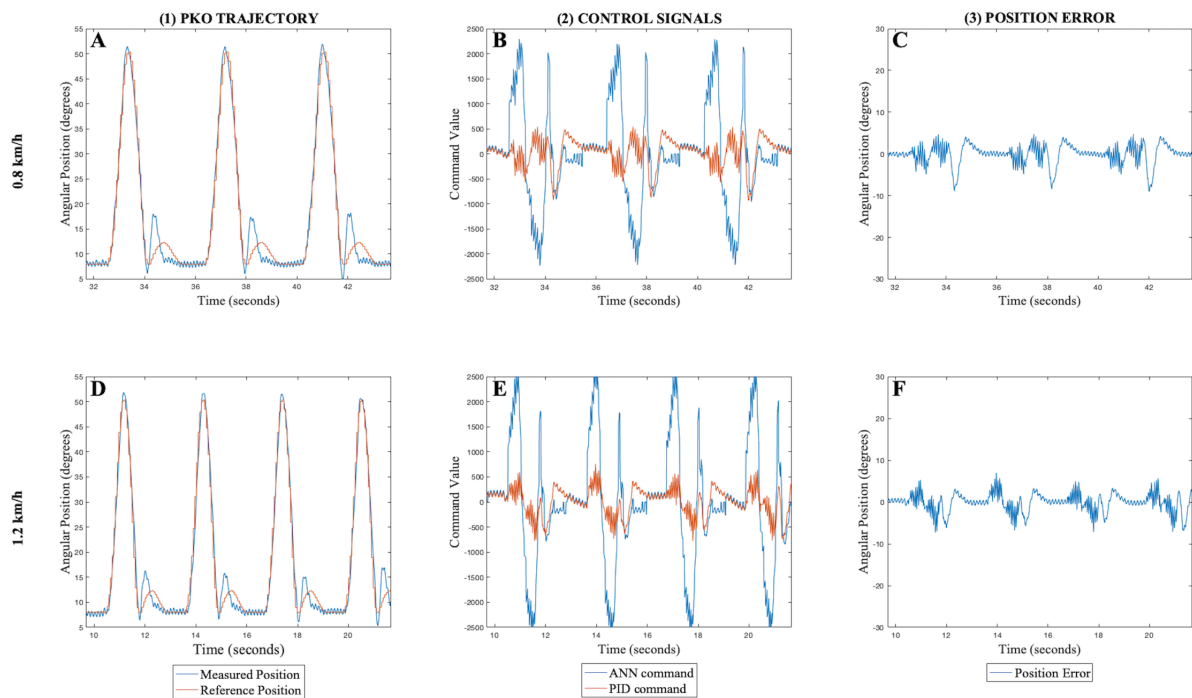


Figure 39 - Results of the FEL controller with a learned PKO IDM for the validation with load. A, D: PKO Measured trajectory and Reference trajectory signals for 0.8 km/h and 1.2 km/h, respectively; B, E: ANN and PID command signals for 0.8 km/h and 1.2 km/h, respectively; C, F: PKO Position Error for 0.8 km/h and 1.2 km/h, respectively.

With the learned IDM of the PKO for a gait speed of 1.0 km/h and a knee reference position with 15° of offset, it is possible so see in Figure 39 that the IDM is still valid for speeds of 0.8 km/h and 1.2 km/h and a reference position with only 3° of offset concerning the normal knee trajectory. Both tests achieved good trajectory responses and low position errors. Thus, this strategy proves to be independent of changes in ANN speed and in the offset value of the reference position.

Next, the results of the validation of the orthosis-user interaction was performed. One subject walked on a treadmill with 0.8, 1.0 and 1.2 km/h of gait speed and a normal knee trajectory with an offset of 3° was used, as shown in Figure 40.

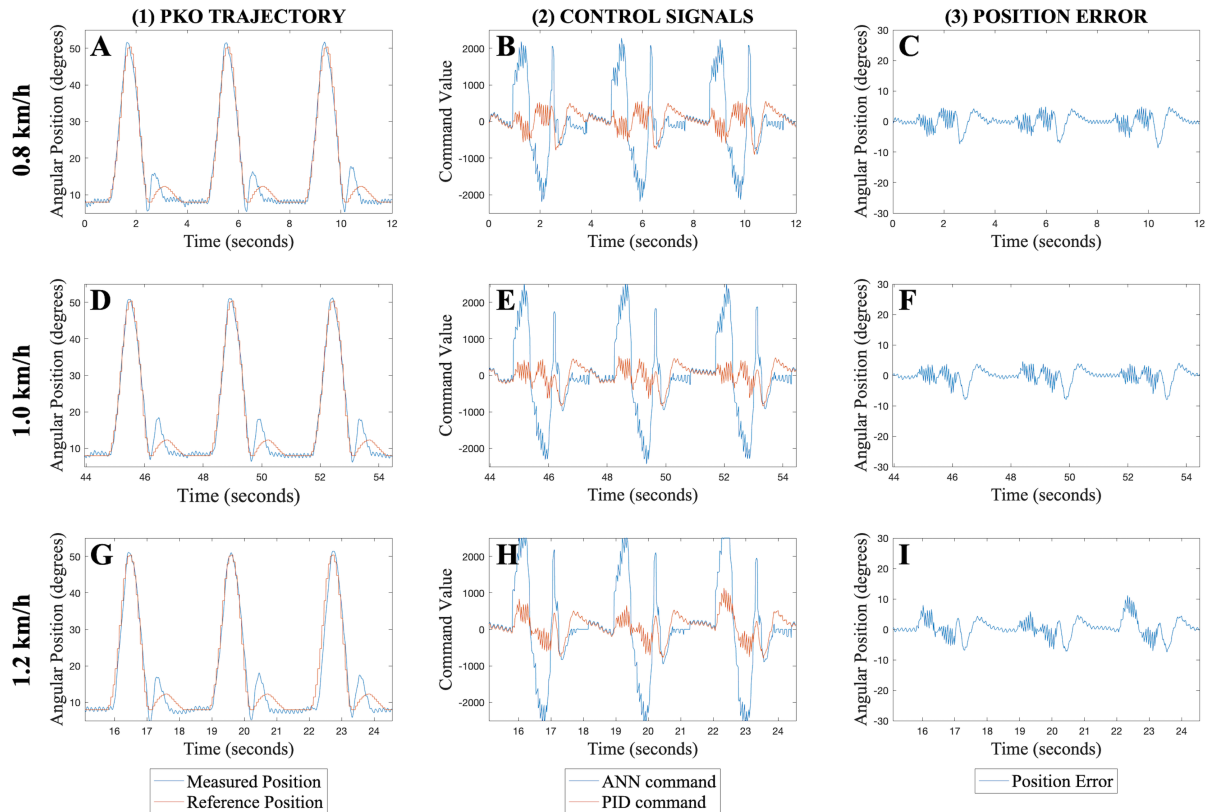


Figure 40 - Results of the FEL controller with a learned PKO IDM for the user-orthosis interaction validation. A, D, G: PKO Measured trajectory and Reference trajectory signals for 0.8 km/h, 1.0 km/h and 1.2 km/h, respectively; B, E, H: ANN and PID command signals for 0.8 km/h, 1.0 km/h and 1.2 km/h respectively; C, F, I: PKO Position Error for 0.8 km/h, 1.0 km/h and 1.2 km/h, respectively.

The obtained results are similar with the results obtained in the validation without load, either in the performed trajectory and the position error. Due to the ANN trained in the validation without load and only by changing the normalization limits, it was possible to use the trained ANN with a learned PKO IDM to the different validation speeds.

With the two validations performed it is now possible to compare the obtained results with a solo PID control.

Both in the validation without load and the validation of the user-orthosis interaction and for all tested speeds, the solo PID achieved a Normalized Root Mean Squared Error (NRMSE) around 22% and a phase delay of 230 ms.

Concerning the FEL controller in the validation without load, it accomplished a mean NRMSE of 5.13% and a mean phase delay of 13 ms. The feedback controller only contributed to the total control with 5.9% (mean), for all tested speeds. Regarding the validation of the user-orthosis interaction, a mean NRMSE of 5.87%, a mean phase delay of 12.5 ms and a mean

contribution of 6.52% of the feedback controller to the total control was accomplished. The results for both validations were very similar. Comparing the validation without load and the validation of the user-orthosis interaction: (1) the mean NRMSE increased by 0.74%; (2) the phase delay decreased by 0.5 ms; (3) the mean feedback contribution increased 1% to the total control command. This could be due to the load added in the system (the user), that may difficult the controller to follow the reference trajectory. The knee trajectory has a big and fast range of motion during the swing phase and the required acceleration is bigger. Nevertheless, it is worth mentioning that the feedback command contribution to the total control signal never reached values equal or below 5%, except during the speed of 1.0 km//h.

When comparing both control approaches, the FEL controller achieved better results in all metrics. Analyzing both validations, the FEL controller decreased the NRMSE 16.5% and 93% for the phase delay (approximately).

Table 14, summarizes the obtained results achieved with the PKO as assistive device, using PID and FEL as controllers for the speeds of 0.8, 1.0 and 1,2 km/h. The NRMSE, Phase Delay and the contributions to the total control signal of the feedback command (u_{fb}) and feedforward command (u_{ff}) are displayed for the preliminary validation and treadmill validation (mean of all trials of the two subjects).

Table 14 - Achieved results for both validations with the PKO.

Validation	Speed km/h	Low-Level Control	NRMSE %	Phase Shift ms	u_{ff} Contribution %	u_{fb} Contribution %	
Without load	1.0	PID	22.33	210	-	-	
		FEL	Initial	22.33	210	0	10
			Middle	9.89	60	60.7	39.3
			End	5.54	0	96.5	4.5
	0.8	PID	21.58	260	-	-	
		FEL	5.17	20	94	6.9	
	1.2	PID	22	230	-	-	
		FEL	4.68	20	93.7	6.3	

Table 15 - continued.

Validation	Speed km/h	Low-Level Control	NRMSE %	Phase Shift ms	u_{ff} Contribution %	u_{fb} Contribution %
User- Orthosis Interaction	1.0	PID	22.33	210	-	-
		FEL	5.69	0	93.89	6.11
	0.8	PID	21.58	260	-	-
		FEL	5.55	15	93.06	6.94
	1.2	PID	22	230	-	-
		FEL	6.37	22.5	93.48	6.52

Finally, Figure 41 shows the validation of the FEL controller to external perturbations during the user-orthosis interaction, with a disturbance applied during walking in the terminal stance phase (before toe-off event). The selected speed was 1 km/h and as reference position a normal knee trajectory with an offset of 3°.

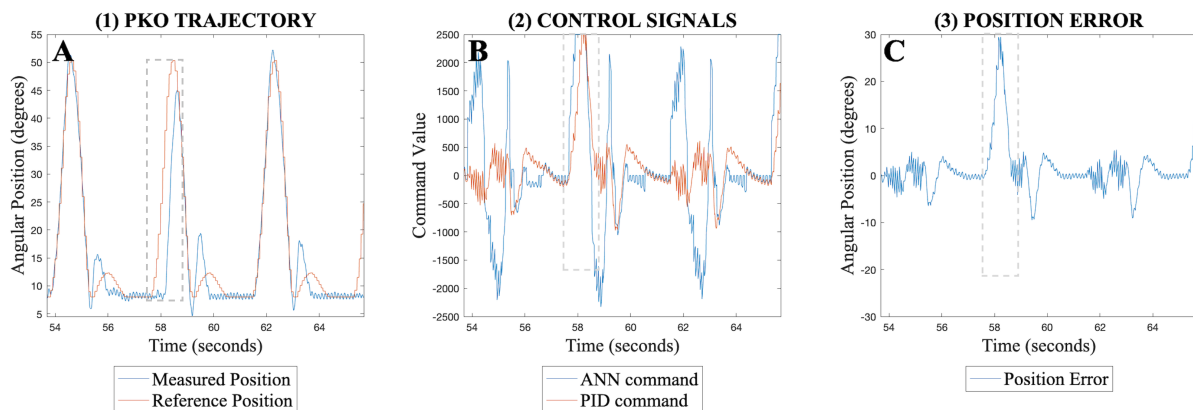


Figure 41 - Results of FEL controller to external disturbances during the user-orthosis interaction. A: PKO Measured and Reference position signals; B: ANN and PID command signals; C: PKO Position Error.

With the learned IDM, it was possible to analyze the behavior of the FEL controller to external disturbances that could affect the normal user-orthosis interaction. Hereby, in Figure 41.A, it is possible to inspect the moments when the user counteracted the orthosis

movement, preventing it from increasing its angular position, in one occasion (at 58 s signaled with a dashed grey rectangle). Looking to Figure 41.B, in that same moment, the PID command signal increases, since the error signal grows (see Figure 41.D) due to the displacement between the reference and measured position signals. The feedforward command signal stayed periodic, as the reference signals and the learned IDM did not change. Therefore, as expected, the feedback controller acts in an attempt to cancel the increased error signal to correct the PKO angular position, preventing any damage of the user and of the device.

5.3.1.2 Powered Ankle-Foot Orthosis

The following results show only the signals with the ANN tuned to the normal ankle trajectory, once several scale factors were used until the normal trajectory was reached.

Figure 42 shows the obtained results for the validation without load of the real-time FEL controller with the selected gait speed of 0.8 km/h, focusing only on the final phase of the FEL controller and comparing it to the PID feedback controller, the existing low-level controller in SmartOs.

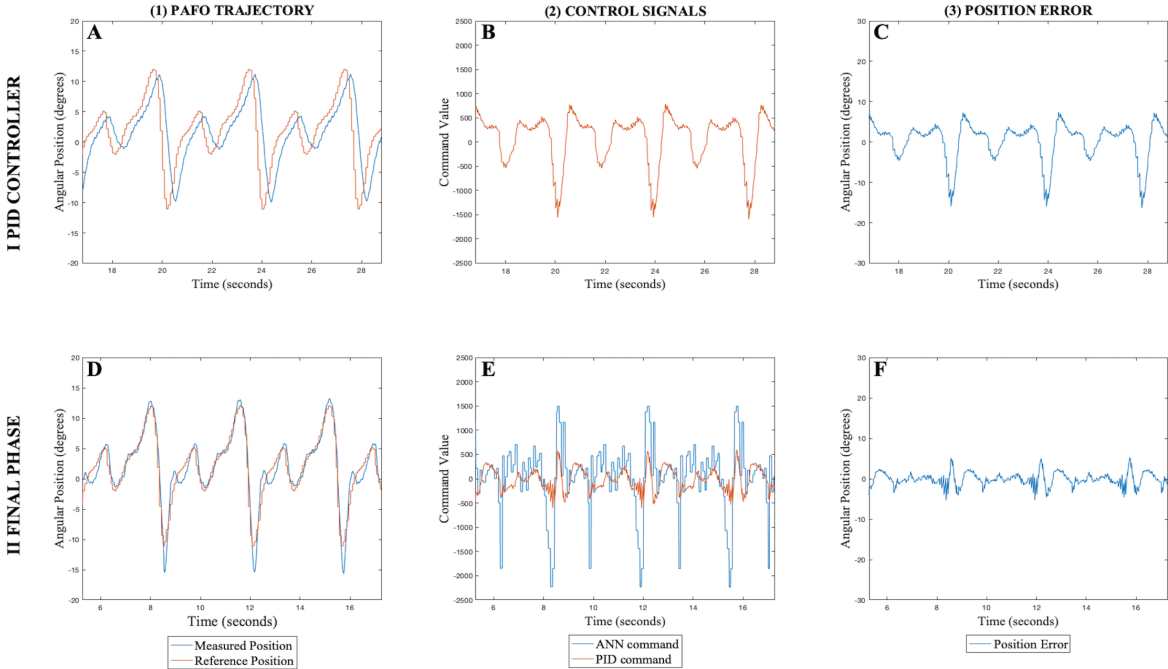


Figure 42 - Results of PID and FEL controllers in validation without load. A, D: PAFO Measured trajectory and Reference trajectory signals in the solo PID and Final Phase of FEL, respectively; B, E: ANN and PID command signals in in the solo PID and Final Phase of FEL, respectively; C, F: PAFO Position Error in the solo PID and Final Phase of FEL, respectively.

Concerning the PID controller, a phase delay of 250 ms is observed in Figure 42.A, which causes the position error to vary from -16° to 7° (Figure 42.C). Figure 42.B shows the corresponding feedback command.

For the Final Phase of the FEL controller, it is verified a phase delay of 20 ms (Figure 42.D) decreasing the position error, which varies now from -5° to 5° (Figure 42.F). Figure 46.E shows the predominance of the feedforward controller over the feedback controller. Nevertheless, it is observed that the measured position sometimes exceeds the reference position, due to an existing error in the learned IDM.

These findings suggest that the tuned ANN is capable of learning the IDM of the PAFO, taking on the control and dismissing the feedback controller from tracking the position references. Again, a time-effective response with lower position error and phase delay was accomplished, in contrasting to solo PID feedback control implementation.

Hereafter, with the current set-up found for the IDM of the PAFO, the FEL controller was tested for the gait speeds of 1.0 and 1.2 km/h. Figure 43 displays the obtained results.

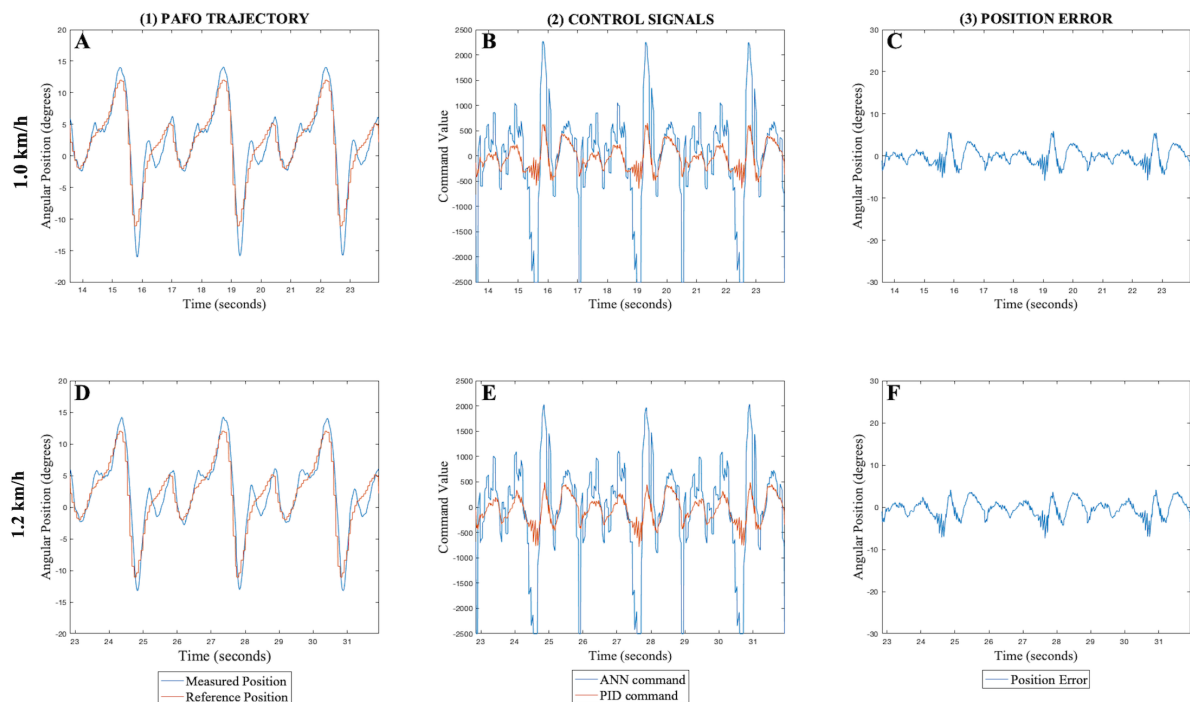


Figure 43 - Results of the FEL controller with the learned IDM of the PAFO in the validation without load. A, D: PAFO Measured trajectory and Reference trajectory signals for 1.0 km/h and 1.2 km/h, respectively; B, E: ANN and PID command signals for 1.0 km/h and 1.2 km/h, respectively; C, F: PKO Position Error for 1.0 km/h and 1.2 km/h, respectively.

Contrary to what happened with the PKO validation without load, the learned IDM of the PAFO had to be tuned for the gait speeds of 1.0 km/h and 1.2 km/h. Figure 43 shows low position error for both speeds, varying in form -5° to 5° for 1.0 km/h and -7° to 4° for 1.2 km/h.

Next, the results for the validation of the user-orthosis interaction with one subject walking on a treadmill for 0.8, 1.0 and 1.2 km/h, are presented in Figure 44.

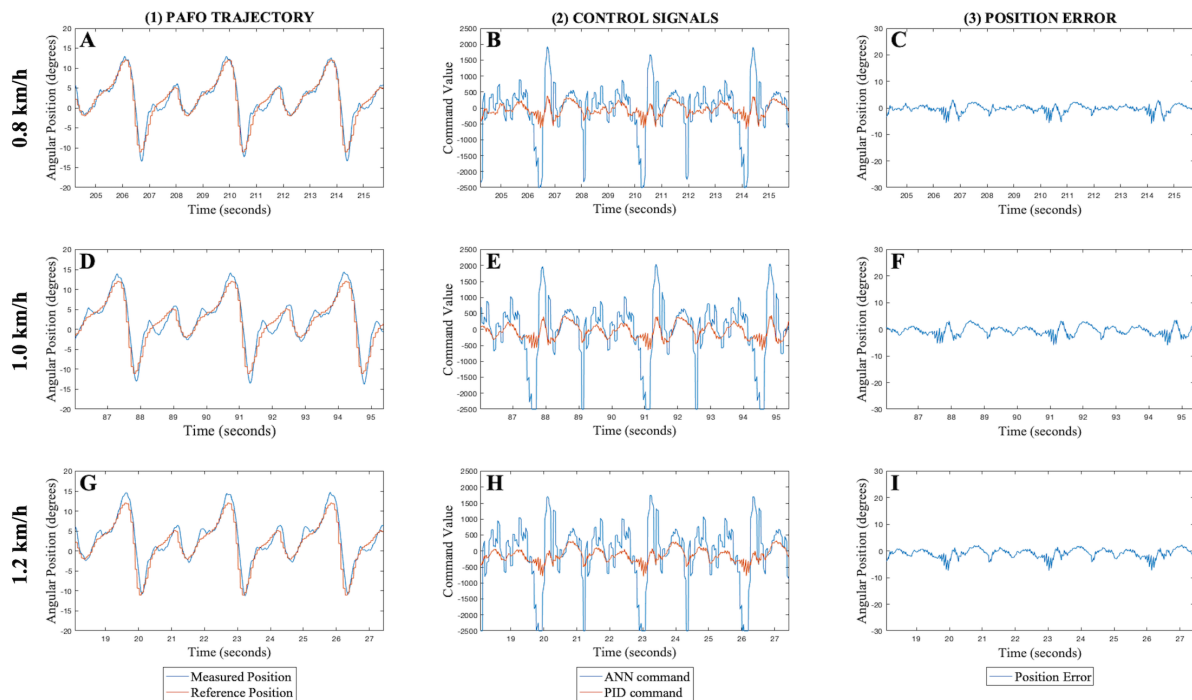


Figure 44 - Results of the FEL controller with a learned IDM of the PAFO for the user-orthosis validation. A, D, G: PKO Measured trajectory and Reference trajectory signals for 0.8 km/h, 1.0 km/h and 1.2 km/h, respectively; B, E, H: ANN and PID command signals for 0.8 km/h, 1.0 km/h and 1.2 km/h respectively; C, F, I: PKO Position Error for 0.8 km/h, 1.0 km/h and 1.2 km/h, respectively.

The results show low position errors and feedback contribution. In this validation, it is also possible to see that the measured position does not exceed the reference position overly, contrarily to the validation without load.

It is now possible to compare the obtained results, both for FEL controller and the solo PID controller.

In the both validations, the solo PID controller achieved a mean NRMSE of 24% and a phase delay of 250 ms for all speeds.

Concerning the FEL controller in the validation without load, it yielded a mean NRMSE of 6.6% and a mean phase delay of 23.3 ms for all tested speeds. The feedback controller only contributing to the total control with 8.3% (mean), for the three analyzed speeds. When the validation of the user-orthosis interaction was performed, a mean NRMSE of 6.51%, a mean phase delay of 25 ms and a contribution of 5.8% (mean) of the feedback controller to the total control was accomplished. Again, the results for both validations were very similar. It was observed that: (1) mean NRMSE decreased 0.09% ; (2) the phase delay increased 1.7 ms (3) and the feedback contribution decreased 2.5%. The decrease in the NRMSE might be due to the load added to the PAFO (the user) that stopped it to exceed the reference signal.

When comparing both controllers, the FEL controller achieved better results in all metrics. Analyzing both validations, the FEL controller decreased the NRMSE 17.5% (approximately) and 90.7% for the phase delay.

Table 16, summarizes the obtained results achieved with the PAFO as an assistive device, the PID and FEL as controllers for the speeds of 0.8, 1.0 and 1,2 km/h. The NRMSE, Phase Delay and the contributions to the total control signal of the feedback command (u_{fb}) and feedforward command (u_{ff}) are displayed for the validation without load and validation of user-orthosis interaction (mean of all trials of the two subjects).

Table 16 - Achieved results for both validations with the PAFO.

Validation	Speed km/h	Low-Level Control	NRMSE %	Phase Delay ms	u_{ff} Contribution %	u_{fb} Contribution %
Without load	0.8	PID	22.28	250	-	-
		FEL	5.63	20	91.5	8.5
	1.0	PID	24.53	260	-	-
		FEL	6.54	20	92.1	7.9
	1.2	PID	26.99	250	-	-
		FEL	7.68	30	91.4	8.6

Table 16 – continued.

User- Orthosis Interaction	0.8	FEL	6.95	30	93.8	6.2
	1.0	PID	24.28	250	-	-
	1.0	FEL	5.99	20	93.1	6.9
	1.2	PID	26.99	250	-	-
	1.2	FEL	6.58	25	95.16	4.84

For the validation of external disturbances to the user-orthosis interaction with the FEL controller using the PAFO, Figure 45 shows the controller behavior. The external disturbances were applied in the initial stance phase (before foot flat event). The selected speed was 1 km/h.

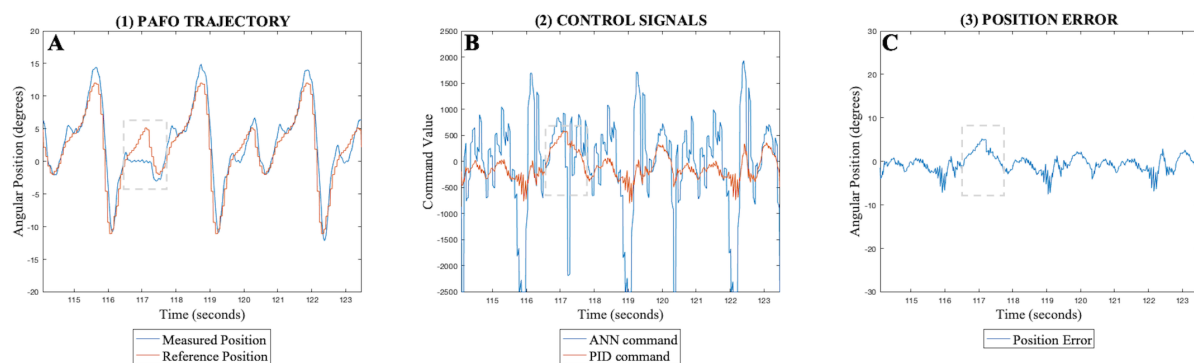


Figure 45 - Results of the FEL controller to external disturbances during user-orthosis interaction. A: PAFO Measured and Reference position signals; B: ANN and PID command signals; C: PAFO Position Error.

In Figure 45.A, it is possible to see the moment in which the user counteracted the orthosis movement, preventing it from increasing its angular position, at 117 s (signalized with a dashed grey square). Figure 45.B shows an increase of the PID command signal, once the error signal grows (see Figure 45.C) due to the displacement between the reference and measured position signals, in that same moment. Respecting the command signal performed by the feedforward controller, it stayed periodic, since the reference signals did not change.

Therefore, the feedback controller increases its contribution in attempt to cancel the increased error signal and correct the PAFO angular position, preventing any damage of the user and of the device.

At the end of both validation procedures with the PKO and PAFO, the two users were asked if they felt any difference, in terms of assistance, when changing from the PID controller to the FEL controller. Both said that the generated assistance during FEL controller supplies more power, forcing the limb to follow the established path. Specifically, with the PAFO this was more obvious during the toe-off and heel-strike events and with the PKO it was during the swing phase. Users felt comfortable when wearing the device with the FEL controller, since it assists the moments that involve greater energy consumption.

5.4 Conclusions

This chapter presents the design and implementation in SmartOs of the FEL controller.

In order to set-up the parameters of the ANN, an offline ANN tuning was performed, both with data concerning the PKO and the PAFO. With this, it was possible to proceed to the real-time implementation and validation, with important safety measures to the system and the user.

A validation without load demonstrated good results, decreasing the position error and nearly eliminating the phase delay between the reference and measured positions of both assistive devices. With the validation of the user-orthosis interaction, the FEL controller effectively performed the position tracking with position errors between 5% to 7%, and with a maximum phase delay of 3%. When comparing to the PID feedback controller, a decrease around 15% to the position error and 200 ms to the phase delay was achieved with the FEL control.

Hereby, the FEL controller demonstrated efficiency to perform the position tracking control, achieving good time effectiveness as precision in positioning. The ANN demonstrated to be speed and scale/offset in reference independent since a pre-trained ANN to a different speed and scaled/offset reference position achieved good results. Robustness was also

accomplished, by successfully correct both the PKO and PAFO positioning, when external disturbances were faced to the user-orthosis interaction.

6 CONCLUSIONS

Neurological diseases or motor injuries can origin certain gait pathologies, which often produce abnormal gait patterns, affecting deeply human mobility. Powered assistive devices, like powered lower-limb exoskeletons and orthoses are starting to complement conventional therapeutic regarding gait rehabilitation, to actively aid or restore the gait patterned affected by gait pathologies. With such devices, certain difficulties associated with conventional therapeutic can be overcome, by relieving the repetitive and heavy work of physical therapies, while improving the orthopedic and neurologic recovery effectiveness.

The human motor control system starts to influence the design of bioinspired architectures for these lower-limb robotic devices, comprising the definition of distinct levels of controllers (high-, mid-, and low-level) distributed hierarchically. At the low-level, the controller has to track the reference state with low error. Several low-level controllers exist, however to achieve good adaptive results deep modeling and mathematical knowledge about the controlled plant is needed. So, it is desired a controller that has adaptive features without arduous implementation process.

In this master dissertation, it was developed and validated a real-time Feedback-Error Learning low-level control. It was embedded in SmartOs, an existing modular, bioinspired, real-time, time-effective, and ergonomic gait system and tested in two lower-limb orthotic actuation devices, the PKO and PAFO.

First, a literature analysis of the FEL controller was performed, to inspect its features, advantages and disadvantages. It was concluded that the FEL controller has an adaptive behavior without demanding high mathematical details about the target plant. This is a great advantage over other lower-level controls used in orthotic devices, such as: PID (and its variants), LQR and H^∞ . The FEL controller includes a feedforward controller that uses an ANN to online learn the IDM of the plant, together with a feedback controller that grants stability during the ANN training process and, handles disturbances afterwards.

SmartOs is as a modular wearable robotic gait system, capable of providing personalized gait rehabilitative tasks. It is composed by several modules, including wearable sensor

systems, actuation systems, gait analysis tools, and control strategies. A validation of SmartOs temporal requirements was performed to ensure an effective actuation and gait monitoring. The results demonstrated that SmartOs can time-effectively perform the necessary gait control and gait monitoring. Since SmartOs only has a PID low-level control, the FEL control was implemented in this smart system.

This work aimed for real-time implementation in the SmartOs. Thus, a study of the ANN features to be used in a function approximator application was addressed. The MLP was defined as the appropriated ANN structure to acquire the assistive device IDM. Therefore, an SGD training method with a backpropagation algorithm that minimizes the feedback error command and ADAM optimizer that uses adaptive learning rates per weight connection was implemented, ensuring adaptability to the ANN training and decreasing the learning time.

An offline tuned of the developed ANN was performed, so the initial set-up of the ANN could be found for the feedforward controller of the PAFO and PKO, such as the initial weights, number of hidden nodes and initial learning rate. With the initial set-up found for the ANN it was possible to use the FEL controller in a real-time scenario, with the development of validation situations.

A validation without load of the FEL controller verified good performance, in terms of learning the plants' IDM, position error and phase delay. With the confirmation of the feedforward controller ability in learning the plant's IDM, a validation of the user-orthosis interaction with two subjects walking on a treadmill was performed. When the PKO and PAFO were worn by the users with the trajectory tracking assistive strategy, the proposed FEL controller was capable of following the reference trajectory for the speeds of 0.8, 1.0 and 1.2 km/h. It demonstrated position errors between 5% to 7%, and with a maximum phase delay of 3%. When compared to the PID feedback controller, a decrease around 15% in the position error and 200 ms in the phase delay was achieved with the FEL controller. Good responses to external disturbances were also verified, where the FEL controller shown capabilities to correct the deviations that the devices suffered. Hereby, the FEL controller implemented in SmartOs suggested to be a time-effective with low tracking error and phase delay, low training time when compared to the PID feedback control.

Furthermore, with the developed work, all established goals were successfully achieved and the RQs defined in Chapter 1 can be answered:

- RQ1: *Which are the FEL controller advantages over other low-level controllers applied to powered assistive devices?*

This RQ was answered in Chapter 2. The FEL controller is composed by a feedforward controller that has the plant's IDM to predict the right control commands to properly track the reference input and feedback controller to only handle external disturbances that can affect the plant and provide to the feedforward controller an error signal as miss-match to the learned IDM.

- RQ2: *Is SmartOs a time-effective assistive and monitoring rehabilitation system?*

This RQ was approached in Chapter 4. An empiric validation protocol was developed and applied in order to validate SmartOs as a time-effective architecture. It was verified that both the monitoring (WMSS) and assistive modules (LLOS), and respective sub-modules when operating alone and together, fulfill the demanded time constraints. Consequently, all technical requirements are fulfilled such that the SmartOs is able to provide time-effective gait assistance and monitoring.

- RQ3: *Are ANN capable of effectively acquire an IDM of a powered assistive lower-limb device?*

This RQ is answered in chapter 5. With the real-time implementation, it was possible to conclude that tuned ANN is capable of learning the IDM of a powered assistive lower-limb device. It was accomplished with small training time and low performance error. Moreover, the ANN demonstrated versatility to modifications in the reference trajectory and gait, due to normalizations methods. Also, training with a pre-trained ANN with different conditions enabled to successfully obtain the IDM of the target assistive device, giving an important advantage to clinical environments once it decreases the error risk, training time and enabling an adaptive feature to the controller.

- RQ4: *Can FEL control provide good real-time control performance to a powered assistive lower-limb device?*

Chapter 5 also answers this RQ. With the performed validations, it was concluded that the FEL controller achieves good real-time results with a low position error and phase delay, both for the PKO and PAFO with a control frequency of 1kHz. These findings open opportunities for the real-time application of FEL controller in user-oriented, time-effective gait rehabilitation, approached by powered assistive lower-limb devices, in scenarios with end-users.

6.1 Future Work

Future work within this dissertation could be: (1) validation of the FEL control with more subjects to fully inspect about its control performance; (2) use a more thorough validation protocol to inspect about the behavior of the FEL controller to external disturbances to the user-orthosis interaction; (3) include a validation with pathologic patients using the PKO and PAFO assistive devices with FEL control; (4) analyze the controller with different disturbance scenarios; (5) develop and apply inquiries to have a qualitative and subjective analyzes of the FEL control in comparison to different low-level controllers; and (6) use different trajectory tracking assistive strategies like torque control or ANN assistive strategies such as myoelectric control or impedance to investigate how the FEL control would behave with these different high-level controllers and inspect about its performance.

REFERENCES

- [1] H. Project, "Integrative Approach for the Emergence of Human-like Locomotion." .
- [2] T. Bacek *et al.*, "BioMot exoskeleton — Towards a smart wearable robot for symbiotic human-robot interaction," *2017 Int. Conf. Rehabil. Robot.*, pp. 1666–1671, 2017.
- [3] M. Bortole *et al.*, "The H2 robotic exoskeleton for gait rehabilitation after stroke: early findings from a clinical study," *J. Neuroeng. Rehabil.*, vol. 12, no. 1, p. 54, 2015.
- [4] S. Viteckova, P. Kutilek, and M. Jirina, "Wearable lower limb robotics: A review," *Biocybern. Biomed. Eng.*, vol. 33, no. 2, pp. 96–105, 2013.
- [5] D. Fish, "Pathology Forum: Characteristic Gait Patterns in Neuromuscular Pathologies," *J. Prosthetics Orthot.*, vol. 9, pp. 163–167, 1997.
- [6] Direcção Geral da Saúde, "Circular Normativa: Programa Nacional de Prevenção e Controlo das Doenças Cardiovasculares," p. 24, 2003.
- [7] K. Anam and A. A. Al-Jumaily, "Active exoskeleton control systems: State of the art," *Procedia Eng.*, vol. 41, pp. 988–994, 2012.
- [8] T. Yan, M. Cempini, C. M. Oddo, and N. Vitiello, "Review of assistive strategies in powered lower-limb orthoses and exoskeletons," *Rob. Auton. Syst.*, vol. 64, pp. 120–136, 2015.
- [9] A. M. Dollar and H. Herr, "Lower Extremity Exoskeletons and Active Orthoses: Challenges and State-of-the-Art," *IEEE Trans. Robot.*, vol. 24, no. 1, pp. 144–158, 2008.
- [10] H. Herr, "Exoskeletons and orthoses : classification , design challenges and future directions," vol. 9, pp. 1–9, 2009.
- [11] M. R. Tucker *et al.*, "Control strategies for active lower extremity prosthetics and orthotics: a review," *J. Neuroeng. Rehabil.*, vol. 12, no. 1, p. 1, 2015.
- [12] P. L. Butler and J. P. Jones, "A Modular Control Architecture for Real-Time Synchronous and Asynchronous Systems," in *Applications of Artificial Intelligence 1993: Machine Vision and Robotics*, 1994, pp. 287–298.
- [13] R. Jiménez-Fabián and O. Verlinden, "Review of control algorithms for robotic ankle systems in lower-limb orthoses, prostheses, and exoskeletons," *Med. Eng. Phys.*, vol.

- 34, no. 4, pp. 397–408, 2012.
- [14] K. Anam and A. A. Al-Jumaily, “Active exoskeleton control systems: State of the art,” *Procedia Eng.*, vol. 41, no. Iris, pp. 988–994, 2012.
- [15] M. Kawato, “Feedback-error-learning neural network for supervised motor learning,” *Adv. neural Comput.*, vol. 6, no. 3, pp. 365–372, 1990.
- [16] S. Viteckova, P. Kutilek, and M. Jirina, “Wearable lower limb robotics: A review.,” *Biocybern. Biomed. Eng.*, vol. 33, no. 2, pp. 96–105, 2013.
- [17] T. Yan, M. Cempini, C. M. Oddo, and N. Vitiello, “Review of assistive strategies in powered lower-limb orthoses and exoskeletons,” *Rob. Auton. Syst.*, vol. 64, pp. 120–136, 2015.
- [18] A. M. Dollar and H. Herr, “Active Orthoses for the Lower-Limbs : Challenges and State of the Art,” vol. 1, no. c, 2007.
- [19] A. G. Dunning, “A review of assistive devices for arm balancing,” 2013.
- [20] H. Herr, “Exoskeletons and orthoses: classification, design challenges and future directions.,” *J. Neuroeng. Rehabil.*, vol. 6, p. 21, 2009.
- [21] A. M. Dollar and H. Herr, “Lower Extremity Exoskeletons and Active Orthoses : Challenges and State-of-the-Art.,” *IEEE Trans. Robot.*, vol. 24, no. 1, pp. 144–158, 2008.
- [22] G. Zeilig, H. Weingarden, M. Zwecker, I. Dudkiewicz, A. Bloch, and A. Esquenazi, “Safety and tolerance of the ReWalk TM exoskeleton suit for ambulation by people with complete spinal cord injury : A pilot study,” *J. Spinal Cord Med.*, vol. 35, no. 2, 2012.
- [23] D. Sanz-Merodio, M. Cestari, J. C. Arevalo, and E. Garcia, “Gait parameter adaptation for lower-limb exoskeletons,” *IWBBO*, pp. 18–20, 2013.
- [24] K. A. Strausser and H. Kazerooni, “The Development and Testing of a Human Machine Interface for a Mobile Medical Exoskeleton,” *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 4911–4916, 2011.
- [25] Y. Sankai, “HAL : Hybrid Assistive Limb based on Cybernetics,” *Robot. Res. Springer*, pp. 25–34, 2011.
- [26] M. Bortole *et al.*, “The H2 robotic exoskeleton for gait rehabilitation after stroke: early findings from a clinical study,” *J. Neuroeng. Rehabil.*, vol. 12, no. 1, p. 54, 2015.
- [27] K. N. Winfree, P. Stegall, and S. K. Agrawal, “Design of a Minimally Constraining , Passively Supported Gait Training Exoskeleton : ALEX II,” *IEEE Int. Conf. Rehabil. Robot.*

Rehab Week, 2011.

- [28] G. Aguirre-ollinger, "Learning muscle activation patterns via nonlinear oscillators : application to lower-limb assistance," *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 1182–1189, 2013.
- [29] A. Gams, T. Petric, T. Debevec, and J. Babic, "Effects of robotic knee-exoskeleton on human energy expenditure," *IEEE Trans. Biomed. Eng.*, vol. 60, pp. 1–9, 2013.
- [30] M. Arazpour, A. Chitsazan, M. A. Bani, G. Rouhi, F. T. Ghomshe, and S. W. Hutchins, "The effect of a knee ankle foot orthosis incorporating an active knee mechanism on gait of a person with poliomyelitis," *Int. Soc. PROSTHETICS Orthot.*, vol. 37, no. 5, pp. 2013–2016, 2015.
- [31] A. M. Dollar and H. Herr, "Design of a quasi-passive knee exoskeleton to assist running," *IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS*, pp. 747–754, 2008.
- [32] P. Kao, C. L. Lewis, and D. P. Ferris, "Invariant ankle moment patterns when walking with and without a robotic ankle exoskeleton," *J. Biomech.*, vol. 43, no. 2, pp. 203–209, 2010.
- [33] J. A. Norris, K. P. Granata, M. R. Mitros, E. M. Byrne, and A. P. Marsh, "Effect of augmented plantarflexion power on preferred walking speed and economy in young and older adults," *Gait Posture*, vol. 25, pp. 620–627, 2007.
- [34] P. Malcolm, P. Fiers, V. Segers, I. Van Caekenberghe, M. Lenoir, and D. De Clercq, "Experimental study on the role of the ankle push off in the walk-to-run transition by means of a powered ankle-foot-exoskeleton," *Gait Posture*, vol. 30, pp. 322–327, 2009.
- [35] K. Anam and A. A. Al-Jumaily, "Active exoskeleton control systems: State of the art," *Procedia Eng.*, vol. 41, no. December, pp. 988–994, 2012.
- [36] K. Anam and A. A. Al-jumaily, "Active Exoskeleton Control Systems : State of the Art," *Int. Symp. Robot. Intell. Sensors*, pp. 988–994, 2012.
- [37] J. Cao, S. Q. Xie, R. Das, and G. L. Zhu, "Control strategies for effective robot assisted gait rehabilitation: The state of art and future prospects," *Med. Eng. Phys*, vol. 36, no. 12, pp. 1555–1566, 2014.
- [38] Y. Huang, B. Vanderborght, R. Van Ham, and Q. Wang, "Torque stiffness-controlled dynamic walking with central pattern generators," *Biol. Cybern.*, vol. 108, no. 6, pp. 803–823, 2014.

- [39] J. Cao, S. Q. Xie, R. Das, and G. L. Zhu, "Control strategies for effective robot assisted gait rehabilitation: The state of art and future prospects," *Med. Eng. Phys.*, vol. 36, no. 12, pp. 1555–1566, 2014.
- [40] J. Hidler, D. Nichols, M. Pelliccio, and K. Brady, "Advances in the understanding and treatment of stroke impairment using robotic devices.," *Top. Stroke Rehabil.*, vol. 12, no. 2, pp. 22–35, 2005.
- [41] P. L. Nutler and J. P. Jones, "A Modular Control Architecture for Real-Time Synchronous and Asynchronous Systems," *Exp. Environ. Comput. Vis. Image Process.*, vol. 1964, pp. 157–182, 1994.
- [42] J. Figueiredo, P. Félix, C. P. Santos, and J. C. Moreno, "Towards Human-Knee Orthosis Interaction Based on Adaptive Impedance Control Through Stiffness Adjustment *," 2015.
- [43] J. G. Ziegler and N. B. Nichols, "Optimum Settings for Automatic Control," *Trans. ASME*, p. 10, 1942.
- [44] L. Wang, T. J. D. Barnes, and W. R. Cluett, "New frequency-domain design method for PID controllers," *IEE Proc. - Control Theory Appl.*, vol. 142, no. 4, pp. 265–271, 1995.
- [45] J. Åström, *PID Control*. 2002.
- [46] Yun Li, Kiam Heong Ang, and G. C. Y. Chong, "PID control system analysis and design," *IEEE Control Syst. Mag.*, vol. 26, no. 1, pp. 32–41, 2006.
- [47] R. B. Nimbalkar and T. Loni, "A Review on Significance of PID Controller for Speed Control of DC Motor," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 3, no. 4, pp. 2080 – 2082, 2015.
- [48] R. W. H. Sargent, "Optimal control," *J. Comput. Appl. Math*, vol. 124, no. 1–2, pp. 361–371, 2000.
- [49] B. Houska, H. J. Ferreau, and M. Diehl, "ACADO toolkit-An open-source framework for automatic control and dynamic optimization," *Optim. Control Appl. Methods*, vol. 32, no. 3, pp. 298–312, 2011.
- [50] P. Garrido, "Elementos de Control Ótimo." 2010.
- [51] F. Soares, "Regulador Quadrático Linear RQL (ou LQR)." 2016.
- [52] G. Zames, "Feedback and optimal sensitivity: model reference transformation, multiplicative semiforms and approximate inverse," *IEEE Transit. Autom. Control*.

- Control*, vol. 26, no. 2, pp. 301–320, 1981.
- [53] I. Horowitz, “Invited paper: Survey of quantitative feedback theory (QFT),” *Int. J. Control*, vol. 53, no. 2, pp. 255–291, 1991.
- [54] D. Abramovitch, “Some Crisp Thoughts on Fuzzy Logic,” *Nature*, pp. 168–172, 1994.
- [55] H. Kwakernaak, “Robust control and H^∞ -optimization-Tutorial paper,” *Automatica*, vol. 29, pp. 255–273, 1993.
- [56] D. McFarlane and K. Glover, “A loop-shaping design procedure using H infinity synthesis,” *IEEE Trans. Automat. Contr.*, vol. 37, no. 6, pp. 759–769, 1992.
- [57] S. S. Nair, “Automatic weight selection algorithm for designing H infinity controller for active magnetic bearing,” *Int. J. Eng. Sci. Technol.*, vol. 3, no. 1, pp. 122–138, 2011.
- [58] H. Miyamoto, M. Kawato, T. Setoyama, and R. Suzuki, “Feedback-Error-Learning Neural Network for Trajectory Control of a Robotic Manipulator,” *Neural Networks*, no. 1, pp. 251–265.
- [59] D. M. Wolpert, R. C. Miall, and M. Kawato, “Internal models in the cerebellum,” *Trends Cogn. Sci.*, vol. 2, no. 9, pp. 338–347, 1998.
- [60] M. Kawato and H. Gomi, “A computational model of four regions of the cerebellum based on feedback-error learning,” *Biol. Cybern.*, vol. 68, no. 2, pp. 95–103, 1992.
- [61] M. Kawato, K. Furukawa, and R. Suzuki, “A hierarchical neural-network model for control and learning of voluntary movement,” *Biol. Cybern.*, vol. 57, no. 3, pp. 169–185, 1987.
- [62] V. D. Kalanovic, D. Popovic, and N. T. Skaug, “Feedback Error Learning Neural Network for Trans-Femoral Prosthesis,” vol. 8, no. 1, pp. 71–80, 2000.
- [63] K. Kurosawa, R. Futami, T. Watanabe, and N. Hoshimiya, “Joint angle control by FES using a feedback error learning controller,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 13, no. 3, pp. 259–371, 2005.
- [64] X. Ruan, M. Ding, D. Gong, and J. Qiao, “On-line adaptive control for inverted pendulum balancing based on feedback-error-learning,” *Neurocomputing*, vol. 70, no. 4–6, pp. 770–776, 2007.
- [65] J. Nakanishi and S. Schaal, “Feedback error learning and nonlinear adaptive control,” *Neural Networks*, vol. 17, no. 10, pp. 1453–1465, 2004.
- [66] M. A. Gomes, G. L. M. Silveira, and A. A. G. Siqueira, “Gait-pattern adaptation algorithms

- based on neural network for lower limbs active orthoses," *IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS*, pp. 4475–4480, 2009.
- [67] N. Gopalan, M. P. Deisenroth, and J. Peters, "Feedback Error Learning for Rhythmic Motor Primitives."
- [68] S. Wang *et al.*, "Design and Evaluation of the Mindwalker Exoskeleton," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 4320, no. c, pp. 277–86, 2014.
- [69] A. et al Frisoli, "Positive effects of robotic exoskeleton training of upper limb reaching movements after stroke," *J. Neuroeng. Rehabil.*, vol. 9, no. 1, p. 36.
- [70] J. S. L. Tsai, W. W. Wang, L. C. Hsu, and L. C. Fu, "An articulated rehabilitation robot for upper limb physiotherapy and training," *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2010.
- [71] F. et al Resquín, "Adaptive hybrid robotic system for rehabilitation of reaching movement after a brain injury: a usability study," *J. Neuroeng. Rehabil.*, vol. 14, no. 1, p. 104, 2017.
- [72] T. Nef, M. Guidali, and R. Riener, "ARMin III - arm therapy exoskeleton with an ergonomic shoulder actuation," *Appl. Bionics Biomech*, vol. 6, no. 2, pp. 127–142, 2009.
- [73] C. Carignan, J. Tang, and S. Roderick, "Development of an exoskeleton haptic interface for virtual task training," *IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS*, pp. 3697–3702, 2009.
- [74] A. P. E. Meireles Magali RG, "A comprehensive review for industrial applicability of ANNs.," *IEEE Trans Ind Electron*, vol. 50, no. 3, pp. 585–601, 2003.
- [75] F. Passold and M. R. Stemmer, "Feedback Error Learning Neural Network Applied to a Scara Robot," *Robot Motion Control*, pp. 197–202, 2004.
- [76] Z.-H. Jiang, "Trajectory Control of Robot Manipulators Using a Neural Network Controller," pp. 361–377, 2006.
- [77] A. V. Topalov and O. Kaynak, "Neural Network Closed-Loop Control Using Sliding Mode Feedback-Error-Learning.," *Int. Conf. Neural Inf. Process.*, pp. 269–274, 2004.
- [78] J. L. McClelland, D. E. Rumelhart, and G. E. Hinton, "The appeal of parallel distributed processing." MIT Press, Cambridge, p. pp.3-44, 1986.
- [79] G. F. Luger and W. A. Stubblefield, *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, 2nd ed. Redwood City, California: Benjamin/Cumming Publishing, 1993.

- [80] S. Marsland, *Machine Learning. An algorithmic perspective.* .
- [81] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms.* Spartan, Washington DC, 1962.
- [82] S. Ruder, “An overview of gradient descent optimization,” pp. 1–14, 2016.
- [83] D. P. Kingma and J. L. Ba, “ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION,” *ICLR*, 2015.
- [84] “RASPBERRY PI 3 MODEL B.” [Online]. Available: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. [Accessed: 13-Oct-2018].
- [85] “Ubuntu MATE for the Raspberry Pi 2 and Raspberry Pi 3.” .
- [86] “STM32F4DISCOVERY.” [Online]. Available: <http://www.st.com/en/evaluation-tools/stm32f4discovery.html>. [Accessed: 13-Oct-2018].
- [87] “Free RTOS.” [Online]. Available: <http://www.freertos.org/>. [Accessed: 13-Oct-2018].
- [88] M. Bortole, “Robotic Exoskeleton With an Assist-as-Needed Control for Gait Rehabilitation after Stroke,” 2014.
- [89] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by back-propagating errors.,” *Nature*, vol. 323, no. 99, pp. 533–536, 1986.
- [90] J. Sola and J. Sevilla, “Importance of input data normalization for the application of neural networks to complex industrial problems,” *IEEE Trans. Nucl. Sci.*, vol. 44, no. 3 PART 3, pp. 1464–1468, 1997.
- [91] J. F. Veneman, R. Kruidhof, E. E. G. Hekman, R. Ekkelenkamp, and E. H. F. Van Asseldonk, “Design and Evaluation of the LOPES Exoskeleton Robot for Interactive Gait Rehabilitation,” *IEEE Trans. NEURAL Syst. Rehabil. Eng.*, vol. 15, no. 3, pp. 379–386, 2007.

