Control and Development of a Robotic Cycling Trainer

Patrícia Valcarcel Lopes

Thesis to obtain the Master of Science Degree in

Biomedical Engineering

Supervisors: Prof. Jorge Manuel Mateus Martins
Dr. Giacomo Severini

Examination Committee

Chairperson: Prof. João Miguel Raposo Sanches
Supervisor: Prof. Jorge Manuel Mateus Martins
Member of the Committee: Prof. Duarte Pedro Mata de Oliveira Valério

July 2021

Preface

From February to December 2020, the work outlined in this report was completed at the Rehabilitation Engineering and Robotics group at University College Dublin (Dublin, Ireland), under the supervision of Dr. Giacomo Severini and as part of an Erasmus Placement traineeship. This work was co-supervised by Professor Jorge Martins from Instituto Superior Técnico.

Declaration

I declare that this document is an original work of my own authorship and that it fulfils all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

This thesis wouldn't be possible without the support I received from some people. I would like to acknowledge my dissertation supervisor Dr. Giacomo Severini, who received me at UCD with enthusiasm, support, and always available to answer my questions (even the silliest ones). His guidance helped me in all the time of research and writing of this thesis. Besides, I would like to thank Prof. Jorge Martins, who motivated me in the first place to find and work on this project. A special thank you to Andrea that helped me in so many ways during my time at UCD.

I would like to thank my parents, Narciso and Helena, for their friendship, encouragement, and caring over all these years, for always being there for me through thick and thin, and without whom this project (and my academic journey) would not be possible. You never gave up on me, even when I was unclear about my path and intentions, and you were always willing to work with me to find a solution to my difficulties. These years were an adventure for the three of us, and you heard every battle I went through (and even light some candles for me when I was in true despair). You were always patient with me as I struggled to find my way and accomplish my objectives. You never put any pressure on me, even though I took a little longer than expected and attended a few more universities than usual! My academic path was unique, but owing to your unconditional support, I was able to be successful. You were the ones who told me to slow down, that I was working too hard, that you believed in me and my talents, and that you knew (even when I didn't) that I could make it. I would not be here without you, and I will be forever grateful.

To my friends from Mirandela, you have been present in my life since I graduated from high school. We grew up and stood by one other for many years. We never lost touch, even while I was transferring from one university to another; you accompanied me to every party, you took part of every summer plan, and you listened to every piece of news I brought to the table. Throughout the years, you have always been there for me, taking care of me and making me forget about my worries. To my girls from Braga, I can say that you were the best thing that happened to me during those two years I stayed there, when I was always grumpy and dissatisfied with my own journey. You gave me so many wonderful and funny memories, and I am sure I couldn't do it without you. Even though I was following my dreams when I returned to Lisbon, I was heartbroken to leave all of you. And to the friends I made in Lisbon, we met

in the first week of Tecnico and, despite the fact that we all took different paths, we stood together. You helped me, a small-town girl, in adjusting to and feeling at home in the big city. You were always there for me, even when I was away for two years, and when I returned, you made me feel as if I have never left. You encouraged me to follow my heart and work toward my goals, you heard me weep when I was unsure of my decisions, and you encouraged me when I was too tired to do so. But you were also there to celebrate with me during those incredible days when everything was perfect and I was proud of my path, and I couldn't be happier when we used to meet together for lunch every single day. To the rest of colleagues thanks for the partnership and the moments we shared.

To Paulo, my friend and lover, who has stood by my side through my successes and failures. Thank you for reminding me of my strengths and internal motivations, as well as for hearing me countless times in the middle of the night. I am grateful for the strength it took to pull me up after my huge falls and moments of doubt. For not letting me forget to trust, believe, and persevere in the end. For being my company throughout the past few months and listening to me struggle, suffer, and celebrate each step I took while working on this Thesis. You've heard me talk about this project so many times that I think you know as much about it as I do. These years were not easy, but they would be a lot harder without you by my side.

I am so grateful to be surrounded by such wonderful people. Thank you to each and every one of you!

Abstract

Stroke is the second leading cause of death worldwide and a major cause of disability with increasing incidence. Investing in rehabilitation options is crucial for a variety of reasons. Considering that the development of robotic rehabilitation devices could be a solution and that cycling training is an important method to restore walking ability, a cycling trainer was proposed. This work aims to extend a developed system by implementing it in a fitter software (LabVIEW), creating an optimized position controller, implementing a trajectory generation algorithm, and creating a User Graphical Interface (GUI).

A PID control was introduced to the system. A study was conducted to determine the best controller for the device by comparing, through a series of tests, those discovered using the Autotuning Wizard with those discovered using Genetic Algorithm optimization. An algorithm for generating a pedal trajectory from selected points was developed and tested. The implemented GUI was designed with the required steps for session setup and with useful feedback.

Of the 12 controllers tested, one was selected to be included in the system. From the study conducted, the controllers tuned with a step signal and discovered using GA performed best in the tests. A circular trajectory and a foot gait trajectory were used to evaluate the trajectory planning algorithm. The device proved successful in following the circular trajectory but had difficulty following the foot gait trajectory.

In conclusion, the thesis' objectives were met, bringing the device one step closer to being used in gait rehabilitation.

Keywords

Stroke Rehabilitation, Gait Rehabilitation, Robotic Device, PID Control, Trajectory Planning, Graphical User Interface.

Resumo

O acidente vascular cerebral é a segunda causa de morte em todo o mundo e das principais causas de incapacidade com crescente incidência. Investir em opções de reabilitação é crucial por várias razões. Considerando que o desenvolvimento de sistemas robóticos de reabilitação podem ser uma solução e que o treino com bicicleta é importante para restaurar a capacidade de locomoção, foi proposto um treinador robótico de ciclismo. Este trabalho tem como objetivo continuar o desenvolvimento do treinador, implementando o sistema num software mais indicado (LabVIEW), adicionar um controlador de posição otimizado, implementar um algoritmo para geração de trajetórias e desenvolver uma Interface Gráfica de Utilizador (GUI).

Um controlador PID foi introduzido realizando-se um estudo para determinar o melhor. Por meio de testes, comparou-se os encontrados usando o Autotuning Wizard com aqueles determinados usando uma otimização com algoritmo genético (GA). Um algoritmo para gerar a trajetória do pedal a partir de pontos selecionados foi desenvolvido e testado com uma trajetória circular e uma de marcha do pé. A GUI foi projetada com os passos necessárias para a configuração de treino e com feedback útil.

A partir do estudo realizado com 12 controladores, percebeu-se que os que foram encontrados utilizado o sinal de degrau e o GA tiveram melhor desempenho. O dispositivo teve sucesso em seguir a trajetória circular, mas apresentou dificuldades em seguir a trajetória da marcha do pé.

Concluindo, conseguiu-se atingir os objetivos da tese, colocando este dispositivo um passo mais próximo de ser utilizado por pacientes na reabilitação da marcha.

Palavras Chave

Reabilitação de AVC, Reabilitação da Passada, Dispositivo Robótico, Controlo PID, Planeamento de Trajectória, Interface Gráfica.

Contents

List of Figures

List of Tables

Acronyms

ADLs	Activities of daily living
AKROD	Active Knee Rehabilitation Orthotic Device
ALEX	Active Leg Exoskeleton
BWS	Body-Weight Support
BWSTT	Body-Weight Support Treadmill Training
DOF	Degrees-Of-Freedom
EEG	Electroencephalogram
EMG	Electromyogram
FES	Functional Electrical Stimulation
GA	Genetic Algorithm
GT	Gait Trainer
GUI	Graphical User Interface
HaCT	Haptic Cycling Trainer
HAL	Hybrid Assistive Leg
ICRO	Intrisically Compliant Robotic Orthosis
iEMG	Intramuscular Electromyogram
KAFO	Knee-Ankle-Foot-Orthosis
LOPES	Lower Extremity Powered ExoSkeleton
NUVABAT	Northeastern University Virtual Ankle and Balance Trainer
PAGO	Powered Active Gait Orthosis
PBWSTT	Partial Body-Weight Support Treadmill Training
PD	Parkinson Disease
PGO	Powered Gait Orthosis

PID	Proportional-Integral-Derivative control
PV	Process Variable
RGT	Robotic Gait Trainer
RGTW	Robotic Gait Trainer in Water
RMSE	Root-Mean-Square Error
ROI	Region of Interest
SCI	Spinal Cord Injury
sEMG	surface Electromyogram
SP	SetPoint
VI	Virtual Instrument
VR	Virtual Reality

Chapter 1

Introduction

1.1 Motivation

Stroke is the second leading cause of death and a primary cause of disability worldwide with an increasing incidence due to the population aging. [19] In the United States of America, from the 795,000 people suffered with stroke, 26% remain with difficulties when performing activities of daily living and 50% have reduced mobility. [19] In Europe, the reality is not different, stroke has affected 1.5 million people and generated 439,000 deaths. [20] This event has deep consequences in people's lives due to the low rate of mobility recovery and due to an increased risk of poor outcome within the first year after the incident, which, naturally, contribute to their quality of life. [21] Numbers suggest that stroke survivors have limited or no walking capacity after the event, with less than 10% of patients leaving the hospital with capacity to walk independently outdoors. [22]

Stroke rehabilitation is a complex process that involves multiple health care specialties and approaches. [23] A human brain recovers from stroke in three main ways: adaptation, regeneration, and neuroplasticity. The most successful techniques must incorporate at least one of these processes. [23] For plasticity to fully occur, rehabilitation interventions must be task-specific and goal-oriented rather than general and nonspecific movements. In addition, the goal-oriented tasks must be challenging and interesting enough to maintain an individual's attention and increase effort, the task should allow for repetition through multiple attempts. [24] It was observed a strong relationship between the beginning of rehabilitation and the functional outcome. The earlier therapy is initiated, the more effective the treatment. [25] There is good evidence that increased amounts of exercise provide better activities of daily living and walking outcomes. [23]

When trying to recover the walking capacity, the cycling exercise can be a significant tool as it shares a similar kinematic pattern with walking as they are both cyclical; required reciprocal flexion and extension movements of the lower limb; and demand alternative and coordinated antagonist activation. [26] Cycling engages reciprocal movement of the limbs as well as requires coordination of corresponding muscles, can stimulate motor regions in the central nervous system and activates the cerebral cortex improving balance and motor learning, which is directly connected with the increasing of gait ability. [27] For these reasons, the cycling exercise can be seen as a solution and might be an alternative to motor function rehabilitation method. [28]

Post-stroke rehabilitation will increase soon which will cause a stronger pressure on health care budgets since only 3% to 4% of gross spending on health care in Western countries is spent on stroke. Stroke has economic consequences too, in 2015, the European Union estimated the overall stroke cost of 25 billion euros [20] and, by 2010 in the US, the total direct and indirect cost of stroke was 73,7 billion dollars. [21]

There are ethical and economic reasons to invest in increasing stroke rehabilitation options and efficacy. [22] Developing novel robotic rehabilitation devices may be a solution to this problem. Rehabilitation robots have the main advantage of lightening all labor-intensive phases of physical rehabilitation, allowing a reduction in the therapist effort as they no longer need to set the paretic limbs or assist trunk movements. [29] In this way, physiotherapists can concentrate on physical recovery during clinical therapy and to supervise multiple patients during treatments at the same time. [22] Therefore, body weight support seems to be a valuable advantage that robotic rehabilitation can provide in gait recovery. [7] Robots bring the possibility of increasing the training intensity, with an increase in duration and number of training sessions and provide highly repetitive training of complex gait cycles in a safe environment. One therapist alone would be able to train two or more patients at once, resulting in a significant reduction in personal costs. [8] [22] Rehabilitation robots could fill a gap when there are not enough skilled health practitioners to help stroke patients to recover when leaving the hospital or care centers, they could still keep practicing at home in order to achieve even more successful results. [15] It can also support the patients to keep cardiovascular fitness and to restore the adequate level of cardiorespiratory efficiency. [30] [22]

As the cycling exercise has proven to have several health benefits, now the research community starts to looking towards the creation of robotic cycling ergometers, combining the advantages of rehabilitation robotics with the advantages of this exercise. But first, these systems must be understood and further developed in order to not only provide circular motion training but also therapists chosen trajectories and to allow load training.

With this being said, it is natural to think of rehabilitation robots as an important tool to be developed in order to help even more stroke survivors to regain their quality life after the event.

1.2 Scope and Objectives

This thesis aimed to further the development of a cycling trainer device for gait rehabilitation. It was proposed the implementation of an existing system in the LabVIEW software, proceeding with the debugging process and corrections. A position controller for the system was implemented. An analysis of several Proportional-Integral-Derivative control (PID) controllers found using the LabVIEW Auto Tuning Wizard and a Genetic Algorithm with the optimization toolbox from MATLAB was conducted in order to select the appropriate control strategy. These controllers were tested using a series of tests performed with different setpoint signals (step, sinusoidal waves and chirp signals), which allowed us to analyse the system response with each controller and conclude which was best suited to the device.

Afterward, a trajectory planning algorithm was required, keeping in mind one of the system's unique features, which enables the user to build arbitrary trajectories for the pedal by selecting points within the pedal range. With this tool already in place, a set of pre-defined trajectories were suggested and implemented, and the system's response to these trajectories was studied. Within the development of this project, it was also intended to develop a Graphical User Interface (Graphical User Interface (GUI)), where users could define the simulation's trajectory and its characteristics, analyze the already interpolated trajectory, and monitor the system's response.

1.3 Thesis Outline

The present document is divided into 7 Chapters. Chapter 1 is the introduction, which explains the motivation for this project along with the scope and objectives. Chapter 2 seeks to provide the reader with the latest developments in rehabilitation robots, as well as the existing problem of stroke rehabilitation and regular rehabilitation therapies. In Chapters 3 and 4, the methods are described. The system is outlined in Chapter 3, along with its LabVIEW implementation. A study on PID controllers was developed, in which two strategies were used to find suitable controllers. In Chapter 4, an algorithm is introduced to allow the system to generate trajectories from data points selected by the user on the software's front panel, the procedure and a set of pre-defined trajectories are described. The system's Graphical User Interface is presented and its implementation explained, emphasizing new features and providing instructions about the use of the interface. Through Chapter 5 the found controllers in Chapter 3 were then thoroughly analyzed and compared. The results of Chapter 4 are presented in Chapter 6 where the best PID parameters for the system are selected, and the system's capability to follow two different trajectories is investigated. Finally, in Chapter 7, the overall project is reviewed, along with the main conclusions gathered and the future work directions that must be pursued in order to improve. Aside from the chapters mentioned, this document also contains an appendix that provides complementary information to help the reader better understand the results and conclusions.

Chapter 2

Background

2.1 Stroke

Stroke is the name given to a disease affecting the arteries leading to and within the brain, it is an acute compromise of cerebral perfusion or cerebrovascular accident. It is characterized by sudden onset of neurological symptoms, such as paralysis or loss of sensation, resulting from destruction of brain tissue. [28] About 85% strokes are ischemic, meaning a vessel supplying blood to the brain is obstructed, and the rest are hemorrhagic, when a weakened vessel ruptures and bleeds into surrounding brain tissue. [31] [32] [33] While ischemic stroke is the most frequent, hemorrhagic stroke is the most common cause of death and lost disability-adjusted life years. [22]. Stroke is the second leading cause of death and a primary cause of disability worldwide with an increasing incidence due to the population aging. [19] Concerning mobility recovery, a 2008 study showed that 50% of patients with stroke leave the rehabilitation hospital on a wheelchair, less than 15% are able to walk indoor without aids, less than 10% are able to walk outdoors, and fewer than 5% are able to climb stairs. [22] Of the 795,000 new stroke sufferers in the United States, 26% remain disabled in basic activities of daily living and 50% have limited mobility due to hemiparesis. [19]. In Europe, stroke accounts for 1.5 million diagnosed individuals and 438.000 deaths. [20] With respect to Portugal, in 2016, there were 18,659 hospitalizations for ischemic stroke and 4,785 hospitalizations for hemorrhagic stroke. [21] Regardless of the low rate of regaining mobility, stroke patients have an increased risk of poor outcome in the first year after the event in relation to other factors that naturally affect quality of life. These factors include re-hospitalization (33%), recurrent event (7 to 13%), dementia (7 to 23%), mild cognitive impairment (35 to 47%), depression (30 to 50%), and fatigue (35% to 92%). [34]

Taken together, these numbers suggest that stroke survivors have limited or no ambulation even after rehabilitation, so there is a continuing need to improve the effectiveness of gait rehabilitation for stroke survivors. [35] There is also an urgent need for approximately 350,000 rehabilitation technicians, with only less than 20,000 professionals currently working in rehabilitation. [36]

In fact, post-stroke rehabilitation will soon increase, putting more pressure on healthcare budgets. [20] In the United States, the total direct and indirect costs of stroke in 2010 were $73.7 billion and the median lifetime cost of ischemic stroke was estimated at $140,048. [21] Furthermore, according to the Directorate-General of Health in Portugal alone it was consumed 330,5 million euros of pharmacologic supplies related to brain and cardiovascular diseases . [21]

Having this, it is expected that, for ethical reasons in addition to economic ones, it is imperative to increase rehabilitation options for stroke rehabilitation and more effectiveness. [22].

2.1.1 Stroke Rehabilitation Techniques

Stroke rehabilitation has been broadly defined as any aspect of stroke care that aims to reduce disability and promote participation in activities of daily living. Stroke rehabilitation is a process of assisting a person who has been disabled by a stroke; its objectives are to help survivors return to a normal life by preventing deterioration of function through its improvement and to achieve the highest possible level of independence (physical, psychological, social, and financial) within the limits of the persistent stroke impairments. By regaining and relearning daily living skills through rehabilitation, many stroke survivors achieve greater independence in Activities of daily living (ADLs) and improved functional capacity. [24]

Stroke rehabilitation is a complex process that involves multiple medical specialties and different approaches, depending on the nature of the patient's deficits. A Cochrane review provided good evidence that home-based rehabilitation for stroke patients with mild to moderate deficits is feasible in an urban setting and that similar or even better clinical outcomes can be achieved for patients treated in an inpatient and home setting. Research in the past decade has expanded our understanding of the mechanisms underlying stroke recovery and has led to the development of new treatment modalities allowing more patients to survive stroke with varying degrees of disability. [24]

The human brain recovers from stroke through three main pathways: Adaptation, regeneration, and neuroplasticity. [23] Adaptation is the reliance on alternative physical movements or devices to compensate for post-stroke deficits. However, while adaptation is helpful, it may also be harmful to the recovery process because of learned disuse. [24] Regeneration is the process through which neurons and associated cells and circuitry grow to replace the damage caused by a stroke. It is not yet regarded a standard therapeutic practice for stroke recovery since there are still unanswered questions concerning the type of stem cell to use, how to transport it, the dose, and the long-term safety implications. [24] Neuroplasticity is generally defined as a rewiring in the neural network and it's considered to be the main stroke recovery process. Soon after stroke, activation is decreased in cortical areas directly affected by the stroke that are associated with a change in the localization of certain tasks, as movement or talking. As time progresses through the acute and subacute period, these damaged neural networks reconnect in areas adjacent to the affected area which coincide with clinical recovery. Research studies have demonstrated that neuroplasticity is driven by several key principles, and can lead to recovery mechanisms and functional adaptation as it is related to changes in excitatory/inhibitory balance, spatial extent and activation of cortical maps and structural remodeling. [22] [24]

It is critical to determine the right time to begin rehabilitation because the best results are usually achieved within the first few months after stroke, although some recovery may occur later but with limited opportunities to achieve an optimal outcome. In addition, patients with ischemic stroke who were randomized to very early rehabilitation were much more likely to be alive 6 months after the stroke event than those who received standard care. These patients spent less time in the hospital and reported significantly higher quality of life, independence in ADLs, and improved mental health. [37] Regarding the best time to start rehabilitation to promote neuroplasticity recovery, it is thought to be within 3 months of the stroke event. [22] There is no consensus on when to begin rehabilitation after stroke because there are no specific guidelines for early mobilization, which is one reason why patients undergoing thrombolysis have strict blood pressure guidelines imposed on them to reduce the risk of hemorrhage and there is an hesitation to increase physical activity in these patients for fear of increasing blood pressure. Nevertheless, the most effective approach to restoring brain functionality after stroke is still completely unknown, and thus there is an urgent need for further high-quality research to better understand the body's rehabilitation mechanisms. [23]

Regarding the amount of rehabilitation that should be performed, there is good evidence that an increased amount of exercise leads to better outcomes in ADLs and walking, especially when therapy is performed in the first 6 months after stroke. It was established that increasing the number of therapy sessions per day from one to two sessions results in better outcomes. [23]

Generally accepted practice at this time includes consulting therapists for initial assessment of patients within the first 48 hours, using less intensive therapy as determined by rehabilitation teams and tolerated by patients, and, for those who can tolerate it, increasing the intensity of rehabilitation in the outpatient setting.

2.1.1.A The cycling exercise

As mentioned previously, considering stroke patients being mostly incapable of walking again after the event, there is a demand for lower limb rehabilitation methods with the aim of walking recovery. Cycling shares a similar kinematic pattern with walking as they are both cyclical. [26] Furthermore, this exercise engages reciprocal movement of the limbs as well as requires coordination of corresponding muscles, can stimulate motor regions in the central nervous system and activates the cerebral cortex improving balance and motor learning, which is directly connected with the increasing of gait ability. [27] Considering these similarities with walking, cycling leg exercise can be seen as a solution and might be an alternative motor function rehabilitation method. [28]

Apart from the advantage of being similar to the walking exercise, cycling has more advantages as a rehabilitation method. [28] [27] It also can improve muscle strength and facilitates muscle control of the lower limbs, which allows the patient to put more weight on the affected leg when standing, and improves aerobic capacity, strength, and cardiopulmonary function of patients. [28] Apart from the use as a gait rehabilitation method, it can be used to minimize the risk of secondary diseases that derive from patient immobility. [38] Cycling devices can perform a continuous and prolonged session, whereas training capacity can be adjusted to patient's health status, physiologic response, and rehabilitation evolution. [28] Moreover, different intensities of training can be performed just by altering a few parameters (e.g, workload, speed) in the cycling device without changing how the exercise is performed. These features, and the ability to exercise while seated, allow cycling to be accessible to patients in different disease phases (acute, subacute, or chronic). [28]

Different therapies have been used together with cycling leg exercise to improve cycling's potential as a rehabilitation method, such as limb-load cycling, functional electrical stimulation (Functional Electrical Stimulation (FES)) and feedback.

There are 2 main modes of cycling exercise: active and passive. The positive effects of active cycling in stroke patients have been attributed to active contractions of the lower limbs muscles, although passive cycling exercise can trigger sensory inputs, which in turn can be beneficial to recovery. Passive cycling exercise can be used in people with ambulatory dysfunction in the acute rehabilitation phase who are too weak or medically unstable to do motor active movements to regain motor function. [39]

However, according to the analysis of reviewed studies, it is possible to divide the application of cycling into 2 types of rehabilitation methods: motor function rehabilitation and aerobic training, which addresses the preservation of immobility-related secondary diseases as well as the rehabilitation of the motor functions. [28] Considering that the balance control is not a necessity to do cycling exercise, this can be safely used as a rehabilitation method in the early postroke phase, when gait training is not yet possible. [38]

Studies have been made to prove the effectiveness of this exercise, results demonstrated that stationary cycling trained proved to have a positive effect on dynamic balance as measured by using the time to get up and go test, which suggests that it is indeed effective to improve locomotor function similarly to the effectiveness of treadmill exercise in stroke patients. [40] The results can last, as it was shown that patients who participated in the cycling exercise programme achieved better balance and motor abilities immediately after the cycling exercise programme as well as three weeks later compared with patients who participated in regular exercise training. [41]

Concurrent use of cycling with functional electrical stimulation had a positive effect on balance compared with control and, when compared with cycling alone, the effect was higher. [40] When performing early rehabilitation at the subacute stage post stroke, cycling may improve balance and motor performance. Patients who perform cycling exercise for three weeks have a better chance of standing independently at the end of the intervention than those who do not. [41] Even as a complementary rehabilitation exercise to the conventional therapy, stationary cycling exercise led to better balance and gait abilities than the conventional therapy alone. [27]

Although effectiveness is proven for this exercise, currently, the optimal cycling protocol to maximize outcomes is not known. [40]

2.1.2 The necessity for robotic rehabilitation

Bearing in mind that stroke is a major problem with several people around the globe who consequently need to be rehabilitated afterwards, it is understandable that there is a considerable need to develop novel rehabilitation techniques, in particular, to develop rehabilitation robots.

Beginning with the most mentioned advantage, a rehabilitation robot can lighten all labour-intensive phases of physical rehabilitation, allowing a reduction in the therapist effort as they no longer need to set the paretic limbs or assist trunk movements. [29] This helps the physiotherapist to concentrate on physical recovery during clinical therapy and to supervise multiple patients during treatments at the same time. [22] Therefore, body weight support seems to be the indispensable condition for facilitating gait recovery with robotic devices. [35] While robotic gait training with these appliances, patients are assisted with partial body-weight support, where a robotic device provides physical guidance to move the patients' legs into a correct gait pattern. [8]

One of the other greatest benefits of robotic gait training is the possibility of increasing the training intensity (i.e. the duration and number of training sessions) and highly repetitive training of complex gait cycles in a safe environment. One therapist alone may execute these demanding therapies and be able to train two or more patients at once, resulting in a significant reduction in personal costs. [8] [22] Robotic gait-training devices are well-suited to provide standardized and task-oriented motor training in order to move the patient's limbs as effectively as possible. This could be a significant improvement over the considerable inter-therapist variability that is quantifiable even among highly experienced human therapists. [42] [22]

When there is a lack of trained and skilled health practitioners for treating stroke, rehabilitation robots can alleviate this gap. [37]

People with stroke are known to suffer from an extremely poor cardiovascular fitness, with a reduction in mobility and a consequent reduction in quality of life. [30] There is also a proven secondary effect related to the body weight support and to robotic rehabilitation which is the possibility of favouring the restoration of an adequate level of cardiorespiratory efficiency, reducing energy consumption and cardiorespiratory load. [22]

Furthermore, the rehabilitation robot can provide measurable and comparable quantitative assessment of the rehabilitation process using sensors embedded into the system. [8] These robots allow online and offline instrumented and objective evaluation of several parameters related to patient performance. This includes, but is not limited to, the evaluation of the patient's success with a biofeedback. [22] As patients' engagement and participation in conventional exercises is considered a key factor to increase rehabilitation performances and thereby boost plasticity. [22] Robots can incorporate other technologies (e.g. This fact highlights the role of complex robotic devices, which are capable of demonstrating the full motion coordination aspects while also enabling to train aspects relating to the simpler movement components. [8] [43]

With this, is important to emphasize that the use of robots should not replace the neurorehabilitation therapy performed by a physiotherapist. Robots, like all technological devices, must be used as aids in the hands of the physiotherapist and never as instruments of recovery per se. [44]

2.2 Robotic devices for lower-limb rehabilitation

The previous section presented the problem of stroke and how it affects people around the world and introduces the need for the development of new rehabilitation devices, namely rehabilitation robots. In particular, robotic rehabilitation devices are typically based on the so-called phenomenon of motor learning, resulting from intensive, repetitive and task-oriented motor activities that require patient's effort and attention. [45] This review focuses on lower-limb rehabilitation robotics, and this section aims to analyze existing devices, limitations of current reality, and future steps.

2.2.1 Existing Devices

Before summarizing the reality of robotic devices that are being developed to help patients to recover their walking ability, it is important to notice that there are several ways of classification. The most cited classification of rehabilitation devices was first introduced in 2011 by Iñaki Díaz et al [1], where it was defined each class according to the rehabilitation principle of the system, however it is also possible to classify these devices according to the way they assist the patient.

According to Iñaki Díaz et al [1], the robotics systems currently developed are grouped as follows:

1. Treadmill gait trainers,

2. Foot-plate-based gait trainers,

3. Overground gait trainers,

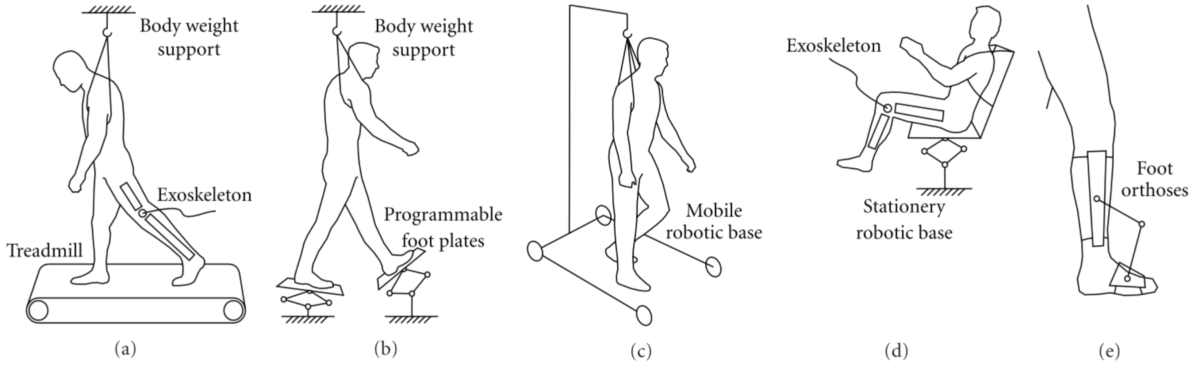


Figure 2.1: Types of lower-limb rehabilitation devices from [1]: (a) treadmill gait trainers, (b) foot-plate-based gait trainers, (c) overground gait trainers, (d) stationary gait and ankle trainers, and (e) active foot orthoses. For more information, the reader is referred to [[1],Figure 1]

4. Stationary gait trainers,

5. Ankle rehabilitation systems, that can be divided into:

    - stationary systems;
    - active foot orthoses.

The visual representation of these systems is possible to be observed in Figure 2.1 and in the next sub-chapters each class will be further analyzed.

2.2.1.A Treadmill Gait trainers

Traditionally, the treadmill rehabilitation technique is known as Partial Body-Weight Support Treadmill Training (PBWSTT) where three therapists assist the patient's legs and hips while walking on a treadmill and, at the same time, the patient's body weight is supported by an overhead harness. This technique demands an intensive work from the therapists as well as a great number of people to assist only one patient at the time. Many robotic systems have been developed on a treadmill to automate and improve this technique as a means to solve these problems. [1]

The treadmill gait approach is a robot device that is connected to the patient's lower limbs, resorting to a set of cuffs, that will help them to follow the proper gait pattern on the treadmill. This robot ensures a precise synchronization between the speed of the patient's gait and the treadmill. Usually, several sensors are implemented into the system in order to evaluate the user' progress and performance and to ensure the operational safety of the device during training exercises. Typically, the robotic devices used in this rehabilitation system are exoskeleton structures, that together with the Body-Weight Support (BWS) system and a treadmill, represent the most important components or robot-assisted gait rehabilitation systems. Considering that, BWS systems are often included in the robot family in an unsuitable manner, the fact that there are intelligent sensors in the system is the key point that allows the differentiation of robots from electromechanical devices. In its turn, exoskeletons are wearable devices that operate mechanically and simultaneously on the human body (although with possible interference with the limb natural movement) and therefore can be classified as assistive devices for human impaired movement or for human power augmentation, according to their application. [45] By 2011 only 3 systems where available in the market: the **Lokomat (Hocoma AG),** the **LokoHelp/LokoStation** and the **ReoAmbulator.** [1] Being, recently, the Lokomat considered the best selling commercial gait rehabilitator and trainer for clinical patients. [46]

The Lokomat is a robotic system consisting of the strong orthosis gait device of the hip and knee joint, the BWS and the treadmill with built-in computer-controlled linear actuators. Gait pattern and guidance force are individually adjustable to the patient's need to optimize the functional training in the sagittal, frontal and transverse planes. This system can also evaluate the physiologic stiffness of the patient's hip and knee joints and the isometric force exerted, respectively, for hip and knee extension. [47] [48] [45]

The LokoHelp/LokoStation (Woodway) is a BWS electromechanical device specially developed for improving gait after brain injury. [1] With this LokoStation, therapists have the ability to conveniently redistribute the weight due to the off-loading system. Offering static and dynamic support which allows different body weight distribution according to the gait cycle phase.

The **ReoAmbulator** (marked in the USA as the ”AutoAmbulator”) is another sophisticated treadmill device combined with advanced robotics that helps patients to replicate normal walking patterns. The device consists of a pair of articulated arms hinged outward to the sides of two upright structures that house the computer control and parts of the body weight unloading mechanism. While the Lokomat BWS system can adapt to the tension and the position of the cable to reduce the dynamic load of the patient's body during the training session, the BWS system used in the AutoAmbulator can only statically reduce the weight of the patient. Each arm is provided with motor-driven pivotal joints at the hip and knee and move with four Degrees-Of-Freedom (DOF), two for the extension and flexion of the knee joint and the hip joint, respectively. The robot arms provide only support in the sagittal plane. The drive motors are computer controlled so that the position, time and distance are monitored to provide a smooth, accurate and coordinated gait pattern according to the variable speeds of the treadmill. Moreover, the ReoAmbulator is equipped with safety interlocks, redundant travel and variable torque limits and other sensors to ensure patient safety during training. The AutoAmbulator has been employed for rehabilitation and educational research studies, demonstrating improved balance and gait, similarly to conventional/manual physical treatment. [2]

Recently, the device developed by the University of Twente has become one of the most well-known exoskeleton devices. The **Lower Extremity Powered ExoSkeleton (LOPES)**, is a robotic device designed to evaluate motor skills and assist stroke patients in walking rehabilitation. Unlike traditional rehabilitation devices as guardians of the leg, LOPES has eight DOF, allowing better movements with less interference and friction and an adequate task-specific gait training. [3] Notably, three actuated rotational joints and a great number of DOF of the exoskeleton may offer a wide range of possibilities to help patients during walking, adapt the support to the individual patient and regain the ability to walk. [45]

Apart from these exoskeletons, there is still available the **Anklebot**, the **Active Leg Exoskeleton (ALEX)** and the **Intrisically Compliant Robotic Orthosis (ICRO)**. The first, also considered as an ankle rehabilitation system, allows ankle movement in all three DOF. In particular, dorsi-plantar flexion and inversion-eversion may be performed via linear actuators mounted in parallel, whilst internal-external rotation is limited at the ankle with the orientation of the foot in the transverse plane. [4] The **ALEX** consists of direct current motors at the hip and knee sagittal plane joints [49], which let the patient's foot move in predefined trajectories [50], playing a major role in stroke patient rehabilitation by increasing patients' gait pattern and walking speeds. [51]. **ICRO** has two passive DOF (vertical and lateral translations), actuated hip and knee sagittal plane rotations powered by intrinsically compliant pneumatic muscle actuators. This device may be also provided with an adaptive impedance control scheme that allows decreasing the robotic assistance when the subjects increase their voluntary participation. [6] [52]

It is worth to say that there is no standard way of categorizing rehabilitation robots, in this way, the treadmill gait trainers can also be considered as stationary walking systems once it is implemented by using a fixed structure combined with a moving ground platform and have been developed to automate traditional therapies. [45]

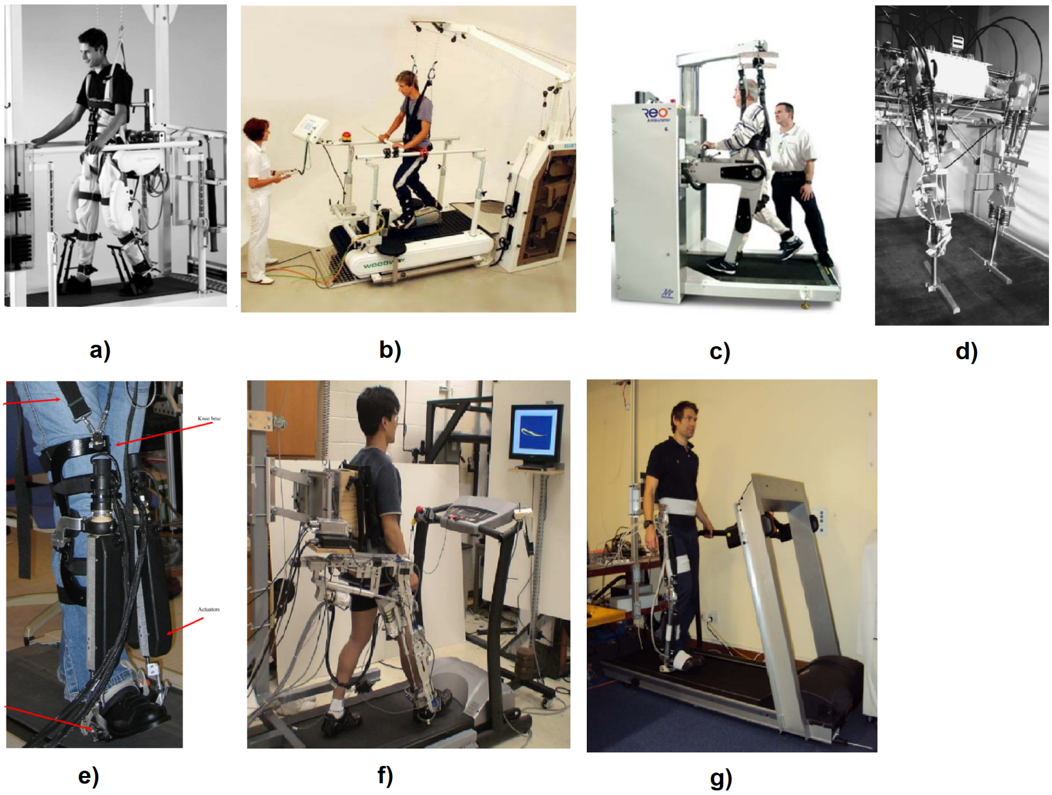


Figure 2.2: Representation of the mentioned treadmill gait trainers. a) Lokomat from [2] b)Lokohelp/ LokoStation from [2] c)Reoambulator from [2] d)LOPES from [3] e)Anklebot from [4] f)ALEX from [5] g)ICRO from [6]

2.2.1.B Foot-Plate-Based Gait Trainers

This class of rehabilitation robots are characterized by the use of footplates to guide the patient's feet and thereby reproduce gait trajectories, these can be seen as an alternative to the treadmill-centred devices, reported previously. These footplates can apply mechanical force to the distal segments of the lower-limbs and so allow the reproduction, without friction and interference, of the stance and swing phases of gait while the patient is on the device. These devices only act on the feet, enabling the thigh and shank to freely move without the friction, thus leading to higher rates of independent walking than exoskeleton-based training. [45]

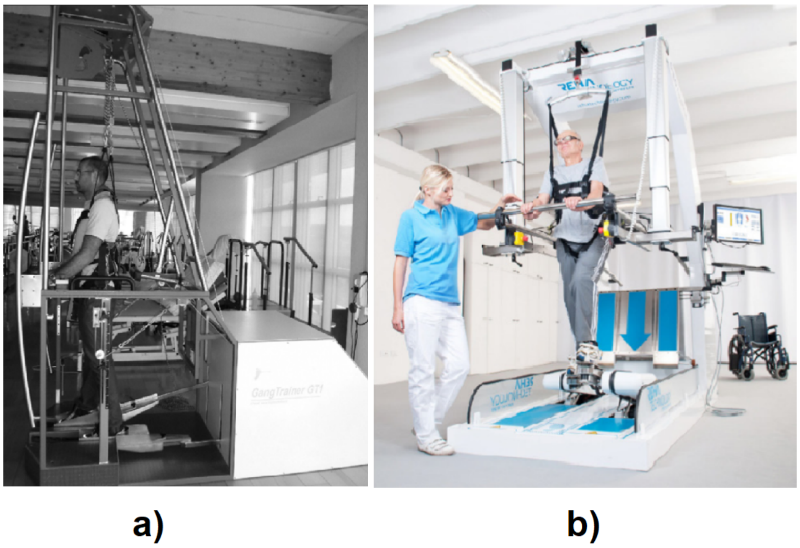


Figure 2.3: Representation of the mentioned foot-plate-based gait trainers, a) GT from [7] and b)G-EO from [8]

In this section, we have the **Gait Trainer (**GT**)** system by Reha-Stim which is a mechanized rehabilitator designed and built for repeated practice of a "physiological model of gait" without the necessary intervention of a therapist. During gait training, the patient is equipped with the harness for BWS and positioned on two platforms that simulate the path while the device is controlling the patient's centre of mass in both vertical and horizontal directions. This robot consists of an automated elliptical bicycle, equipped with six DOF force sensors located under each footplate and Electromyogram (EMG) sensors able to perceive the user's intentions. The feet are supported and bound by two brackets that may be moved to adjust the gait of the patient. [45]

There is the **G-EO system** (Reha-technology) which is able to simulate the repetitive training of relevant situations for locomotion in everyday life such as walking or ascending and descending stairs. Here, the patient's feet are fixed to platforms that, with the help of six engines, can move in all directions (i.e. upwards, downwards, forwards and backwards). [45]

Lokohelp (Woodway) is a device that can be easily placed in the middle of the treadmill, parallel to the walking direction and fixed to the front, simulating the gait pattern and imitating the stance and swing phases. [1] This device can be used in combination with the LokoStation, which allows the BWS, to perform therapies. Locomotion training with this newly developed system is a feasible tool in severely affected patients after brain injury, stroke and Spinal Cord Injury (SCI). [53]

2.2.1.C Overground Gait Trainers

These systems consist of robots that servo-follow the patient's walking motions overground, letting patients move under their own control in place of moving through predetermined movement patterns. It is quite clear that by 2011, nearly all programs have been tested by I. Diaz et al [1] have also been commercialized.

One of those devices is the **ReWalk**, from ReWalk Robotics, which is an orthosis with electric engines that commands the hips and knee joints by a computerized system. This is housed in a backpack in addition to the engines' battery and carried on one's shoulders. This exoskeleton is activated by a sensor, positioned in the anterior upper body and is controlled by the user through leaning of the upper body or small changes in the centre of gravity. [45]

Hybrid Assistive Leg (HAL) is a user-grounded robotic orthosis developed by the University of Tsukuba, Japan, together with the Cyberdyne Systems Company, which helps users in their daily life activities. [45] It has been developed for a wide variety of uses, from recovery to hard work support, and has been produced in many versions. [1]

The **KineAssist** (by Kinea Design), is a mobile robotic base that provides partial body-weight support and postural control on the torso allowing many axes of motion on the trunk and pelvis. It follows the patient's gait overground in forward, rotation and sidestepping directions while leaving the user's legs accessible so they can receive assistance from therapists. [54]

Ekso (Ekso Bionics) represents a front-runner in robotic exoskeleton technology, thanks to its high operability, structural strength, lightness and the ergonomics. It is a self-powered exoskeleton and consists of two aluminium frame legs, a battery and a kind of brackets on which different types of load can be attached and to increase the strength and endurance of the patient. [45] [9]

Recently developed, the **Andago** (by Hocoma AG) is a system where the patient wears a harness, connected through ropes to the frame of the device, that supports his body weight. This robotic frame has two electrically driven wheels and four casters for moving forward, backward, and turning according to the user's intention. [8]

There are already a few solutions regarding the using of soft robotics to develop rehabilitation robots. One example is the **Robotic Gait Trainer in Water (RGTW)**, created in 2008, designed for the development of an underwater gait training orthosis consisting of a hip-knee-ankle-foot orthosis with pneumatic McKibben actuators performing as the actuation system. The angular motion used for the control system was determined by a healthy subject when walking underwater to achieve repetitive physiological gait patterns to improve gait dysfunctions. [2]

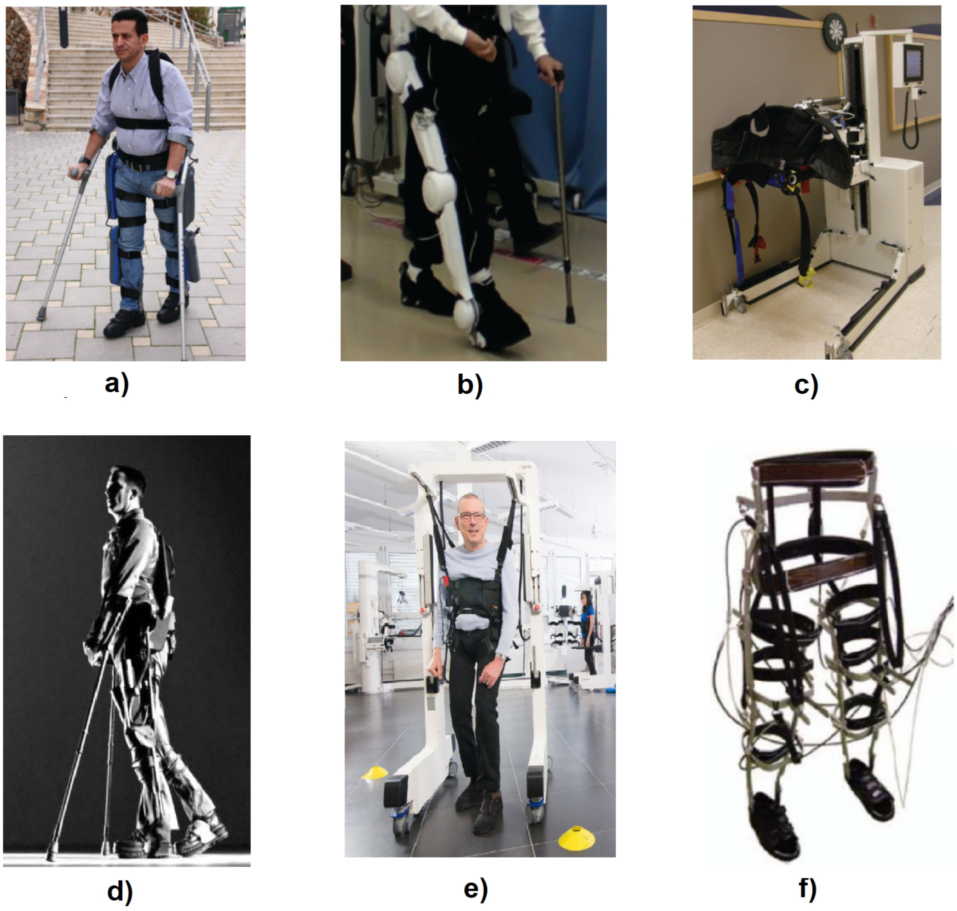


Figure 2.4: Some of the currently developed overground gait trainers. a) Rewalk from [1] b)HAL from [1] c)KineAssist from [1] d)EKSO from [9] e)ANDAGO [8] f)RGTW from

2.2.1.D Stationary Gait Trainers

These systems have the purpose of obtaining efficient muscle strengthening and endurance development, as well as joint mobility and movement coordination. [1]

The **MotionMaker** (by Swortec SA) is a stationary training system which allows to carry out fitness exercises with the active participation of the paralyzed limbs. The patient's limbs are connected to the foot orthosis to activate the natural ground reaction forces. [1] Moreover, the **Lambda,** is a rehabilitation fitness robot used for mobilization of lower extremities, providing their movement in the sagittal plane, including an additional rotation for the ankle mobilization. [10]

Due to the non-standardization of the robotic classes, Calabro, et al (2016) [45] defined the foot-plate-based trainers as stationary systems.

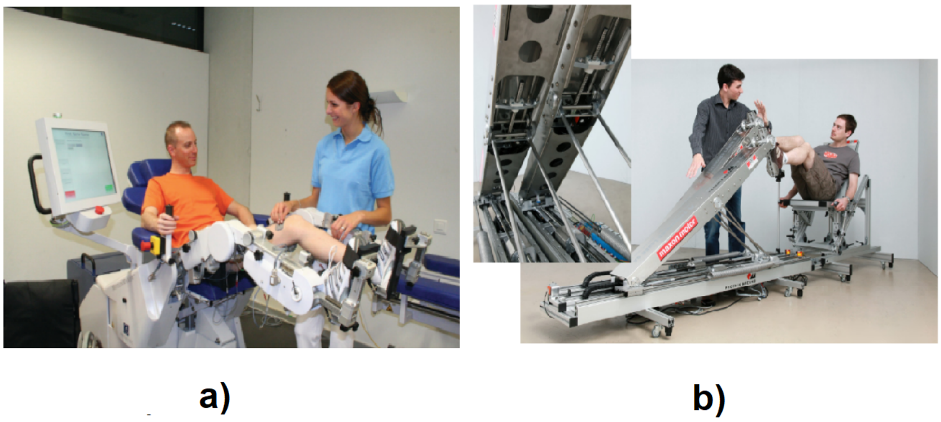


Figure 2.5: Stationary Gait Trainers: a) MotionMaker from [1] and b) Lambda from [10]

2.2.1.E Ankle and Knee Rehabilitation Systems

After the stroke event, the patient with neurological impairment can have a reduction or even no muscle activity in the knee and ankle area leading to the so-called *drop foot*, where there is the inability of an individual to lift the foot. Besides, the ankle motion is considered to be a quite complicated process, due to the presence of complex bone structures. With this, there is the necessity to focus on the approach to the rehabilitation of the knee and ankle, and so, the *stationary systems* and the *active foot orthoses* were developed. [1]

Stationary Systems are the robotic mechanisms intended to exercise the human ankle/knee motion without walking. [1] The **Rutgers Ankle,** was the first of this kind, consisting of a Stewart platform-type haptic interface with 6 DOF resistive forces applied on the user's foot according to virtual reality-based exercises. [11] Also, the Instituto Italiano di Tecnologia has developed a **High Performance Ankle Rehabilitation Robot**, which allows plantar/dorsiflexion and inversion/eversion using an improved performance parallel mechanism that eliminates singularities due to actuation redundancy and greatly enhances the workspace dexterity. [12] The **Active Knee Rehabilitation Orthotic Device (AKROD)** supports the knee joint with variable damping controls to promote the regeneration of the motor recovery and other neurological conditions in patients with knee injury. [13] The **Northeastern University Virtual Ankle and Balance Trainer (NUVABAT)** rehabilitation system is a low-cost, lightweight, mechatronic rehabilitation platform for performing ankle exercise in sitting and standing positions as well as weight shifting and balance testing in standing positions. [14]

Active foot orthoses, on the contrary to stationary systems, are actuated exoskeletons wore by the user while walking overground or in a treadmill. Two attempts to develop such systems were the **Powered Gait Orthosis (PGO)** [15] and the **Powered Active Gait Orthosis (PAGO)**. [16] [1] By 2011, the only commercialized system for rehabilitation was the **Anklebot** (by Interactive Motion Technologies, Inc), an ankle robot developed at the Massachusetts Institute of Technology to rehabilitate the ankle after stroke. This system allowed a normal range of motion in all 3 DOF of the foot relative to the shank while walking. [55]

There are soft-robotics solutions too, is the case of **Knee-Ankle-Foot-Orthosis (KAFO)** which is an orthosis powered by artificial pneumatic muscles during human walking [17]. The **Robotic Gait Trainer (RGT)** developed in the Human Machine Integration Laboratory at the Arizona State University is also a soft-robotic solution as a walking device meant to be used on a treadmill. [56] It is naturally compliant due to the spring in muscle actuators and can achieve a more natural gait by allowing the patient's ankle joint to move in eversion, inversion, plantarflexion, and dorsiflexion. By 2013, it was elaborated a six DOF robotic orthosis which implemented four pneumatic muscle actuators arranged in two pairs or antagonistic mono-articular muscles at the hip and knee joint angles. This device had the goal of encouraging patients' voluntary contribution to the robotic gait training process. [2]

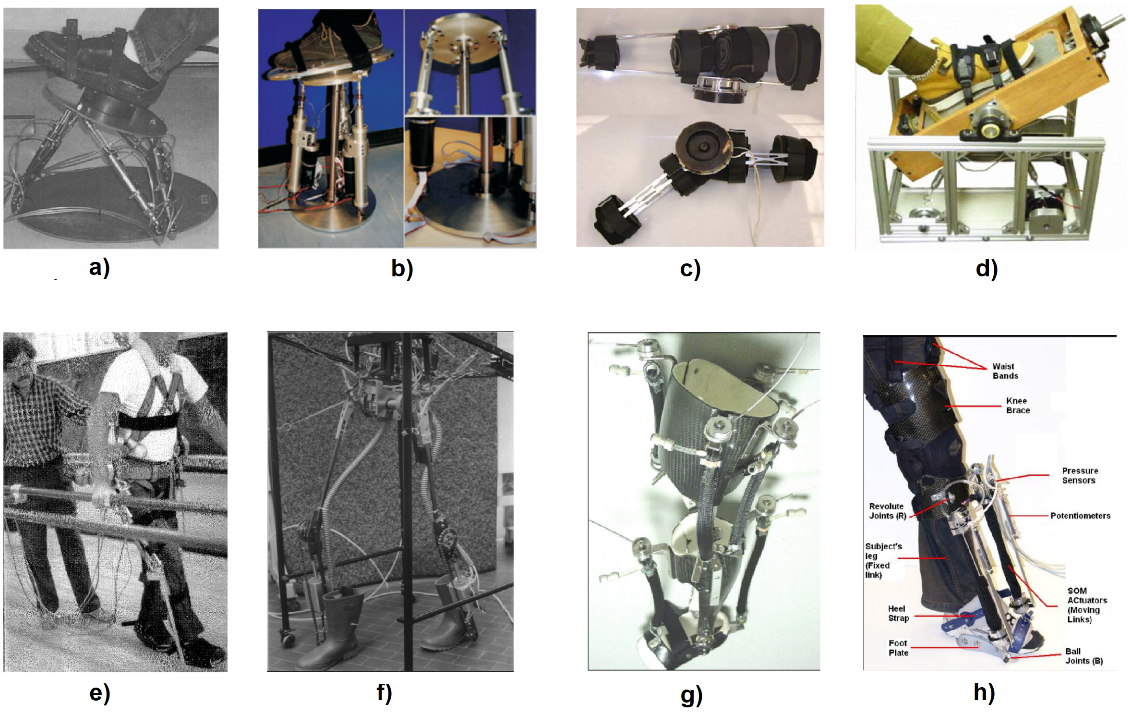


Figure 2.6: Ankle and Knee rehabilitation devices mentioned in this section. a) Rugters ankle [11] b)High Performance ankle rehabilitation robot [12] c) AKROD [13] d)NUVABAT [14] e)PGO [15] f)PAGO [16] g)KAFO [17] and h) RGT

2.2.2 Control Strategies

When talking about rehabilitation devices, two aspects had to be taken into consideration: the mechanical part of the device and the control system. [57] The main purpose of control strategies is to engage movement sequences to assist in patient's gait recovery, physically move the user's legs in a normative gait pattern during walking an to establish a strategy similar to the exercises performed by therapists. [9] [8] The most important issue is hence developing control strategies that entail physical interaction with the patient's limbs. [57]

Usually, control strategies can be generally divided into four categories: position tracking control, force and impedance controls, adaptive control and bio-signals based control. [57]

2.2.2.A Position Control

The position control method is a trajectory-tracking control, intending to drive the lower limbs to gait on the fixed mode. The gait is formed by a proportional position feedback controller and joint angles which are suitable for lower limb muscle strength. [58] Being the main purpose of rehabilitation training to restore the lower limb motor functions to normal levels, is, therefore, required to have a normal gait pattern as a reference to the control system, as a training goal and as a rehabilitation evaluation standard. [59] [60] When a patient has hemiplegia or physical disabilities and finding his normal gait data is difficult, then, a predetermined trajectory obtained from data of healthy gait is often used. [59] Position controllers are important in the first phases of rehabilitation when ”passive” mode is required to help the impaired limb achieve continuous and repetitive training. Nevertheless, the primary issue to be approached in position control is how to generate proper trajectories. [57]

Commonly, the desired trajectory is determined from normative trajectories as pre-recorded trajectories from unimpaired volunteers or pre-recorded trajectories during therapist-guided assistance. There are other strategies: one consists in adapting the new trajectory-based in contact forces between the device and the limb, or including a re-planning (the minimum jerk) desired trajectory at each time from the actual performance of the participant, or adjusting the replay-timing of the requested trajectory from time sample to time sample according to the difference between the actual measured state of the participant and the desired state. [61]

Rehabilitation robots are a dynamic and uncertain system, consequently, it can be hard to achieve ideal results by using ”model-based” controllers, even though an additional controller can be used to compensate for modelling errors. [57] The main limitation of the trajectory tracking control is the decrease in motor learning due to the physically guiding therapy. This guidance changes the dynamics of the task leading to a difference between the trained task and the target, which not fully obey the motor learn rationale that training needs to be specific. Additionally, guidance reduces the patient's physical effort and the burdens on the patient's motor system to discover the principles needed to perform the task successfully. [62] Marchal-Crespo and Reinkensmeyer [61] have summarised this phenomenon as ”Slacking Hypothesis” which means that a rehabilitation robot could decrease recovery as a decrease in motor output, effort, energy consumption and/or attention. Which is something to have in mind when designing a novel robot rehabilitation control, thus, recently, bio-cooperative control was introduced to control and encourage patient’s participation both physiologically and psychologically. [62]

2.2.2.B Impedance Control

Impedance control is more than a given control law, is a control approach which aims to determine the dynamic behaviour of the robot. It provides an assistive force when the limb diverged from the desired gait trajectory, and a restoring force, that is proportional to the deviation. [9] It focuses on allowing flexibility from the rehabilitation robot, which avoids excessive force between the device and the patient’s limbs. It is implemented to engage the subject in voluntary work during rehabilitation by encouraging active participation and allowing patient’s natural variability and comfort, which can address the position control failures described earlier. [9]

From the approach of implementation, impedance control can be divided into two categories: one is based on torque and the other is based on position. The former is based on forward-facing impedance equations, but the explicit expression of impedance equations may not exist in the control structures. The latter is based on the admittance control, which is the reverse impedance equations. [58]

There is a new challenge regarding impedance parameters, as it can make the robot reveal different compliance. When there are low impedance levels, there is the increased risk that the patient will move beyond the physiological range of motion, alternatively, when there are high impedance levels it will force the patient in a passive state and hardly achieve active training. As impedance parameters should not be always fixed, to maintain the balance between low and high impedance, adaptive methods are necessary to guarantee the dynamic performance of robots. [57] On top of impedance control, it is possible to implement algorithms to adapt the desired values of the robot impedance. So, an impedance controlled system can be seen as consisting of up to three levels: on top, a level that determines the values of the desired impedance according to the rehabilitation strategy; in the middle, a level that generates reference torques (or positions in the admittance control) according to the desired impedance; and on the bottom, an inner loop in form of force or position controller that acts directly on the actuators. In addition, this lower level may include compensation components for inertial, velocity and gravitational effects. [63]

Since humans show movement variability, a deadband is usually introduced into the control schemes to allow healthy variability without the robot increasing its assistance force. There is a variance of the impedance controllers, such as the triggered assistance, which allows the patient to attempt a movement without any guidance from the device, but initiates some form of impedance-based assistance after the performance reaches a determined threshold. The contrasting approach exists, where the triggering assistance is provided when the patient is below a threshold for a fixed time. If the patient can not finish the task, then the robot will assist the participant to finish the task at a constant speed until the position error is below a threshold. [61]

There are some limitations that should be considered when using impedance control. First, there is the problem of low impedance levels increasing the risk of the subject and robot start walking out of phase and the necessity of the impedance level to match the patient's capabilities and progress, which can vary broadly throughout the rehabilitation process. Different algorithms have been proposed to avoid the coordination problem, where the reference pattern of the robotic controller can be accelerated or decelerated according to the difference between the robot's and patient's gait. [64]

2.2.2.C Hybrid Control

In hybrid position/force control, the interaction between human subject and device plays an important role. Devices with this control allow the patient to experience different mobility patterns, helping his nervous system to learn trajectories and allow passive and active exercises, that can help strengthen patient's muscles and enhance recovery. [46] [57] A clear advantage of these devices is the possibility to control the robot to move along the desired trajectory and still maintain a specific human-robot contact force. It is aimed to provide a safe, comfortable and flexible place to healing and for the rehabilitation process. [58]

This control strategy was developed considering that sometimes it is necessary to control the position of the robot in some directions, while in other directions it is necessary to control the interaction force between the device and the patient. Thus, when contacting with the patient, the robot's task space is separated in two subspaces, the position and the force subspace, and it completes the tracking control over position and force in the corresponding subspace. As it helps patients walk freely, requiring the active and full engagement of the patient, it is believed to accelerate the recovery process. [58]

However, it only allows the user to apply a certain resistance force along a fixed trajectory and do not allow voluntary active movements from the patient. [57] Recently, adaptive controllers with forgetting factors [[65], [66]] have been suggested to systematically reduce the feedforward assisting force to achieve when tracking errors are minimal. Another approach to adaptive assistance is to use a system for optimisation. In the patient-cooperative framework, the robot aims to reduce real-time torques of human-robot interactions. [61]

2.2.2.D Bio-signals based control

Bio-signals enables the robotic devices to be controlled in a more natural form by using electromyography (EMG) signals recorded from the user's muscles, as it has been proven that there is a correlation between EMG signals, limb movement and muscle activities. [67] [57] In case of patients with mobility impairments, it is rather complicated to perform diagnoses, so these signals are used as feedback. [46] The EMG signal is the electrical activity produced by the skeletal muscle and it is mainly composed of surface Electromyogram (sEMG) and Intramuscular Electromyogram (iEMG). sEMG is obtained by adding electrodes to the skin's surface, while the iEMG is obtained by inserting a needle electrode under the skin into the muscle tissue under the skin. The Electroencephalogram (EEG) is the electrical activity of the brain, which is collected by electrodes attached to the scalp, and it represents the voltage fluctuations caused by the flow of ions between the neurons in the brain. [58]

Mostly used interactive controlling for lower-limb rehabilitation robotics use the surface electromyogram (sEMG) and electroencephalogram (EEG), due to the non-intrusive way of getting both signals, because there is no need of medical expert and its performance can get guarantees. [58] The most important advantage of an interactive control based on EEG is that it is limited to the extent of physical disability, meaning that even if the patient has completely lost the motor function of the limbs, considering that the brain can still produce motion control signals, the method is equally appropriate. This is especially important for patients with total spinal cord injury, because their brains normally work, but the control signal transduction pathway is cut off, so that muscles do not obtain the information required to be controlled. [58]

There has been more interest in creating devices with EMG signals incorporated and to be used throughout the therapy work cycle, which provides considerable amounts of data to be analysed to prove efficacy. [46]

2.2.2.E Training modalities

The effectiveness of robot-assisted rehabilitation depends substantially on the ability to assist a patient's movement in the appropriate way for different recovery stages. As the rehabilitation process is divided into three stages: preliminary, intermediate and advanced stage, where patients will gradually regain the range of motion and strength at the injured limb, therefore, the patient needs to perform passive and active exercises which are adapted to theirs recovery phase. [57] The training modalities are a strategy to accomplish this goal of providing an adequate kind of exercise to each phase of the patient's recovery.

Training modalities used in robots for stroke rehabilitation are often divided into four groups, namely:

- Passive;
- Active;
- Active Assistive;
- Active Resistive.

Passive exercise can improve the movement ability, maintain range of motion, and reduce muscle atrophy, through repetitive exercise. [68] [58] Usually it is used in the preliminary stage, when the patient has very low muscle strength which is not sufficient to complete active movements and thus should perform passive training, like moving the injured limb along a predefined trajectory with help from the therapist or exercise in a common rehabilitation device. At this stage, we can only rely on the help of external forces to help the patient achieve the training goals. The robot's legs should move the patient's legs for rehabilitation training and the lower-limb rehabilitation robot must provide sufficient strength for passive training. [68] This model has the advantage of being efficient even when the patient lacks motivation. [58] Passive exercise is a consistent modality with the **"robot-in-charge"** control allowed in the LOPES rehabilitation robot, [57]

Active exercise is a modality applied since the intermediate stage of recovery and allows the patient to move by his own effort without external assistance and resistance, as the patient already has a certain muscle strength and the active motion of the smaller torque can be performed on the rehabilitation device. At this point, the robot is required to perceive the patient's state and strength/torque while following the patient's movements.The model for active exercise can be adapted to the patient's intention, which enhances the patient's motivation. [58] LOPES rehabilitation robot has a control consistent with this exercise called **"patient-in-charge"**, where the user can walk freely within the device. [57]

Active assistive exercise is also applied since the intermediate stage of recovery when the patient moves his limb following desired movements. At this stage, the muscles have strength but not sufficient to be fully trained without help from the robot legs. [58] This strategy requires external, physical assistance to aid the accomplish of patients' target movements, it is comparable to the practice of therapists who regularly manually apply this technique to provide "assistance-as-needed." [61] Following this, the assistive control strategies are based in the underlying idea that the robot should only intervene if the user moves along the desired trajectory, and when this condition is not met, the robot should create a restoring force. [68] LOPES robot has an equal mode called **"therapist-in-charge"**, which is conducted between the patient-in-charge mode, and that considers the patient's own walking efforts and the robot assistance level to select the most appropriate torques to apply to the leg-joints. [57]

In **active resistive mode**, the robot provides force opposing the movement of the patient. This exercise has the goal of strengthening patient's muscles which are already able to move in the full range of motion. Different motion and amount of resistive force can be implemented to isotonic, isometric or isokinetic exercises, where the robotic leg provides a specific force, which is opposite to the direction of the leg. [58] This model is suitable to patients with high recovery, as patients in the advanced rehabilitation stage, where the objective of the exercises is to allow the patient to perform activities of daily living. [68] [58] It can also be called the **"challenge-based"** controllers, since these controllers make, in some ways, the task more difficult or challenging, rather than making the task easier as the other controllers do. [61]

In addition to the standard training approaches used in robot-assisted recovery, recently designed modalities have also been introduced for their unique robotic architectures or special training purposes. [57] There is the trend of robotic-assisted therapy to develop training styles by incorporating traditional therapeutic activity forms offered by the trainer. [57] A final field of development of robotic therapy control algorithms is mobile robots, which don't contact with patients but instead work alongside them, leading and facilitating rehabilitation tasks. [61]

2.2.3 Evidence

It is now time to analyse the proven scientific evidence for lower-limb rehabilitation robots.

Beginning with a systematic review by M. Bruni et al [69] concluded that the use of robotic-assisted therapy enhanced motor control in stroke patients, yet combined with traditional physical therapy, which is consistent with the finding reached by Mehrholz and Pohl [70] in their 2017 report. Besides, their research found that the sooner the training began, the better the recovery outcomes, because patients with a subacute stroke who were equipped with electromechanical devices in conjunction with traditional physiotherapy care obtained better results (in terms of the the transition to independent walking) than those receiving the conventional physical therapy. The reason to justify the outcome is that the intervention in these robotic systems offers the possibility of carrying out more rigorous, repetitive and task-oriented training that convention-specific over-ground walking trainings would not be feasible. However, no clear signs of a better robotic treatment than convectional therapy were found in chronic stroke patients. [69]. Since the review of data from patients who received traditional therapy revealed that motor skills were enhanced while functional improvements remained low, there was a possibility that robotic rehabilitation would be a successful post-stroke treatment with the best outcomes in the sub-acute phase. [69]

The Cochrane systematic review of 36 trials involving 1472 participants found that the greatest advantages in terms of independence in walking and walking speed can be obtained in individuals who are not ambulatory at the beginning of the study and in those for whom early post-stroke intervention is applied, which is particularly important for robotic devices recommended to be used in early stages of recovery, like the cycling exercise. It has also been noted that there has been no specific description of the frequency, duration and timing of robotic gait training. [70] Several researchers agree that patients can benefit from external support provided by robotic rehabilitation devices until they recover on-the-ground walking capacity without further assistance. [71]

The largest amount of functional and neurological rehabilitation following stroke happens in the first six weeks following stroke. In the light of this, four studies in the sub-acute phase of stroke reported positive effects of exoskeleton training. Two studies [[72], [73]] demonstrated enhanced walking independence with repetitive exoskeletal gait training for more constrained stroke participants, consistent with results using treadmill-based robotics. [35]

A recent systematic review came to the same conclusion as Bruni et al [69] and Mehrholz and Pohl [70], and included the well-known Lokomat in the tests. Notably, only a few studies to date have focused on robotic-assisted gait rehabilitation in chronic stroke, showing that higher intensity training in Lokomat may also improve walking speed in chronic hemiplegia patients with post-stroke. [74] Furthermore, a study by Lo and Triche [75] found that robotic gait training could be more effective in improving walking speed, distance and knee-extensor strength than conventional therapy. Conventional training on the other hand was found to improve the gait speed slightly more than the Lokomat rehabilitation programme. [76] Interestingly, Domingo and Lam [77] have shown that Lokomat is a valid and reliable tool for detecting abnormalities and differences in the proprioception of the lower extremities in people with incomplete SCI, although walking in the Lokomat exoskeleton without guidance changes the temporal step and neuromuscular deambulation control [78] The results of walking on Lokomat were comparable to those of Gait Trainer I, showing increased oxygen consumption (not due to speed) and did not influenced the force control. [79] In comparison with traditional gait training [80], [81], however, significant progress was observed for most of the measured parameters (speed, coordination kinematic support of the paretic limb).

Only a few studies have reported on the AutoAmbulator robotic-assisted device, and the clinical benefits of the robot are still under discussion.

LOPES was assessed by Fleerkotte et al [45] who found significant improvements in speed, walking distance, spatio-temporal measures and hip range of motion for chronic incomplete SCI patients.

Individuals affected by SCI and stroke were treated with GT and the results showed a significant improvement in walking speed, resistance and in the activity of gastrocnemius muscle in the case of paresis of central origin, related to plantar flexing of the foot at the ankle joint and bending of the leg at the knee joint. In this regard, Picelli et al. [82] compared the effects of GT rehabilitation with an equal amount of conventional physiotherapy in 41 Parkinson Disease (PD) patients and showed significant improvements in walking ability, gait speed, cadence and tiredness between robotic gait and conventional training. [45]

Hesse and colleagues evaluated muscle activation patterns in subacute stroke patients during walking and stair climbing with and without the G-EO system, showing a comparable muscle activation in both groups, and thus indicating that the device could represent a new therapeutic opportunity. [45] [83]

Goffedo [84] with twenty-six stroke subjects, aimed at comparing the effects of treadmill-based Robot-Assisted Gait training, overground Robot-Assisted Gait training and conventional gait training in stroke subjects through clinical and gait assessments. The results showed a significant improvement in the clinical outcomes in both robotic and conventional therapy treatments.

Although the authors were unable to clarify the reason, patients receiving end-effect training were found to have better outcomes than those submitted to exoskeleton systems. his finding is consistent with the previous study, which only in post-acute stroke patients demonstrated the efficacy of the end-effector GT system, although exoskeletons had contested findings in both acute and sub-acute phases. [69] [85] It was also found that the walking speed at the end of the training period was higher when the end-effector systems were used compared to the exoskeletal ones. [70] Some reviews on robotic gait training highlighted better outcomes in patients using end-effectors compared to stationary exoskeletons for both stroke and SCI; some others found robotic treatment superiority to conventional treatments. All the reviews outlined limitations with varied treatment intensity, frequency, and a small number of objects in the study design, population, and assessment analysis. [86]

Robotic technologies have recently been widespread in rehabilitation, and their efficacy has been demonstrated in clinical studies as indicated above, but there is no evidence available to researchers that robotic technology is more effective in the training of stroke patients. [84] Studies on robotic rehabilitation of gait in patients with stroke in the sub-acute phase found contradictory results, with some authors showing greater improvement in rehabilitation outcomes after robotic training [[71], [87]] and others finding no substantial differences between conventional and robotic care. [86]

Clinical trials have suggested that manual therapy could still be more successful than robotic gait training in both subacute and chronic phases. [80] It is important to consider how different robotic methods respond to various recovery problems and patients, as well as to the needs of all users (patients, therapists and clinicians) in general. [22] As recently reported in Cochrane Systematic Review [70], the identification of patient characteristics most likely to benefit from robotic therapy is crucial.

Most of the studies are aimed at answering the question ”Are robotic devices effective for all kinds of post-stroke patients?” [22] However, Morone et al [71] have highlighted the need to change the question to ”Who are the most effective robotic devices for?” The objective should not be to test the efficacy of all patients, but to have all the options available to improve their efficacy. [22] The key point for the diffusion and correct use of new technologies is to identify the group of patients for whom and the rehabilitation phase during which each type of technology is most beneficial.

2.2.4 Limitations of current technology and future steps

Biomedical engineers should establish a connection between the most recent neurological findings with the specifics of the robots designed for gait rehabilitation, not only for simulating walking patterns and mimicking the therapists but also to favouring and widening the possibilities for gait recovery. The role of the robot should not replace the therapist, but rather provide an instrument in their hands for training many determinants of a multisystem rehabilitation and improving patients' outcomes and skills. [22] This leads to the constant necessity for device and clinical studies improvement to achieve the goal of providing effective and optimal rehabilitation robotics.

While robot-aided gait rehabilitation has been described as a promising strategy for rehabilitation, the practical improvements made during robotic gait training are still limited to date. The efficacy study of powered exoskeletons and other devices in stroke rehabilitation will further strengthen and lead to findings for or against its utilization for gait rehabilitation. The patient's acuity should be considered as well as his functional presentation, furthermore the extent of benefit has yet remain to be determined through high quality research. [35] While it is generally accepted that most spontaneous clinical rehabilitation appears to occur during the first 3 months after the onset of the stroke, various recovery trends can then emerge based on several dynamic causes, such procedures and associated results should be taken into account to help understand where to anticipate healing, prepare the most suitable therapy and assess the duration of rehabilitation, including the robotic one. [69] In literature reviews, how the robot environment is conducted concerning the characteristics of the patient is never specified. We should shift the concept of a common procedure for all patients, and facilitate care personalization, adjusting the robots to the particular state of the patient. [86]

Robot-assisted gait teaching is approximately as effective as traditional Body-Weight Support Treadmill Training (BWSTT) therapy though involving the therapists considerably less physical activity. However, the goal of robotics in therapy is not merely to simplify existing rehabilitation procedures, but to optimize recovery. Therefore, Marchal et al [8], suggested two directions for future research.

The first path is to concentrate on conducting randomized controlled clinical trials in order to rigorously equate robotic control systems with each other and traditional physiotherapies. The specific control strategy chosen for the robot-assisted gait training seems to play a key role in the outcomes of rehabilitation. However, the question of the most successful robot-assisted gait training technique is still wide open. A second approach is to encourage the use of greater detail in determining which preference of technological features (robot form, actuation, control algorithm, etc.) is most desirable for which rehabilitation activities, which kinds of neurological injuries, and at what recovery point. Adjusting the technical features to the particular pathophysiology of the condition, the rehabilitation stage and the particular activity being trained may enhance its therapeutic benefit. [8]

Technology may become a means to trigger motor responses previously not feasible with the physiotherapist's practice alone. [88] If this is valid, we would switch from a concept of repetition or force generation to a concept of motion awareness. In addition to the usual motor regions, brain areas such as the insula, the amygdala, and other neuronal pathways linked to the deepest centers are triggered when a movement is made under robotic control. These centers are crucial in deciding the movement's memory and stimulation. [89]

Exoskeleton-driven motions may become not only an action per se, but also a sensation and a modern motor planning modality that is structurally combined with the reprogramming of the motion control experience according to the concepts of motor visualization and action perception. [86] Intention is just as crucial as action, as psychological states such as motive, inspiration, and determination are considered to be vital to the progress of recovery. [90] Therefore, shifting from the concept of empowerment to the concept of action awareness, followed by action intention and then by motivational and emotional strength is important. The number of repetitions done by a robotic device is an outcome and not the focus of the procedure. [91]

Virtual Reality (VR) is considered to be the most innovative technology and will play an significant role in the neuromotor rehabilitation in the next few years (especially when combined with robotic therapy). In reality, Mirelman et al measured the impact of training on individuals' gait after stroke with a VR robot device relative to a robot alone, finding that the former was superior in enhancing their walking capacity. [92]

Lastly, Dennis R. Louie and Janice J. [35] left us some questions that should be addressed in the future. They indicated that the effect on stroke recovery according to characteristics of multiple exoskeletons, like the number of joints actuated, level of assistance and coordination of stepping should be analyzed. They also questioned how much exoskeletal gait practice is appropriate for patients who have had a stroke in order to regain the most ability to walk. It is important for the future of gait recovery that all of these limitations, questions and suggestions are taken into account in order to keep the progress in the robotic rehabilitation field.

Chapter 3

System Implementation and PID Control

3.1 The Haptic Cycling Trainer

This thesis had the objective of continue to develop the already proposed and built system described in this Section. The purpose of this novel device, its main characteristics, and how the system will operate are all outlined here.

As described before in Chapter 2, there is the need to keep developing robotics rehabilitation devices, in specific, devices that allow for gait recovery of stroke and injury patients. Joining the necessity for innovative devices with the validated benefits of cycling trainer, a new cycling ergometer concept was proposed to be used by the patient as shown in Figure 3.1, where the wheelchair can be placed close to the cycling device allowing to easily adjust the position between the chair and the cycling trainer according to the patient's needs. The cycling ergometer comprises a crank arm with dynamically variable length change with range from 8.5 cm to 24 cm. The aim of this design is for the therapist to be able to build a personalized 360º route based on their knowledge of muscle activation patterns that governs the relative muscle activation timings of the patient's legs and improve motor relearning of gait or similar gait patterns. During cycling exercises, this system can provide visual feedback, loads, and perturbations. The proposed system, which is part of the stationary cycle ergometer, has the advantage of being more accessible than alternative approaches such as treadmills or robotic exoskeletons.

In specific, it was established a few key characteristics for this system as:

- Provide a changing crank length function, accomplished via the crank-arm dynamic length adjustment system;

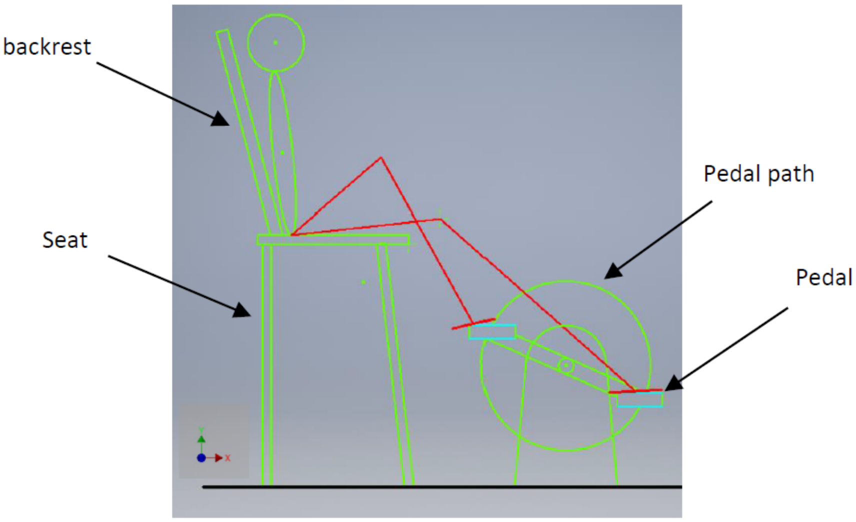


Figure 3.1: Schematic of the patient using the cycling ergometer

- Allow the operator to define a desired non-circular trajectory for each pedal, in order to encourage specified muscular activation patterns or to mimic a particular kinematic output;

- Allow the operator to dynamically change the resistance of the bike fly-wheel of the cycloergometer within each cycle.

Movement control methods were investigated to determine the most appropriate approach to implement non-circular cycle routes, while the current devices used at clinical rehabilitation facilities were explored to determine their modes of use, benefits, and limitations. A clinical specialist was consulted for guidance on current technologies and how gaps could be addressed in potential devices, allowing better visualization of what the system should tackle. A general layout was sketched and modeled using CAD software. Hence, with the information gathered, the Haptic Cycling Trainer (Haptic Cycling Trainer (HaCT)) was proposed.

In Figure 3.2 it is shown how the HaCT is supposed to function, with each crank arm length adjustable independently during the cycle rotation through motors (as shown in the right side of Figure 3.2) in contrast with the currently available cycle ergometers with a constant crank arm length allowing only to perform circular trajectories (as schematized in the left side of Figure 3.2). This mediated asymmetry has the ability to generate gait-like movement exercises. The actuation mechanism will be combined with a dynamic braking system, which will consist of a magnetic brake operating on the cycloergometer's flywheel to dynamically change the cycling load. Force sensors embedded in the pedals will be used to control both the actuation and braking mechanisms as well as provide visual feedback for both patient and therapist.

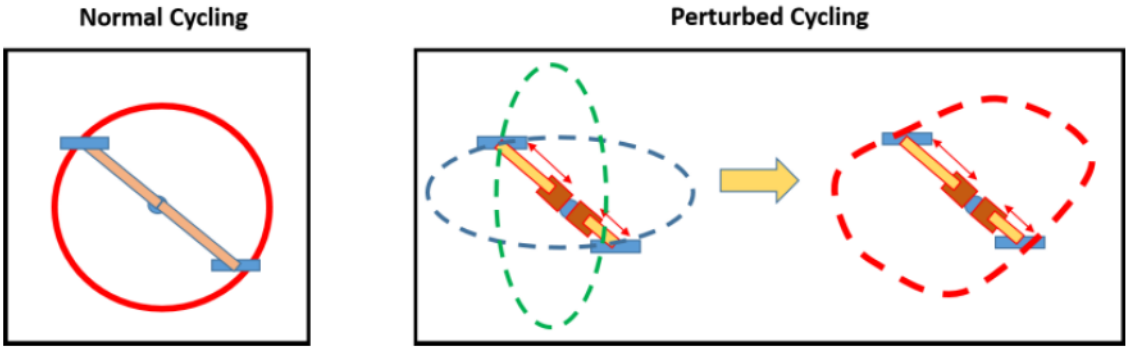


Figure 3.2: Representation of the key difference of the HaCT device and the current available cycle ergometer. On the left is represented the normal cycling functioning and in the right is represented how this system will operate.

The feature of allowing a fully customized training trajectory is achieved through the "crank-arm dynamic length adjustment system" (CADLA system). One actuator will be used for each pedal assembly to dynamically change the length of the crank arm. The rotational power from the pedal will be transferred to the traditional fly-wheel by each pedal assembly, allowing the difference in length between the two pedal assemblies, which will translate into an asymmetry in the cycling path between the two legs.

In addition to the CADLA system, the dynamic braking system (DynB system) will be used to modify the resistance of the fly-wheel of the cycloergometer dynamically inside each cycle. The system will allow for the creation of custom loading patterns that will create a virtual force-field acting against the cycling movement. The force-field, given the bilateral connection between the two pedals, will generate an asymmetry between the two legs. So, responding to this asymmetry, the subject will have to increase the activation of the different muscle groups during specific parts of the cycling pattern. Both CADLA and the DynB systems will be controlled using a single controller based on an impedance-control design, and, both controllers will rely on information from force and position sensors embedded in the pedals and on position sensors embedded in the assembly of the CADLA system.

The whole HaCT system will be controlled using an easy-to-use software that will allow the user to select desired patterns of crank-arm and braking asymmetries. A visual feedback interface will also be implemented and will guide the subjects in producing the desired kinematic and kinetic patterns during each gait cycle by giving them feedback on their performance. The visual feedback will also inform the therapist of the patient's performance.

3.2 Materials

To develop this project it was necessary to have access to the following material:

- MATLAB and Simulink software;

- LabVIEW software;

3.3 LabVIEW Implementation

A previous model of the system was already developed in Simulink. However, it was chosen to continue to develop the project in LabVIEW once it was necessary to use the CompactRio controller, LabVIEW native, as part of the system hardware. Furthermore, the project needs the implementation of a functional and intuitive user interface, which is more suitable with the LabVIEW features. The modeled system is the equivalent of the cycling system, which is composed of three other sub-systems as represented in Figure 3.3. At this step, it was required to implement the system again in LabVIEW. The cycling system is composed of two main sub-systems: the one that describes the behavior of the motor, the gear, and the ballscrew components; and the second that describes the ballscrew's kinematic and dynamic. With these two components, it is possible to calculate the pedal position (in terms of length of the crank-arm) and velocity.

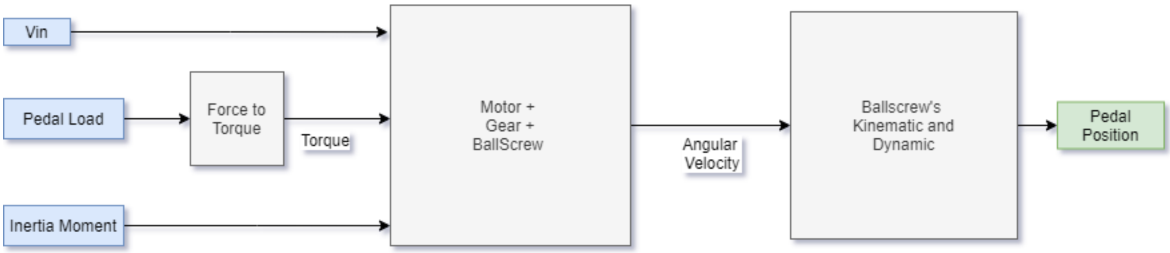


Figure 3.3: Illustration of the Cycling System's blocks that are already implemented in Simulink software. In blue are presented the inputs and in green is represented the output.

In the Simulink model, several mechanical and material characteristics were calculated using a MATLAB script and later used in the block diagram, as the blue inputs presented in Figure 3.3. In LabVIEW, these characteristics are no longer automatically calculated, but alternatively, they are given as an input for the system. In this software, every model has two main files, the front panel which is a simple user interface, and the block diagram which has the block programming and implementation of the system. Here, the mechanical and material characteristics were given as input in the front panel.

In each step of implementation, it was necessary to keep track of possible errors or incompatibilities of both softwares. Each subsystem was first implemented with inputs and outputs checked according to the similar model in Simulink. In the end, intensive debugging was done to check every calculated value and to check if they matched with the ones calculated in Simulink. Throughout the implementation, it was necessary to switch the solvers of both models from automatic step size to a fixed step size of 0.1 seconds, as this was causing significant differences in the calculated data, making comparisons between the two models impossible.

By the end of this part, the blocks presented in Figure 3.3 were correctly implemented in the LabVIEW environment and the model was ready to be enhanced.

3.4 PID Control

A position controller was implemented in closed loop to regulate the pedal position. The Proportional-integral-derivative (PID) control consists of the additive action of the proportional (Kp), integral (Ki) and derivative (Kd) components . It is the most widely used control strategy as the three terms are sufficient to parameterize a structure that permits successful and efficient control for a variety of processes and dynamic systems. Moreover, it has an extremely simple structure which makes it easier to work with and apply intuitive tuning procedures. Adaptation, self-tuning, and gain scheduling can be easily introduced into PID control which makes this a versatile controller. [93]

For this reason, and being this system yet in a preliminary phase, the PID controller was chosen to be implemented in the model, as it is represented in the schematic in Figure 3.4.

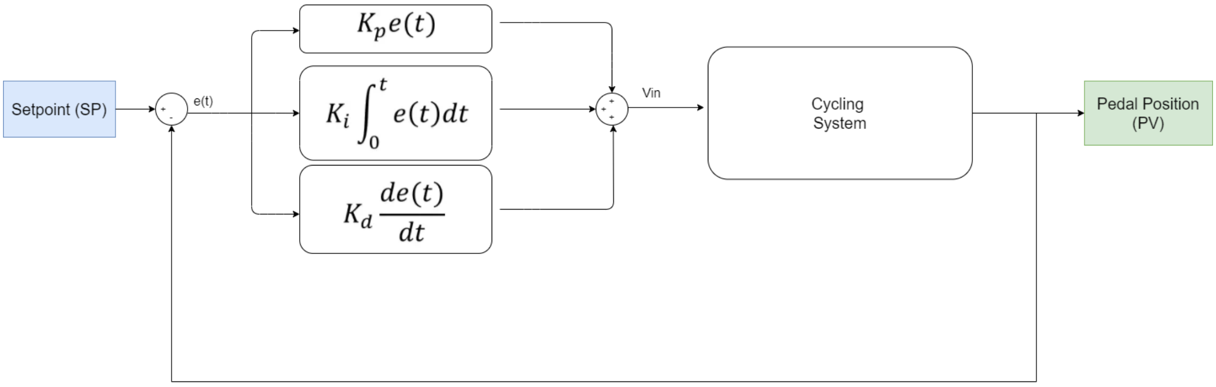


Figure 3.4: Representation of the block system with the PID position controller. The setpoint (SetPoint (SP)) is the input of the system and the output is the Process Variable (PV) which is the pedal position.

Next it was time to do the PID tuning, which was first done through the autotuning wizard available on the software. The goal for the PID controller was to have a rising time fast enough to quickly follow the possible changes in the setpoint signal and to have a smooth behaviour without ringing and significant overshoot. Besides, it was necessary for the controller to allow the system to successfully follow signals with frequencies between 1 Hz and 2 Hz.

3.4.1 PID Autotuning

The LabVIEW software already contains a palette of simple G-language building blocks known as Virtual Instruments (Virtual Instrument (VI)). The “PID Autotuning VI” was available in the Control and Simulation VIs and functions palette, and it was chosen to be included in the model. The “PID Autotuning VI” block implements the basic PID algorithm but also an autotune wizard that performs PID tuning by using Relay Method, allowing autotuning parameters such as “controller type,” “relay cycles,” “relay amplitude,” and “control specification” to be specified. The block allows to find values for Proportional Gain (Kc), Integral Time in Minutes (Ti), and Derivative Time (Td) in return.

Initially, a step signal of 0.2 m of amplitude was defined to find the parameters, based on the range limitations of the system and previous tests done in the Simulink model which was being used to compare results. Then, it was attempted to find a controller using the setpoint as a sine wave of 1 Hz, as described in equation 3.1.

$$setpoint(t) = 0.1625 + 0.0775 \cdot sin(6.2832 \cdot t) \tag{3.1}$$

The autotuning parameters were adjusted in accordance with the VI’s findings in order to achieve the desired response, and after a few tries, a PID controller that was satisfactory in terms of rising time (reaching the target in 0.6 seconds) and smoothness was discovered.

However, further testing of the controller was needed to inspect if it was capable of following signals with frequencies ranging from 1 to 2 Hz. Several experiments were planned and carried out in accordance with the instructions in the Section 3.4.3

3.4.2 Tuning based on The Genetic Algorithm

On the performance results shown in Section 5.1 where the controllers using the Autotuning Wizard were tested, these controllers revealed to be unsatisfactory in terms of stability, exhibiting some oscillatory behavior after settling. It was important to look for additional PID parameters that would allow the system to be stable and have a low error while maintaining a quick settling time. As a result, it was decided to try different tuning methods in order to find an optimal controller.

The Genetic Algorithm (Genetic Algorithm (GA)) is a well-known algorithm, inspired by the biological evolution process that mimics the Darwinian theory of survival of fittest in nature. [94] The GA is an optimization technique developed by Holland in 1962, is successfully used to solve complex problems in several areas for forecasting, information security, or operation management. [95] [94] Some previous papers have presented the application of GA for system identification and PID tuning. [96] For these reasons, it was chosen to tune the PID controller of our model with this new tuning method to find new parameters that would allow the system to have a better performance. The GA optimization was carried out in the MATLAB/Simulink environment, using the system’s previous Simulink model in conjunction with an implemented script.

3.4.2.A GA Theory

The Genetic Algorithm is a population-based meta-heuristic algorithm, which means it searches a large number of solutions. By starting at many independent points and searching in parallel for suboptimal solutions, these algorithms preserve population diversity and prevent solutions being stuck in local optima. [94] [97]

The genetic algorithm starts with no knowledge of the correct solution and relies on responses from its environment and evolution operators such as reproduction, crossover and mutation to arrive at the best solution. The algorithm manipulates not just one potential solution to a problem but a collection of potential solutions, known as population. The potential solutions in the population are called chromosomes. These chromosomes are the encoded representations of all the parameters of the solution. Each chromosome is compared to other chromosomes in the population and awarded a fitness rating that indicates how successful this chromosome is. To encode better solutions, the GA uses genetic operators or evolution operators such as crossover and mutation for the creation of new chromosomes from the existing ones in the population. [97] This is achieved by either merging the existing ones in the population or by modifying existing chromosomes. The selection mechanism for parent chromosomes takes the fitness of the parent into account. This ensures that the better solution will have a higher chance to procreate and donate their beneficial characteristic to their offspring. The objective function assigns each individual a corresponding number called its fitness, which is then assessed and a survival of the fittest strategy is applied. [97] The fitness values of all chromosomes are evaluated by measuring the objective function in a decoded form after an initial chromosome population is randomly generated. As a result of the selection process, a group of the best chromosomes is chosen based on each individual's fitness. The genetic operators, crossover and mutation, are applied to this "surviving" population to improve the next generation solution. Crossover is a recombination operator that combines subparts of two parent chromosomes to produce offspring. This operator extracts common features from different chromosomes in order to achieve even better solutions. Mutation is an operator that introduces variations into the chromosome and occurs occasionally with a small probability. Through the mutation operator, the search space is explored by looking for better points. This process continues until the population converges to the global maximum or another stop criterion is reached. [95] The schematic of the steps taken in a Genetic Algorithm optimization is shown in Figure 3.5.

3.4.2.B GA Optimization Simulations

A research was conducted to find the best controller with the GA optimization. For this analysis, it was decided to run GA simulations to determine the three PID parameters (Kp, Ki, Kd) for a variety of error criteria in order to reduce the error value and, at the end, compare these controllers to find the best fit for our model.

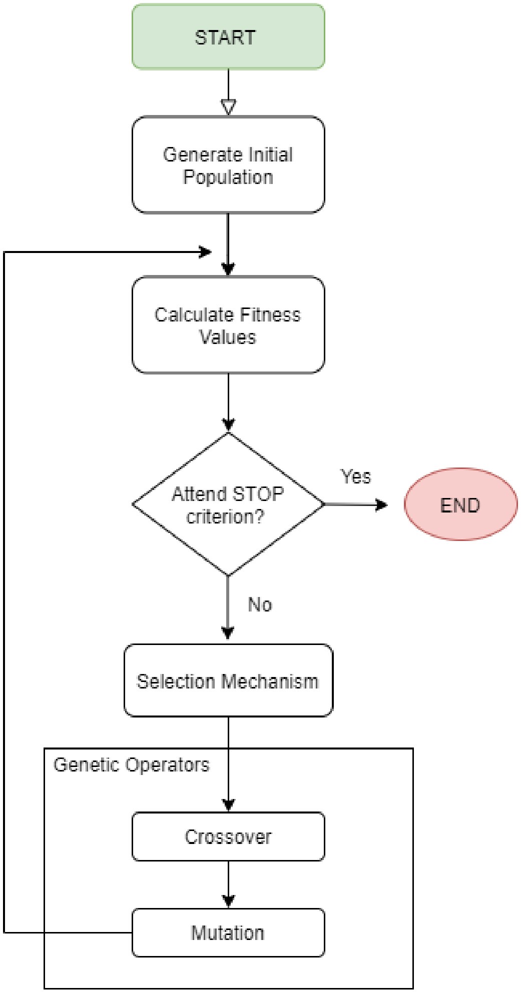


Figure 3.5: Genetic Algorithm flowchart.

In the design methodology of a PID controller, one of the most important performance criterion is the difference (error) between the plant output and the setpoint signal. Using this error criterion as the fitness function of the optimization algorithm results in a small overshoot with a long settling time. In general, fitness functions are based in error equations. The performance indices chosen in our study were: Integral of Time multiplied by Squared Error (ITSE) as described in Equation 3.2, Integral of Absolute Magnitude of the Error (IAE) as in Equation 3.3, Integral of the Square of the Error (ISE) present in Equation 3.4, Integral of Time multiplied by Absolute Error (ITAE) in Equation 3.5 and Mean of the Square of the Error (MSE) in Equation 3.6.

$$ITSE = \int_0^T t \cdot e(t)^2 dt \tag{3.2}$$

$$IAE = \int_0^T |e(t)|\, dt \tag{3.3}$$

$$ISE = \int_0^T e(t)^2 dt \tag{3.4}$$

$$ITAE = \int_0^T t \cdot |e(t)| \, dt \tag{3.5}$$

$$MSE = \frac{1}{t} \int_0^T [e(t)]^2 \, dt \tag{3.6}$$

The GA simulations were run in the MATLAB/Simulink environment, by using the MATLAB optimization toolbox and the previous Simulink model of our system. A script called ”PID_optim” was created to describe the objective function used by the optimization toolbox. The objective function is the function which will be minimized according to the algorithm, here the solution is a set of parameters which will lead to a value. In the written script, the parameters to be optimized were the PID controller parameters (K_p, K_i and K_d).

This script consisted of a function that received the PID parameters generated by each step of the GA, applied these in the PID at the Simulink model, run the Simulink for a defined period of time measuring the error and calculating the performance index, which would be the function's output at the end. To run the the Simulink model from the MATLAB environment it was necessary to define a simulation time, a time step and the setpoint signal. In each iteration of the GA, this script would run to each generated set of parameters to obtain the performance index in order to be evaluated and find convergence to a minimum. Several simulations were done for each performance index, the settings defined at the MATLAB Optimization tool were as the default apart from the ones shown in Table 3.1. The lower bounds were chosen having in mind that the values for the PID gains couldn't be negative. To define the upper bounds which would be best for the GA simulations for every performance index, first, there were made several tests to do a preliminary adjustment to limit values that could lead to better results. If the limit for the upper bounds were too high, the GA would find PID parameters for a slow controller with too much overshoot, so we start with all parameters with the upper bound of 10^{12} and according to results being more closely to the ones desired, the bounds for K_p, K_i and K_d were refined until reached the ones present in the table (3.1). The population size was chosen to be 50 as it was the recommended by the toolbox for simulations with less than 5 variables.

Setpoint: Step Signal

It was studied the effect of the setpoint signal when tuning the PID controller to understand which would lead to better results for the system, this was made by changing the input signal of the system in the Simulink model being used by the script ”PID_optim(x)” which contains the fitness function. The first setpoint signal studied was a step of 0.2m, as the one used for the autotuning with the LabVIEW wizard. In this way, it was found some PID parameters, minimizing each performance index, which were then

Table 3.1: Settings for the GA in the MATLAB Optimization Tool.

Parameter	Value
Fitness Function	@ PID_optim(x)
Number of Variables	3
Lower Bounds	[0 0 0]
Upper Bounds	[10^4 50 50]
Population Size	50

tested according to section 3.4.3. In these simulations the time step was defined as 0.0001 seconds, each simulation in Simulink would run for 1 second and the settings for the MATLAB Optimization Tool were as described in Table 3.1.

Setpoint: Sine Wave

The second setpoint signal studied was a sine wave defined, once again, in the Simulink model used by the script with the fitness function. Here, two steps were employed to find the PID parameters that could conduct the system to behave as required. Firstly, the first simulation was run with a sine wave characterized in Equation 3.1, where the angular frequency was ω=6.2832 rads/s which was approximately 1 Hz.

$$setpoint(t) = 0.1625 + 0.0775 \cdot sin(6.2832 \cdot t) \tag{3.7}$$

In the simulations performed the time step was set to 0.0001 seconds, each simulation in Simulink would run for 10 seconds since it was observed a difficulty for the controllers to converge, and the settings for the MATLAB Optimization Tool were as described in Table 3.1. The results found with this approach were then tested according to the methodology described in Section 3.4.3.

After testing the first set of controllers, it was then necessary to try another approach to find a controller to successfully follow a sine wave with frequencies between 1 and 2 Hz. According to the results for the controllers performance (results can be seen in Appendix A.1), the best controller was the one that minimized the MSE. For this reason, the fitness function that minimizes the MSE was chosen to run simulations to discover PID parameters for different frequencies, starting with 2 Hz and sequentially decreasing to 1.8 Hz, 1.7 Hz, 1.6 Hz, and 1.5 Hz. After these results, and being 1.5 Hz the highest frequency with successful results, the simulations to find PID parameters minimizing the other performance indexes were done using a sine wave with this frequency. The time step was still 0.0001 seconds, and the Simulink simulation time was 10 seconds. The setpoint for this set of simulations is characterized in Equation 3.8, with an angular frequency of approximately 1.5 Hz which was ω=9.4248 rads/s.

$$setpoint(t) = 0.1625 + 0.0775 \cdot sin(9.4248 \cdot t) \tag{3.8}$$

Once again, these results were tested according to the Performance tests prepared in section 3.4.3.

3.4.2.C PID gains and time constants conversion

As mentioned before, there were two groups of controllers found: one using the LabVIEW software and the other using the MATLAB/Simulink environment. In MATLAB/Simulink the PID block requires to use the PID parameters as gains (K_p, K_i and K_d) while the LabVIEW environment uses the PID parameters in gains (K_c) and time constants (T_i and T_d) in minutes. It is necessary to convert these parameters in order to use them all in the required software, and the relationship between all of these parameters can be found in Equations 3.9 - 3.11:

$$K_c = K_p; \tag{3.9}$$

$$T_i = \frac{K_p}{K_i} \cdot \frac{1}{60}\,(minutes); \tag{3.10}$$

$$T_d = \frac{K_d}{K_p} \cdot \frac{1}{60}\,(minutes); \tag{3.11}$$

3.4.3 System's Performance Tests

The controller we are trying to tune is for a post-stroke recovery device that allows the therapist to define arbitrary trajectories with variable amplitude and speed, therefore it must be effective in following a cycling frequency of around 60 RPM (1 Hz) and input amplitude shifts as quickly as possible. As a result, a series of tests were prepared to determine how well the controllers found using the Autotuning Method (Section 3.4.1) and the ones found using the GA optimization (Section 3.4.2) match our requirements.

3.4.3.A Step-response characteristics

The goal of this test was to study the rise time, settling time, overshoot, peak and peak time of each controller. Based on these results, it was possible to find which was the fastest in convergence and with the less overshoot possible. To do this test it was necessary to define the input of the system as a step with 0.2 m of amplitude and run the LabVIEW implemented model (for the Autotuning results) or run the Simulink model (for the GA results) for 4 seconds, time chosen to guarantee that all controllers converge to the setpoint value. The MATLAB environment function "stepinfo(x,t,y)" was used to obtain the desired information, where x was a vector with the step-response data from the system, t was the corresponding time vector and y was the value of the step used in the test, all this information was exported from the models after the simulation. This test was performed for every controller and the results saved in a table

Table 3.2: Chosen parameters for the Chirp Signal block used in the frequency performance test.

Parameter	Value
Initial Time (s)	0.0
Initial Frequency (Hz)	0.1
Target Time (s)	10.0
Frequency at target time (Hz)	4.0

in order to compare each of them. The results of this test can be seen in Section 5.1.1 and 5.2.1, for the controllers from the autotuning wizard and for the controllers discovered with the GA respectively.

3.4.3.B Frequency Test

This test consisted of two parts: the one where it was applied a chirp signal at the system's input and the other where the setpoint had its frequency manually changed. The aim of this experiment was to check how controllers performed when they were subjected to a signal with increasing frequency according to the simulation time (the chirp signal) and a signal whose frequency was increasing sequentially. This test was carried out in the LabVIEW environment for the PIDs found with the Autotuning Wizard, and in the Simulink environment for the GA results. The results for this test can be seen in Sections 5.1.2 (for the LabVIEW results) and 5.2.2 (for the GA results).

Setpoint: Chirp Signal

To perform this test it was necessary to use the "Chirp Signal" block, which generates a sine wave whose frequency increases at a linear rate with time. This sine wave was manipulated as Equation 3.7 to fit in the pedal range with the difference of the frequency that was not a constant but a variable changed according to the chirp signal block algorithm. The input parameters of this block included the initial frequency, the target time and frequency at target time. The values specified for the input of this block are shown in Table 3.2 and Equation 3.12 determines the frequency (in hertz) of the signal at a given time (in seconds) of the Simulink simulation while Equation 3.13 also determines the frequency (in hertz) at a given time (seconds) but for the LabVIEW simulations.

$$f(t) = \frac{f_{target} - f_{initial}}{t_{target} - t_{initial}} \cdot t + f_{initial} \tag{3.12}$$

$$f(t) = \frac{f_{target} - f_{initial}}{2 \cdot t_{target}} \cdot t + f_{initial} \tag{3.13}$$

Setpoint: Sine wave with frequency changed manually

It was observed that the system showed different behaviors when facing continuously changing frequency signals and signals with a constant defined frequency. This test was designed to understand the system performance when having slower changes in frequency. With the chirp signal test performed previously, the frequency is changed each time step which forces the system to adapt constantly without time to converge. With this test, the frequency is increased in phases and kept for a determined period, thereby, it is given enough time for the controller to converge and analyze the system performance when in the presence of frequency stabled signals and having a truthful analysis of the system when in presence of higher frequency signals.

This test required the use of the sine wave block with the frequency parameter connected to a sliding scale, allowing to change the frequency value of the signal every 2 seconds. In this test, the sine wave started with 1 Hz and the frequency was increased 0.1 Hz each 2 seconds until it reaches 2 Hz. The test was 22 seconds long, with 10 different parts for each frequency step.

3.4.3.C Delay and RMSE calculation

The test to detect the delay between the process variable and the setpoint and the Root Mean Square deviation was calculated using the MATLAB environment and employing the functions “finddelay” and “immse”. The former function returns an integer scalar representing the delay between the two input signals, in the case of periodic signals, the delay with the smallest absolute value is returned. The latter function calculates the mean-squared error between the arrays x and y given as input, to obtain the Root-Mean-Square Error (RMSE) it is done the squared root of the mean-squared error.

For this purpose, all controllers in both LabVIEW and Simulink models were run for 4 seconds with a sine wave with 1 Hz. The data from the setpoint and the process variables were exported at the end of the simulation. For the Simulink model, it was only needed to create a workspace variable using the block “to workspace”. To analyze the data from the LabVIEW model, it was needed to export it to a Microsoft Excel file and then read it with MATLAB function “xlsread” to extract the needed information.

The information given by the “finndelay” function is in sample units, therefore we have to convert the results to seconds. The sampling frequency of all signals is the same and it was found to be 1000 Hz, according to Equation 3.14 and knowing that the time step was defined as 0.001 seconds. By definition, the sampling frequency is the number of samples per seconds, which makes it possible to determine the delay time in seconds through Equation 3.15, where T_{delay} is the delay between the two signals in seconds and $N_{samples}$ is the number of samples of delay between the two signals.

$$F_s = \frac{1}{dt} \Leftrightarrow F_s = \frac{1}{0.001} \Leftrightarrow F_s = 1000Hz \tag{3.14}$$

$$T_{delay} = \frac{N_{samples}}{1000} \tag{3.15}$$

Chapter 4

Pedal Trajectory and GUI

One of the most innovative features of this system, as previously mentioned, is the therapist's opportunity to freely design a pedal trajectory based on his experience that will contribute to efficient patient recovery. This section explains how the trajectory block was implemented in LabVIEW, as well as the algorithm used to generate the trajectory from the selected points.

4.1 Trajectory Planning

The purpose of *trajectory planning* is to generate the reference inputs to the motion control system which ensures that the robot executes the planned trajectories. [98] The *path* is called to the location of points in the joint space or operational space, which the robot has to follow to execute the desired motion. Therefore, a path is just a geometric description of motion, while a *trajectory* is the combination of the path and the timing law, which will define velocities and accelerations at each point. [98] In our project, the system must receive the generated trajectory from the points and timing settings defined by the therapist.

In the Graphical User Interface (GUI) to be developed, it is asked from the user (therapist) to select a number of points that will define the path the system will follow. In Figure 4.1, it is shown the IMAQ display in the project's front panel where there is a scheme of the pedal. The two red circles, concentric with the pedal, delimit the pedal's workspace, which is between 8,5cm (represented by the smaller circle) and 24cm (represented by the bigger circle). The user is advised to choose between 6 and 15 points between the circles as this is a reasonable amount of data to create a trajectory. However, this is just a guideline, and the device will consider fewer or more points. In case the user selects any point outside the range, the system will show an error pop-up and will ask for the user to select the points inside the range. The 8 cross points seen in Figure 4.1 are an example of where the points should be placed and how it looks like when selecting all the initial path points.

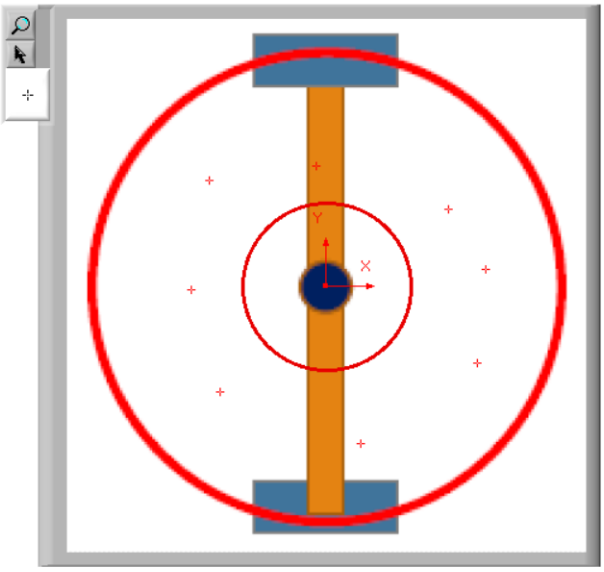


Figure 4.1: LabVIEW IMAQ display with representative image of pedal and its workspace.

Following the selection of path points, the user must enter the simulation time (in seconds) and pedal rotation frequency (in hertz) into the user interface. Afterwards, these two parameters will be used to determine the number of cycles to run as well as the theta step (in radians) of the simulation, data important for the trajectory planning algorithm.

After all the information has been provided, the trajectory planning algorithm will process it and compute the system's trajectory.

4.1.1 Trajectory Planning Algorithm

The trajectory planning algorithm is a series of steps necessary to process the data collected in order to achieve the actual trajectory that will be given to the system. This algorithm is divided into three stages of data manipulation: data processing, vector manipulation, and trajectory interpolation.

Data Processing

At this stage, the user has selected the points in the pedal workspace, as well as the simulation time and frequency. The frequency (f) is first converted to angular velocity (ω) based on the relationship between these two parameters (Equation 4.1). With the angular velocity is then straightforward with Equation 4.2 to obtain the value of the theta step (dθ) having in mind that the time step (dt) is fixed in 0.001 seconds..

$$\omega = 2 \cdot \pi \cdot f \tag{4.1}$$

$$d\theta = \omega \cdot dt \tag{4.2}$$

With the points to define the trajectory already chosen, the next step is to obtain its coordinates, this is accomplished with resource to some built-in VIs in LabVIEW. First, the pedal image is obtained and calibrated using the Vision Assistant tool where it is defined the center of the image and the real-world measurements of the image (in meters). After this, the image is ready to be processed with Region of Interest (ROI) tool where it will take out the contours in the image (the red crosses). These contours will then be carried to an implemented VI which will use this information to extract the coordinates of each red cross and put it in an XY cluster. While the coordinates of the selected points are being extracted, there is a verification process to check if the cross is placed correctly in the pedal workspace. As said previously, these points must be selected within the two red circles representing the minimum and maximum crack arm length, the verification is done by calculating the distance between the point and the circumference center and verifying if it is a value between the smaller and the bigger circumference radius. In case a point is found to be outside the allowed range, an error message is displayed to the user with the caption “Selection out of range” and the simulation stops automatically. At the end of this step, we have all the XY coordinates (in pixels) in a cluster which will be used in the following steps.

Coordinates Manipulation

In this phase, a few manipulations are done to the XY cluster created in the previous step. First, there is a conversion from the units of the coordinates (in pixels) to real-world units (in meters), this step is done applying the “IMAQ Convert Pixel to Real World” VI which uses the previous calibrated image and the array with pixel coordinates to transform these into a Real World coordinates with the desired dimension. Then, the real world coordinates are converted from Cartesian (X,Y) to polar coordinates (L,θ), according to Equations 4.3 and 4.4. At the end of this step, we have an array containing the red cross coordinates in L (meters) and Thetas (rads). This array is then separated in two.

$$L = \sqrt{x^2 + y^2} \;\; (m) \tag{4.3}$$

$$\theta = \begin{cases} \arctan(\frac{y}{x}), & y > 0 \\ \arctan(\frac{y}{x}) + 2\pi, & y < 0 \end{cases} \;\; (rads) \tag{4.4}$$

At the end of the L array, it is added the value of the first element in order to duplicate the first point selected, this will later force the trajectory interpolation to do a complete circle by continuing to interpolate the values up to the first point selected. This procedure also happens for the Theta array, but the angle value added at the end is the value of the first element plus 2 pi to prevent the angle values to have a gap between the first point and the last. Another VI was implemented to analyze the Theta array and get the minimum angle found (which will correspond to the first point selected) and the maximum angle found (corresponding to the last added element) as this information will be necessary for the next steps.

Trajectory Interpolation

It's at this step that an important array is created, the ntheta array which contains all the theta values that we wish to interpolate for the complete trajectory over the simulation time. This VI uses the simulation time (T) defined by the user to create a time array from 0 to T with the defined time steps. Another array is created using the minimum and maximum theta found in the Theta array and with the steps according to the previously calculated dθ. After this, the ratio between the sizes of the time array and the theta array is computed, to know how many theta cycles the time array contains. This ratio will give the "number of cycles" information to the user. With this information, a new theta array is created (called ntheta) which is the same size as the time array and with the initial theta array repeated as necessary to fill in the simulation time (T). The ntheta array must be used in the interpolation block to force the interpolation at the points contained in the array. Therefore, it will be given as input together with the L and the Theta array for the interpolation block.

To proceed with the trajectory interpolation the LabVIEW block "Interpolate 1D VI" was used, which performed one-dimensional interpolation using one of the available interpolation methods and based on the lookup table defined by the X and Y (being the L and Theta vectors in this case). The selected method was the cubic spline interpolation method, which proved to be the one leading to a more fit trajectory to the selected points due to the method of fitting low-degree polynomials to small subsets of data instead of a high-degree polynomial fitting all points at once. In contrast to piecewise linear interpolation, cubic spline interpolation is piecewise smooth, with continuous slope and curvature at the joint points. Furthermore, the spline interpolation prevents Runge's phenomenon, when oscillation occurs between points as had happened with other interpolation methods. [99] [100]

The interpolation block returns as output two arrays: one with the thetas at each time step of the simulation and the other with the corresponding L values. The representation of the interpolated trajectory can be seen in Figure 4.2, where the selected points present in Figure 4.1 can also be seen in blue.

At the end of this step, we have the complete interpolated trajectory that will be performed across the simulation time (T) according to the frequency (f) defined by the user. The L interpolated array is now ready to be transferred to the system's block to perform the simulation.

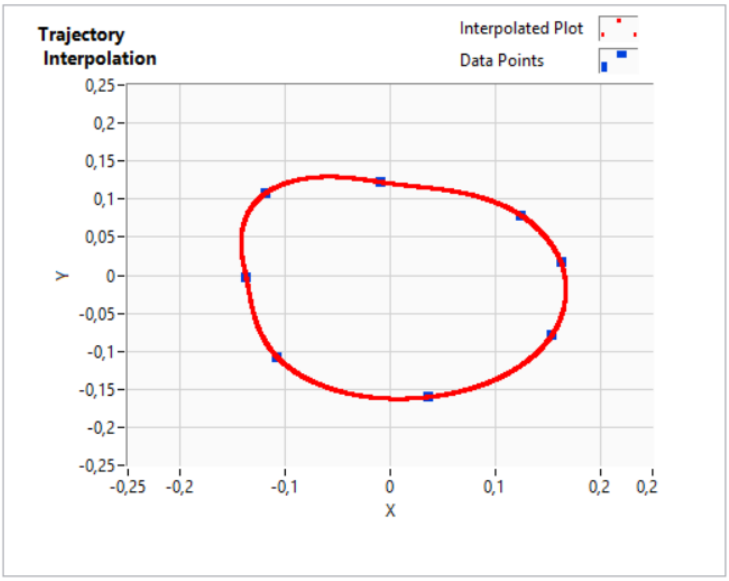


Figure 4.2: Project's front panel display with the interpolated trajectory resulting from the selected points present in Figure 4.1

4.1.2 Predefined Trajectories

The previous section described how the trajectory is created from a set of selected points by the user, nonetheless, there is still the possibility to have predefined trajectories that can be selected for the system to follow. Three predefined trajectories were implemented: a circular shape (Figure 4.3.a), a butterfly similar shape (Figure 4.3.b), and the foot gait pattern (Figure 4.3.c). In case the user chooses to use an already defined trajectory, it is only needed to select which one is the more appropriate and set the simulation time and the system will then perform the trajectory. The foot gait pattern trajectory was created based on the pattern described in Figure 4.4 found in [18], where it is emphasized the key events of the regular foot trajectory within the gait cycle. There were selected 20 points to define the trajectory which had to be placed within the pedal range limits, some data points needed to be relocated from the ideal location due to this limitation and therefore assigning a slightly different shape when comparing with the ideal trajectory (such as the one shown in Figure 4.4).

These predefined trajectories were initially created as the manual method described previously, then the interpolated L array is saved and given to the system when each predefined trajectory is selected at the front panel.

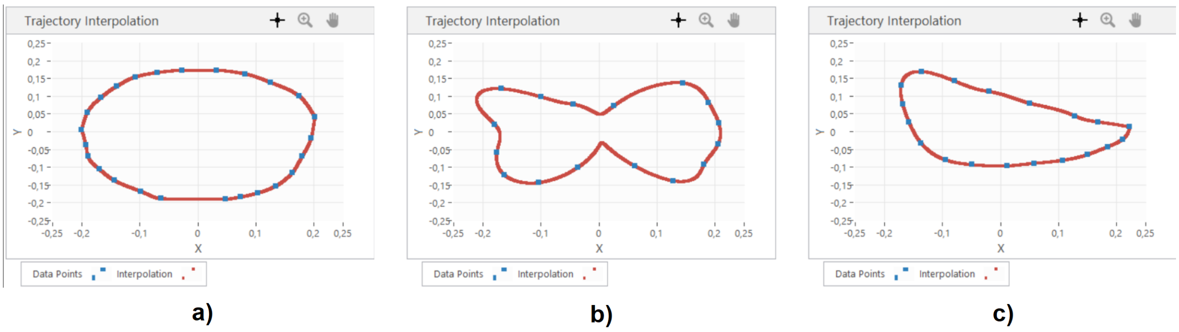


Figure 4.3: XY graph representing the three pre-defined trajectories implemented. a) Circular trajectory, b) Butterfly shape, c) Foot gait pattern.

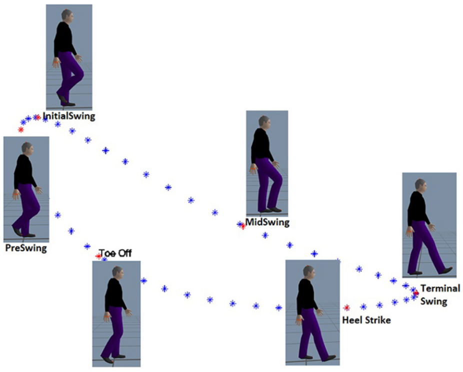


Figure 4.4: Gait pattern trajectory showing the position of the ankle in the saggital plane and representing the key events in the regular gait pattern. Adapted from [18].

4.1.3 PID Selection method

The results of Chapter 5 did not conclude with a preferred PID controller, since all controllers had to be further tested with an actual trajectory rather than simple sinusoidal waves.

Now, with the trajectory algorithm implemented and a few trajectories created, we are in the position to continue to test the previous controllers. By the end of Chapter 5, it was concluded that the group of controllers with the best general performance were the ones found with the GA optimization and tuned with a step signal, these are the ones found with the Simulink model and expressed in gains and not time constants as the parameters used in LabVIEW. First, the gain parameters (Kp, Kd, Ki) were converted to time parameters (Kc, Ti, Td), according to equations present in Section 3.4.2.C, and applied in the LabVIEW software, however, it was noticed that there was a difference when comparing the controllers' response in the LabVIEW model to the Simulink model. This performance difference was due to the windup functionality already integrated into the PID block from LabVIEW, and that was by default disabled in Simulink. To correct this error, the built-in LabVIEW PID block was replaced by a set of blocks that represents the transfer function of the PID controller, which can be seen in Figure 4.5, and had as input the gains Kp, Kd and Ki. With this alternative solution, the controllers had the same response as when they were tested in Simulink, and it was no longer needed to convert the gain parameters.

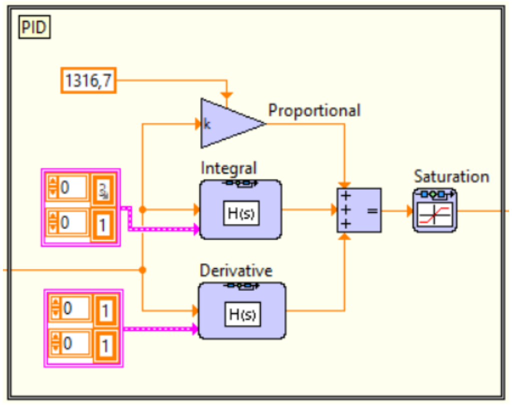


Figure 4.5: New implementation for the PID to correct the windup problem.

To test the controllers it was chosen to have the foot gait pattern trajectory as the setpoint, one of the predefined implemented trajectories, considering that it is the simplest standard exercise the system must be able to follow when training gait recovery. To compare the controllers' performance it was calculated the delay and RMSE values according to the methodology described in Section 3.4.3.C, where the simulation was held for 4 seconds, enough time for the system to perform 2 cycles of the trajectory. Section 6.1 presents and discusses these results.

4.1.4 Trajectory Tracking Evaluation Parameters

To assess the quality of the system performance when using trajectories, the RMSE and the delay between the trajectory and the system response were calculated using the same method as described in Section 3.4.3.C. Additionally, the cross-correlation between the two signals was computed using LabVIEW box "TSA Cross-Correlation", receiving the Setpoint and the Process Variable as input, calculates a biased weighting cross-correlation and returns the cross-correlation vector and the correlogram, which is a graph of the cross-correlation values against the lag. Section 6.2 shows the outcomes of this procedure.

4.2 Graphical User Interface

To improve the user experience throughout the sessions, a Graphical User Interface (GUI) was created with a set of features to allow the user to correctly choose which trajectory planning method they wish to use (manually selected trajectory or use of a pre-defined trajectory) as well as providing necessary feedback information to the user, such as the system tracking trajectory in real time or visualizing the pedal movement.

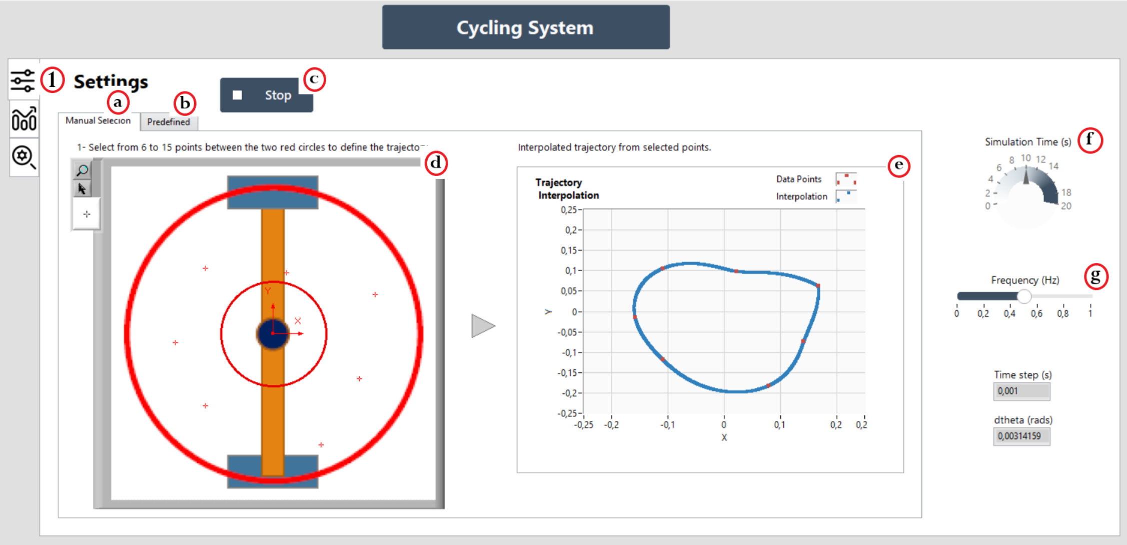
4.2.1 GUI description

The user interface in this project consists of a panel with three tabs that the user can choose from depending on the simulation phase. In each tab, the user can either provide information for the device to conduct the simulation or obtain the simulation's results.

First tab - Settings

The first tab is the Settings tab (Figure 4.6, element 1) which is the first step of the simulation, where the user will provide all the information needed for the simulation, such as the trajectory data points (to interpolate the pedal trajectory), the desired simulation time and pedal frequency. Inside this tab, there is a box containing two tabs: the tab to select the manual trajectory definition (Figure 4.6, element a) and the tab to select the pre-defined trajectories (Figure 4.6, element b). The user should decide which method better fit his necessities. In the manual trajectory definition (Figure 4.6), on the left side there is a graphical box containing the pedal image and its limits (Figure 4.6, element d) and is at this box where the user will define the data points later used for the trajectory interpolation. The user is requested to select from 6 to 15 points at this box, which is a chosen number of elements that are enough to create a trajectory, however in case the user needs to select more points the system will accept it. Then, the user must select the simulation time and the frequency desired (Figure 4.6, elements f and g, respectively). If the user decides the use the pre-defined trajectories (Figure 4.7), he must select the corresponding tab and choose between the 3 available predefined trajectories (Figure 4.7, element h) and select the desired simulation time (element f of Figure 4.7). The frequency of all pre-defined trajectories is set to 0.5 Hz, and for this reason, the user is not required to select any frequency.

After all the settings defined, the user can press the built-in Run button of the LabVIEW software and check the interpolated trajectory in the right box present in both manual (Figure 4.6, element e) or predefined trajectories tab (Figure 4.6, element i), this box allows the user to check if the interpolation has the desired characteristics and to validate the trajectory. If the trajectory is defective in some way, the user should press the Stop button (Figure 4.6 or 4.7, element c) and then adjust the data points selected (by dragging the red crosses) or defining a new set of data points. Otherwise, the user can then move to the second tab (Figure 4.8).



1- Select from 6 to 15 points between the two red circles to define the trajecto

Interpolated trajectory from selected points.

Time step (s) 0,001

dtheta (rads) 0,00314159

Figure 4.6: Project's Graphical User Interface, first tab with the settings for the simulations and the Manual trajectory tab.

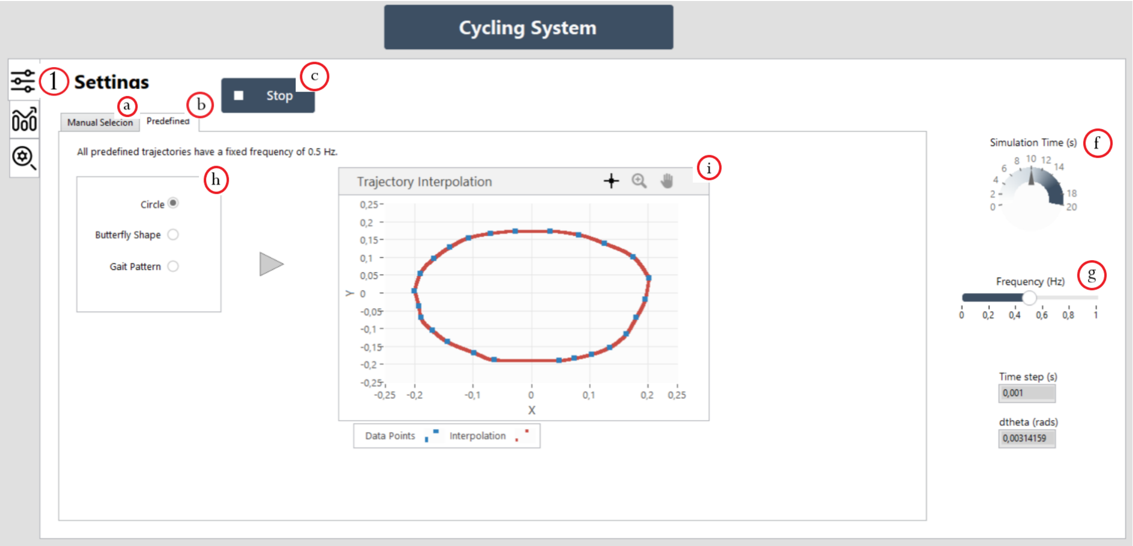
Second tab - Results

The second tab (Figure 4.8) has the Results in real-time. At the upper part of the tab, there is the Stop button (Figure 4.8, element j) and 4 numeric boxes representing the pedal velocity in RPM (Figure 4.8, element k), the number of total cycles the pedal will perform within the simulation time (Figure 4.8, element l), the total distance traveled by the patient (in meters) at the end of the simulation (Figure 4.8, element m) and the current simulation time (Figure 4.8, element n).

The pedal velocity (RPM) displayed in the GUI is calculated through the frequency defined by the user in the first tab and using the following Equation 4.5.

$$RPM = f \cdot 60 \tag{4.5}$$

The number of total cycles is determined during the trajectory definition algorithm, where it is done a calculation of the total time steps in the simulation and the total angle steps are needed to perform the defined trajectory. As both values are not in the same dimension, there is the need to create an array of angle steps with the same size as the time steps for the full simulation time, and for that reason, the angle steps from the trajectory are replicated creating the number of cycles performed within the simulation. To calculate the distance traveled by the patient throughout the simulation, a sub-VI receiving as input the array with the interpolated x coordinates and another with the y coordinates was created. It was then calculated the distance between two consecutive points using Equations 4.6 - 4.8 and, in the end, a summation of all the distances calculated (Di) to obtain the total distance L (Equation 4.9, where i is the index from 1 to N number of total points in both x and y arrays).



All predefined trajectories have a fixed frequency of 0.5 Hz.

Circle ●
Butterfly Shape ○
Gait Pattern ○

Time step (s) 0,001

dtheta (rads) 0,00314159

Figure 4.7: Project's Graphical User Interface, first tab with the Settings for the simulations and the predefined trajectories tab.

$$dx = (x_{i+1} - x_i)^2 \tag{4.6}$$

$$dy = (y_{i+1} - y_i)^2 \tag{4.7}$$

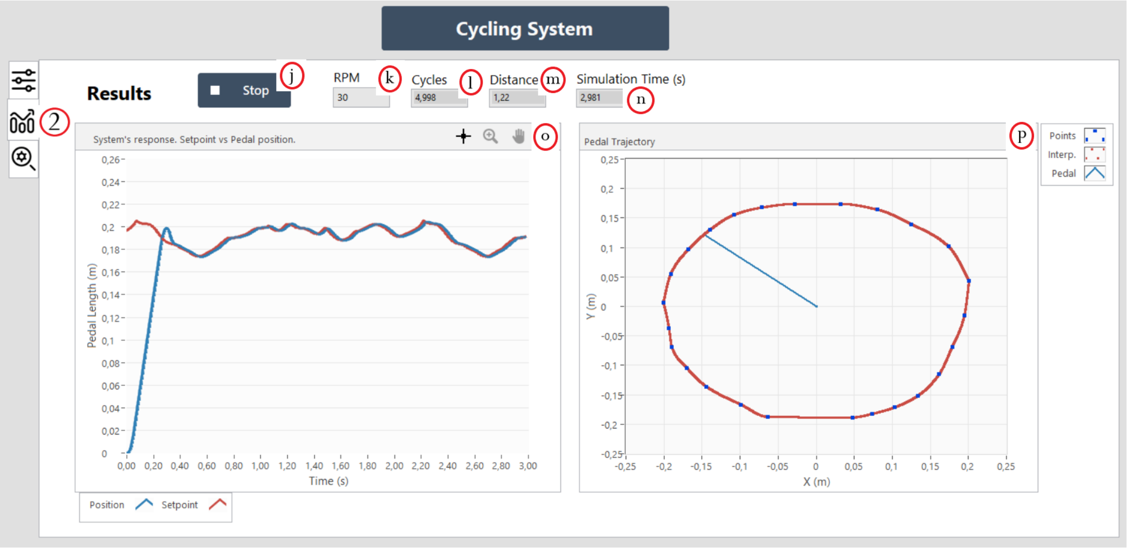
$$Di = \sqrt{dx + dy} \tag{4.8}$$

$$L = \sum_{i=1}^{N} D_i \tag{4.9}$$

At the bottom, there are two graphical displays, the one on the left shows the pedal length versus time (Figure 4.8, element o) where can be seen the desired pedal length (calculated from the trajectory interpolation) in red and the system's response in blue. On the right side, there is an XY graph (Figure 4.8, element p) displaying the interpolated trajectory in Cartesian coordinates in red, the data points selected by the user at the settings part in dark blue, and in light blue, it is represented a vector which moves throughout the simulation portraying the pedal movement. Both graphics are updated in each simulation time step, giving the user real-time tracking results. All of this information allows the user to track the system's response, to compare and evaluate it against the desired settings, and to obtain useful information to the therapy implemented.

Third tab - Verification

As this is an ongoing project that will be developed further, the third tab "Verification Parameters" (Figure



Cycling System

Results

RPM: 30

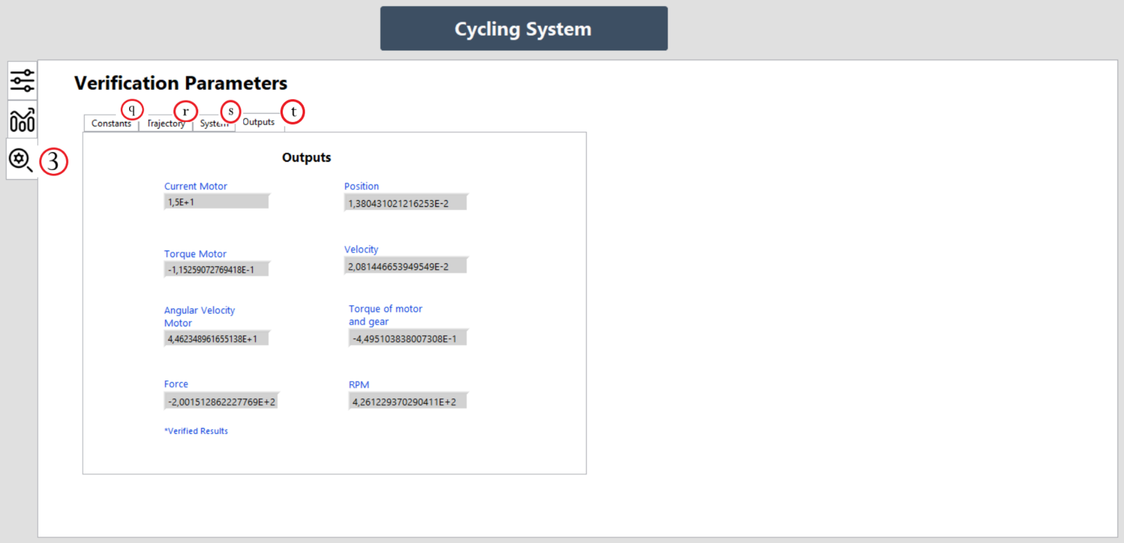
Cycles: 4,998

Distance: 1,22

Simulation Time (s): 2,981

Figure 4.8: Project's Graphical User Interface, second tab with the simulation Results.

4.9) is listed for the engineers working with the system since it returns details unrelated to the functional part of the system (unlike the results tab where it is given information about the implemented therapy and it is especially useful for the practitioner and patient). At this tab there are 4 tabs for different parts of the system and simulation phases containing: inputs given to the system such as mechanical characteristics as motor and pedal constants (present in Figure 4.9, element q); calculated parameters in the trajectory interpolation algorithm, several position arrays in cartesian or polar coordinates (Figure 4.9, element r); system variables calculated during the simulation, e.g. loads, pedal inertia, currents and other variables (Figure 4.9, element s); system's blocks outputs as for example pedal position (Figure 4.9, element t). The easy access to these variables is important for the continuing developing of this software as in case there is a fault in the debugging process, the worker is able to promptly check the variable values to identify the error and to correct it.



Outputs

Current Motor: 1,5E+1

Position: 1,380431021216253E-2

Torque Motor: -1,15259072769418E-1

Velocity: 2,081446653949549E-2

Angular Velocity Motor: 4,462348961655138E+1

Torque of motor and gear: -4,495103838007308E-1

Force: -2,001512862227769E+2

RPM: 4,261229370290411E+2

*Verified Results

Figure 4.9: Project's Graphical User Interface, third tab with the verification parameters.

Chapter 5

PID Control Study Results

5.1 LabVIEW Autotuning results

The parameters in Table 5.1 were discovered after several trials with the LabVIEW Autotuning Wizard to find a suitable PID controller. The parameters found were only the Proportional Gain and the Integral Time, indicating that the controllers discovered were PI controllers.

Table 5.1: Best PID parameters found with Autotuning Wizard.

Controller	Setpoint	PID Parameters	Value
1	Step	Proportional Gain (Kc) Integral Time (Ti, min) Derivative Time (Td, min)	6.733267$\cdot 10^{11}$ 4.704400$\cdot 10^{-2}$ 0
2	Sine Wave	Proportional Gain (Kc) Integral Time (Ti, min) Derivative Time (Td, min)	2.312258$\cdot 10^{9}$ 2.761470$\cdot 10^{-2}$ 0

The step response for controller 1 is shown in Figure 5.1. This controller has a good rising time and converges to the setpoint value in 0.5 seconds; however, there is some overshoot and the signal appears to be slightly oscillatory after convergence.

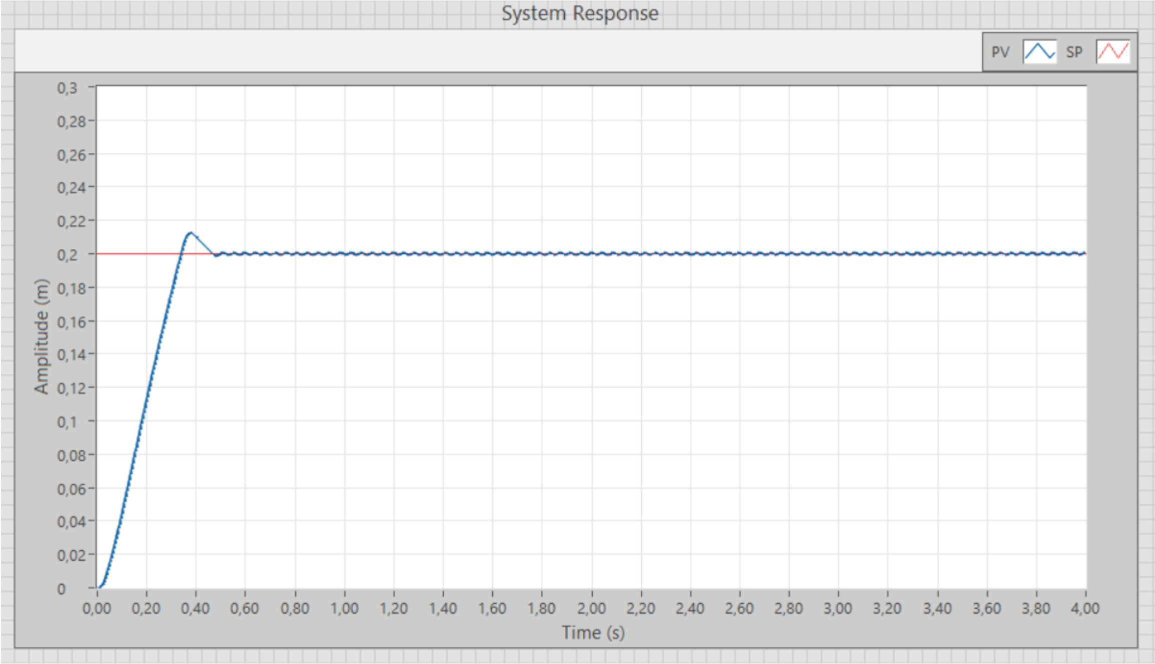


Figure 5.1: System response of controller 1 to a step of 0.2m. It can be seen the step signal (SP) in red and, in blue, the system response (PV)

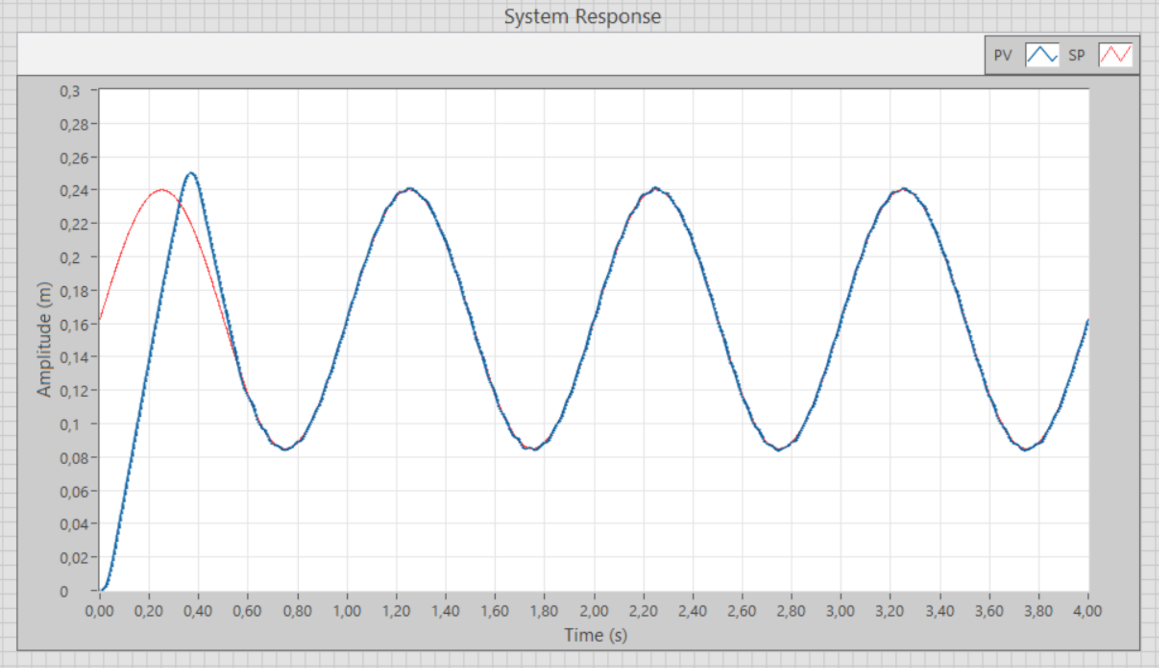


Figure 5.2: System response (PV, blue line) of controller 2 to the sine wave (SP, red line) with 1 Hz used as a setpoint for the Autotuning Wizard.

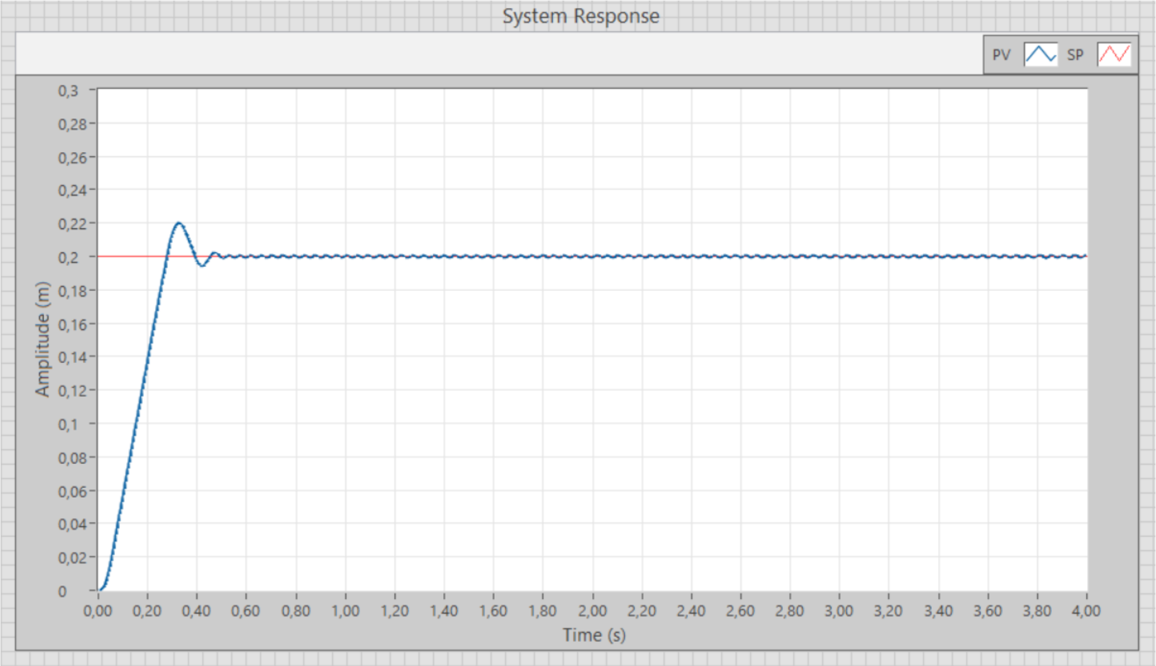


Figure 5.3: System Response of controller 2 for a step of 0.2m, where the Y axis represent the signal amplitude (in meters) and the X axis the time of simulation (in seconds). The setpoint signal (SP) is represented by the red line and the system response (PV) in a blue line

The results for controller 2, which was tuned for a sine wave, are less satisfying in terms of rising time, with convergence to the setpoint taking 0.6 seconds (Figure 5.2). When using this controller with a step signal (Figure 5.3) the device takes about 0.5 seconds to reach the setpoint. After convergence, the signal appears to be mildly oscillatory in both responses (Figure 5.2 and Figure 5.3), with some ringing behavior as shown by controller 1.

These findings are analyzed in depth in Section 5.1.1, according to the step-response characteristics.

5.1.1 Step-response characteristics

This test was performed according to the methodology described in Section 3.4.3. The results obtained for the step response characteristics of controllers 1 and 2 can be found in Table 5.2, with the best results highlighted in blue and the worst in red.

Let us recall that having a controller that can quickly follow changes in the setpoint signal is crucial for our system; in this case, and based on the results shown in Table 5.2, controller 2 is the best match between the two. This controller has a faster rise time and a shorter settling time. The overshoot of 9.35 $\%$, hitting a peak of 0.2187 m, is one of this controller's drawbacks. However, after settling, both controllers display oscillatory behavior, which is undesirable.

Table 5.2: Step Response characteristics of controllers found with the Autotuning wizard from LabVIEW.

Controller	1	2
Rise Time (s)	0.2199	0.2020
Settling Time (s)	0.4430	0.4165
Overshoot (%)	7.4000	9.3500
Peak (m)	0.2148	0.2187
Peak Time (s)	0.3500	0.3310

5.1.2 Frequency Test

The results of the Frequency performance tests are reported in the subsections below. For controllers 1 and 2, the Chirp Signal test and the manually defined frequency test were performed according to the instructions in Section 3.4.3.

Setpoint: Chirp Signal

Equation 5.1 allows us to determine at which frequency the device no longer can reach the setpoint signal, based on Equation 3.13 and the parameters specified for the Chirp signal listed in Table 3.2. The chirp signal test for controller 1 is shown in Figure 5.4, where it can be seen that both signals (the setpoint and the process variable) are no longer superimposed as of second 4.9 of simulation.

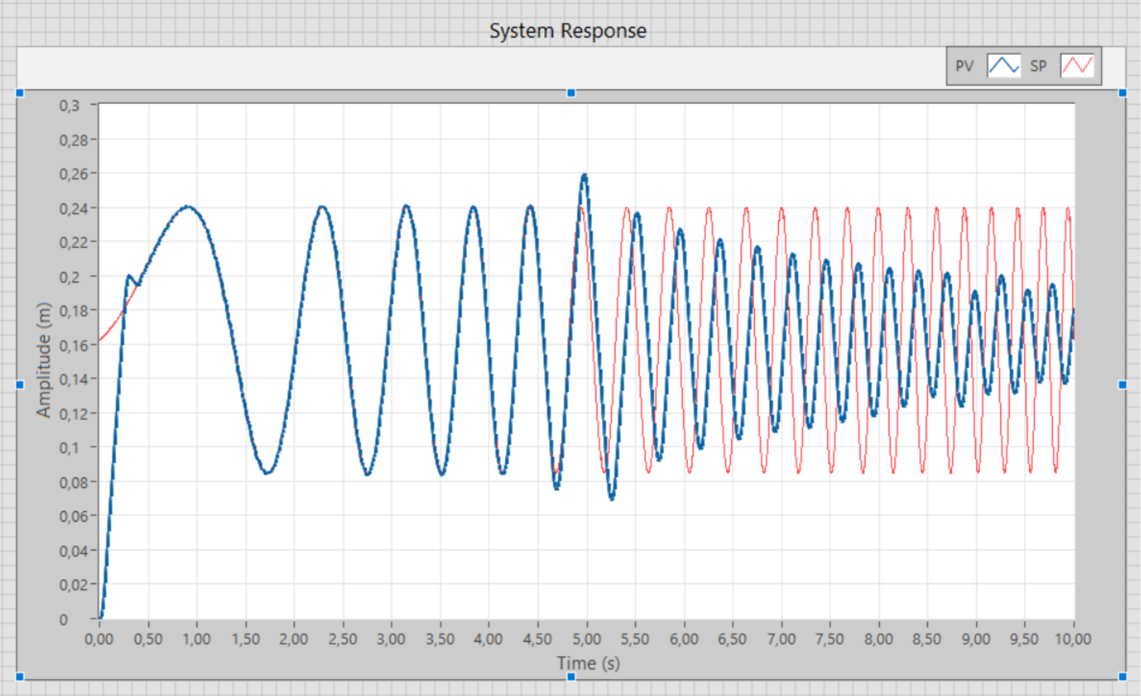


Figure 5.4: Chirp Signal Test for controller 1

$$f(t) = \frac{4 - 0.1}{2 \cdot 10} \cdot t + 0.1 \tag{5.1}$$

Table 5.3: Controller 1 and 2 performance in the Chirp signal test.

Controller	Time Limit (s)	Frequency Limit (Hz)
1	4.90	1.06
2	4.90	1.06

The test for controller 2 was conducted in the same manner as for controller 1, and due to the high similarity of results, it is shown in Appendix A.2. Table 5.3 contains both the results for the maximum time controllers could obey the chirp signal and the corresponding frequency. In this test, both controllers performed similarly, demonstrating that they can follow a signal with increasing frequency up to 1.06 Hz, which is within the frequency range the device is required to follow.

Setpoint: Sine wave with frequency changed manually

Images 5.5 and 5.6, display the results obtained for this test with controllers 1 and 2, respectively. Ten vertical red lines can be seen in both Images, indicating shifts in the setpoint frequency every 2 seconds of simulation; the frequency for each region is also shown in the figure.

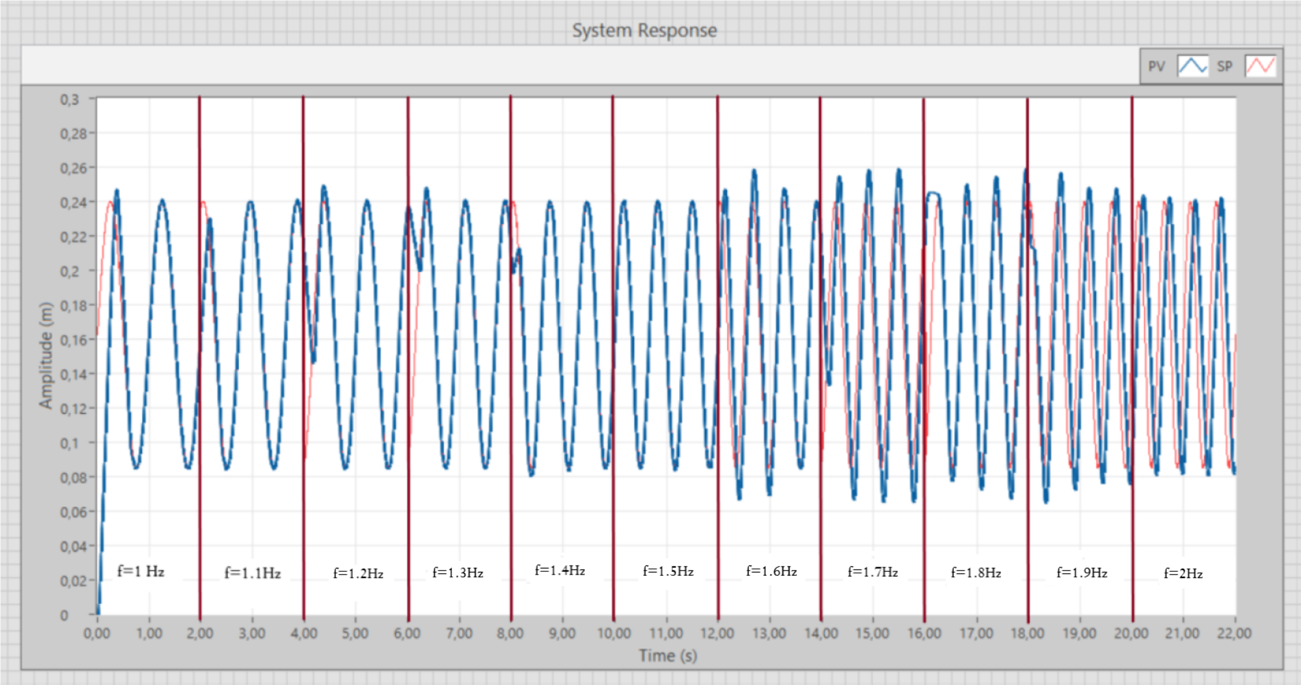


Figure 5.5: Manual Chirp Signal Test for controller 1

When looking at Figure 5.5, it is noticeable that, for frequencies in the range of 1.0 to 1.4 Hz, the model takes a few milliseconds to converge to the SP signal, about 0.40 seconds, corresponding to the settling time found in the step signal test. The signal converges completely after these first milliseconds, maintaining the satisfactory behavior until the next frequency is set at the SP signal. Due to the similarity of both the System's and SP signal at the time of the frequency shift, the signal converges much quicker at 1.5 Hz; although, when adjusting the SP frequency in the other parts of the test, there is a spike in the SP signal, which takes the system some time to settle again. Between 1.6 and 2 Hz, the device appears to have difficulty following the test signal. First, due to a lack of time to settle, it is unable to fully converge to the SP signal, resulting in major overshoot at the sine wave peaks; then, as the frequency rises, it takes longer to settle and starts to have a phase displacement; and finally, as the frequency continues to increase, it takes longer to settle and begins to have a phase displacement. At the time the SP signal reaches the 2 Hz frequency, the system response (PV) is not capable of converging within 2 seconds.

In Figure 5.6, the system response of controller 2 to this test is shown, which is significantly similar to the response of controller 1.

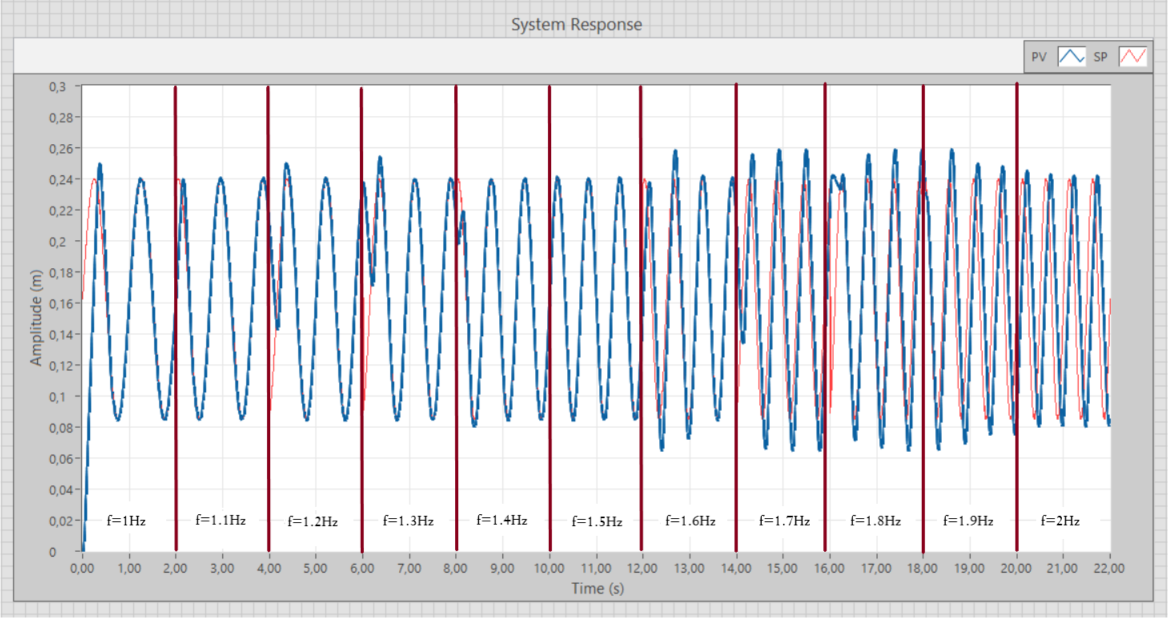


Figure 5.6: Manual Chirp Signal Test for controller 2

In summary, these findings show that both controllers can obey signals with a frequency of up to 1.5 Hz, with enough time to converge and no phase displacement or overshoot. After this frequency, both controllers show difficulties in converging with the SP signal, presenting overshoot and phase displacement.

5.1.3 Delay and RMSE results

The delay test and RMSE calculation were performed as directed in Section 3.4.3.C, and the results for the two controllers are presented in Table 5.4. Based on these findings, it can be concluded that controller 2 has a better match to the sine wave, as shown by a lower delay value and a lower root mean

Table 5.4: Delay and RMSE values found for controllers 1 and 2.

Controller	1	2
Delay (samples)	31	30
Delay (seconds)	$3.1{\cdot}10^{-2}$	$3.00{\cdot}10^{-2}$
RMSE	$3.52{\cdot}10^{-2}$	$3.50{\cdot}10^{-2}$

squared error. Despite the fact that it has better performance in this test, the difference between this controller and the first is minor. The time delays of both controllers are in the range of 30 milliseconds. Assuming a therapy session at 1 Hz, a delay of 30 ms would represent a trajectory shift, in which the patient would place his foot with an inaccuracy of 10 degrees. For this reason, the time delay of both controllers is considered to be inadequate.

5.2 Genetic Algorithm results

According to each fitness function defined in Section 3.4.2.B, the GA found several controllers; all of the controllers found, as well as their characteristics, are listed in Table 5.5.

Table 5.5: Best PID parameters found with the Genetic Algorithm.

Setpoint	Controller	Minimization Function	Fitness Value	Proportional Gain (Kp)	Integral Gain (Ki)	Derivative Gain (Kd)
Step	3	ITSE	$2.6741 \cdot 10^{-4}$	$1.6893 \cdot 10^{3}$	5.0293	4.5115
	4	IAE	$2.6741 \cdot 10^{-4}$	$1.2088 \cdot 10^{3}$	10.7923	19.7306
	5	ISE	$3.9610 \cdot 10^{-3}$	$1.3167 \cdot 10^{3}$	38.6458	44.7140
	6	ITAE	$2.5430 \cdot 10^{-3}$	$1.1880 \cdot 10^{3}$	20.9267	38.8858
	7	MSE	$3.9573 \cdot 10^{-6}$	$1.4609 \cdot 10^{3}$	18.4192	6.7847
Sine Wave	8	ITSE	$7.8753 \cdot 10^{-4}$	$4.3890 \cdot 10^{3}$	16.1422	190.4731
	9	IAE	$5.4530 \cdot 10^{-2}$	$5.6924 \cdot 10^{3}$	13.7644	198.9054
	10	ISE	$5.6610 \cdot 10^{-3}$	$2.3484 \cdot 10^{3}$	3.8705	167.4149
	11	ITAE	$2.7150 \cdot 10^{-2}$	$6.5481 \cdot 10^{3}$	22.7569	14.8852
	12	MSE	$5.6619 \cdot 10^{-7}$	$2.2469 \cdot 10^{3}$	32.5615	191.7545

The GA optimization in MATLAB was used to find controllers 3 to 12 (enumeration according to Table 5.5) with two separate setpoints. Controllers 3–7 were optimized with a step signal of 0.2 m, while controllers 8–12 were optimized with a sine wave according to Equation 3.8. Closer inspection of Table 5.5 shows the fitness values of all controllers discovered with the sine wave are greater than those discovered with the step, indicating a lower fitness quality to the setpoint. Comparing the PID gains from both groups of controllers, the proportional and integral gains are all of the same magnitudes; however, the derivative gains discovered for the sine wave setpoint were generally larger, demonstrating a greater role of the PID's derivative component.

Appendix A.3 shows the step response graphs for all controllers. These findings are analyzed in detail in Section 5.2.1, based on the step-response characteristics.

5.2.1 Step-response characteristics

Table 5.6 presents the results of the GA optimization for the step response characteristics of all controllers discovered.

All of the controllers have the same rise time of 0.9161 seconds, with controller 6 settling the fastest

Table 5.6: Step response characteristics of controllers found with the GA algorithm. In blue are marked the best results found for each step characteristics and in red are the worst results.

Controller	3	4	5	6	7	8	9	10	11	12
Rise Time (s)	0.1961	0.1961	0.1961	0.1961	0.1961	0.1961	0.1961	0.1961	0.1961	0.1961
Settling Time (s)	0.3308	0.3026	0.3176	0.3009	0.3241	0.7768	2.3808	0.3951	NaN	0.3926
Overshoot (%)	4.7836	2.0762	3.1477	2.0296	3.8596	8.4860	9.2486	6.0658	9.8118	6.0261
Peak (m)	0.2096	0.2042	0.2063	0.2041	0.2077	0.2170	0.2185	0.2121	0.2196	0.2121
Peak Time (s)	0.3050	0.2980	0.3010	0.2980	0.3030	0.3140	0.3160	0.3080	0.3170	0.3080

and controller 11 never settling. It was then checked that controller 11 could not converge with the step signal after longer simulation times. Controller 6 had the least amount of overshoot, while controller 11 had the most. Controller 6 was the best controller in terms of reached peak and peak time, while controller 11 was the worst once more. It's interesting that the controllers discovered with ITAE as the minimization function produced both better and worse outcomes. Even though controller 11 had unsatisfactory results with the step response characteristics, when using it to follow a sinusoidal wave this controller proved to be equally effective as the other controllers, implying that we can continue to take it into consideration in the remaining performance tests.

5.2.2 Frequency Test

The results of the Frequency test are presented in the subsections that follow. For controllers 3 to 12, both the Chirp signal test and the manually defined frequency test were performed in the Simulink environment. These experiments were conducted out according to the procedures outlined in Section 3.4.3.

Setpoint: Chirp Signal

Considering the large number of controllers tested and the similarity of the results, only the graph for the first controller tested (controller 3) is shown in Figure 5.7, with the rest of the results being found in Appendix A.4. Equation 5.2 was used to determine the limit frequencies, which was derived from Equation 3.12 and the data from Table 3.2.

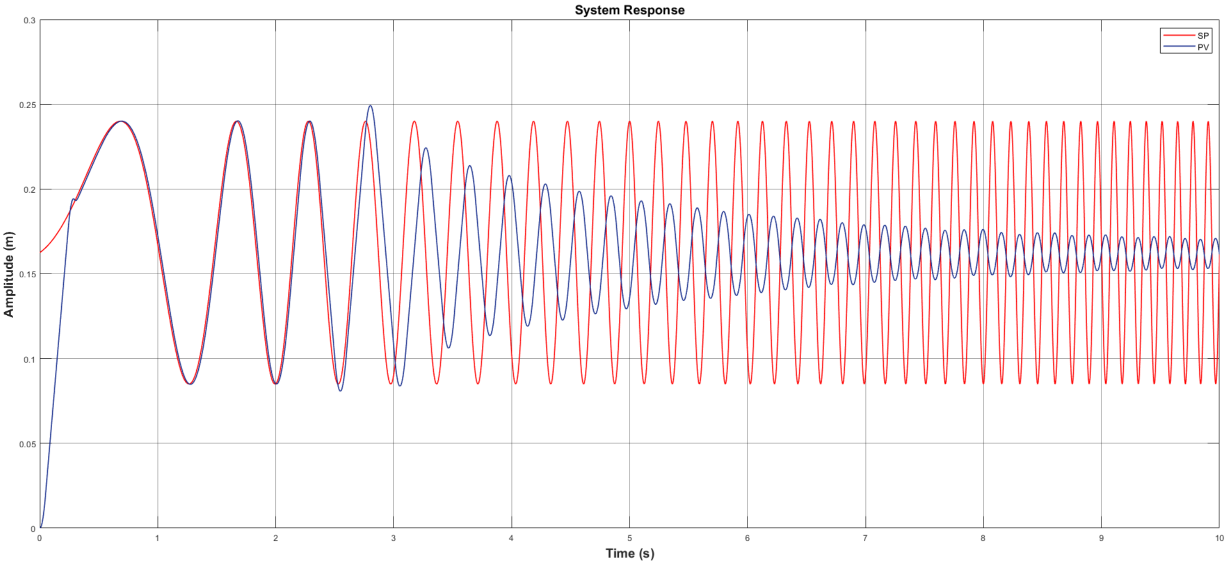


Figure 5.7: Chirp Signal Test for controller 3.

$$f(t) = \frac{4 - 0.1}{10} \cdot t + 0.1 \tag{5.2}$$

Table 5.7: Results for the Chirp signal test for each controller found with the GA.

Controller	Limit Time (s)	Frequency Limit (Hz)
3	2.52	1.08
4	2.78	1.18
5	2.75	1.17
6	2.75	1.17
7	2.75	1.17
8	2.50	1.08
9	2.50	1.07
10	2.51	1.07
11	2.40	1.03
12	2.51	1.07

The results are summarized in Table 5.7, where the limit time is the time when the system no longer aligns with the setpoint and the frequency limit is the frequency determined using Equation 5.2.

Table 5.7 shows that all controllers had similar results for the maximum time they could follow the SP signal, implying that all controllers have similar maximum frequency limits. In general, all of the controllers in this section were capable of tracking sine wave signals with a frequency of about 1 Hz. Controllers 4–7 performed slightly better than the others, being able to follow signals up to 1.18 Hz while the others could only follow signals at lower frequencies of about 1.07 Hz. The ability of controller 11 to imitate the SP signal was the worst, beginning to be out of phase at 1.03 Hz.

Observing the results of these controllers (shown in Appendix A.4), it is important to note that, although their limit frequencies are quite close, their answer in the first few seconds of simulation differs significantly. Controllers 3 to 7 were slightly faster at converging to the chirp signal, but they had a small phase shift during the period they were still able to follow it. When these controllers reached their maximum, they displayed some overshoot at first, followed by a progressive phase displacement.

Surprisingly, the second group of controllers (tuned with a sine wave) behaved differently in the first few seconds of simulation, taking longer to converge to the SP signal and presenting ringing. Controller 8 took 0.5 seconds to converge and ringing, while controllers 9 and 11 experienced this behavior for around 1.5 seconds. Controllers 10 and 12 settled promptly and had residual ringing. In some way, this study is quite revealing, demonstrating that when following low-frequency signals, the second group of controllers performs significantly worse than the first, even considering that they were tuned with a sine wave as a setpoint.

Setpoint: Sine wave with frequency changed manually

The findings for this test are presented in Appendix A.5 due to the large number of controllers. When the graphs with the test results were analyzed, it was noticed that all controllers had a satisfactory settling time between 1 Hz and 1.5 Hz, taking around 0.5 seconds to reach the SP signal.

Controller 3 effectively followed the sine wave up to a frequency of 1.8 Hz, after which the device

Table 5.8: Delay and RMSE results for all controllers found with GA optimization

Controller	3	4	5	6	7	8	9	10	11	12
Delay (samples)	37	42	40	42	39	28	27	32	27	32
Delay (seconds)	$3.7\cdot10^{-2}$	$4.2\cdot10^{-2}$	$4.0\cdot10^{-2}$	$4.2\cdot10^{-2}$	$3.9\cdot10^{-2}$	$2.8\cdot10^{-2}$	$2.7\cdot10^{-2}$	$3.2\cdot10^{-2}$	$2.7\cdot10^{-2}$	$3.2\cdot10^{-2}$
RMSE	$3.40\cdot10^{-2}$	$3.43\cdot10^{-2}$	$3.42\cdot10^{-2}$	$3.43\cdot10^{-2}$	$3.41\cdot10^{-2}$	$3.39\cdot10^{-2}$	$3.39\cdot10^{-2}$	$3.38\cdot10^{-2}$	$3.41\cdot10^{-2}$	$3.38\cdot10^{-2}$

needed more time to settle (about 1 second) and then presented a phase displacement. The behavior of controllers 4-7 was similar, but with a smaller phase displacement after the 1.8 Hz signal. A closer look at the responses of controllers 8-12 reveals a different behavior. Controllers 8 and 10 had some difficulty settling to a 1.6 Hz wave, taking about 1 second to converge, then at 1.7 Hz it took 1.75 seconds to converge, resulting in a large overshoot at the peaks, and at higher frequencies the system was no longer able to settle, resulting in a phase displacement and peaks overshoot. Controller 9 also had difficulty settling with the 1.6 Hz wave, requiring approximately 1 second to converge and 2 seconds at 1.7 Hz, during which it displayed the same behavior as the others, showing phase displacement and overshoot. Controller 11 required 1 second to settle at the 1.6 Hz sine wave and then presented phase displacement and overshoot at peaks while failing to converge to the SP signal. Controller 12 had some difficulty at 1.7 Hz, needing 1 second to settle, but it was no longer able to converge, exhibiting phase displacement and overshoot at the wave peaks.

5.2.3 Delay and RMSE results

Table 5.8 contains the results of the delay and RMSE calculations for all GA optimized PIDs. When comparing the delay results, there is a significant difference between the controllers tuned with the step (controllers 3-7) and the ones tuned with the sine wave (controllers 8-12), with the former having higher delays ranging from 37 to 42 milliseconds from the setpoint and the latter having delays ranging from 27 to 32 milliseconds. Even though the second group of controllers outperforms the first in terms of delay analysis, it is essential to note that even a 20-millisecond delay with the setpoint is significant.

According to the RMSE results, the second group of controllers (controllers 8-12) has better performance than the first group, which can be explained by the fact that these controllers were configured to effectively follow a sine wave and hence are more capable of doing so and with better performance results in experiments using sine waves as the setpoint.

5.3 Controllers performance discussion and conclusions

Following extensive testing of all 12 controllers discovered, it is necessary to discuss the findings and highlight some conclusions. Beginning with the step-response test and providing a summary of all controllers performance, the results are presented in Tables 5.2 and 5.6. The best controller using the first methodology, according to data from Table 5.2, had a rise time of 0.2020 seconds, took about 0.4 seconds to settle and showed an overshoot of 9.35%. Whilst, according to results from Table 5.6, the best controller took 0.1961 seconds of rise time, around 0.3 seconds to settle and overshoot of 2.0296%. These findings suggest that controllers discovered through GA optimization outperformed those discovered through the Autotuning wizard in LabVIEW software. These controllers were found to be better at reaching the setpoint signal, with less overshoot and ringing. When comparing the controllers that were tuned with a step signal (controllers 1, 3-7) against the ones tuned with a sine wave (controllers 2, 8-12), for this test, the former group had better results showing a lower settling time and reduced overshoot. From all controllers, the one presenting better results with a step as a setpoint is controller 6 found with the GA optimization using a step signal as the setpoint and minimizing the ITAE value.

The frequency test was divided into two parts: a chirp signal as the setpoint, where it was determined how far these controllers could follow the SP at a constantly changing frequency; and the second test, which had the goal of evaluating all controllers performance with constant frequency waves at various frequency values, where it was determined when the controllers stopped being able to quickly follow the SP. The results for the first test are shown in Tables 5.3 and 5.7. In summary, these results show that there is no significant difference between all controllers, the two controllers found with the Autotuning method were able to follow the chirp signal until 1.06 Hz of frequency. Closer inspection of Table 5.7 shows that controllers were able to follow the signal until 1.08 Hz - 1.18 Hz, with controllers 4-7 presenting a slightly better performance. Surprisingly these controllers are the ones found with the step as the setpoint signal. From all controllers found, the one with the better outcome for this test is controller 4 found with the GA optimization with a step signal as the setpoint and minimizing the IAE index.

The results of the second test in this section are more qualitative, so a summary table (Table 5.9) was created to allow for a better comparison of all controllers. The only difference between controllers 1 and 2 is the amount of time required to settle at 1.6 Hz, with the first requiring more time, and for higher frequencies, both controllers behave the same. Surprisingly both controllers were able to settle at 1.8 Hz presenting some overshoot at the peaks, this can be explained by the combination of a few conditions: the system did not converge for the 1.7 Hz but it was very close to converging meaning that it did not have enough time to converge; Comparing with the first group (controllers 1 and 2) at 1.6 Hz, controller 3-7 and controller 12 had a better settling time, presenting any difficulty in following the wave. Controllers 3-7 had a better performance, in comparison with the others, for higher frequencies: for 1.7 Hz they settle in 0.5 seconds, for 1.8 they settled in 1 second, and for 1.9 Hz and 2 Hz controller 3 was not able to settle while controller 4-7 were able to settle in less than 1 second presenting only a residual phase displacement. It is interesting how controllers 4-7 outperformed the others in this survey, considering that they were tuned with a step signal.

Since some controllers showed a displacement with the setpoint, it was important to analyze this behavior, and thus the delay and RMSE between the SP and the PV were measured. This test was run for 4 seconds with a sine wave at 1 Hz, and the results are shown in Tables 5.4 and 5.8. Controllers 1 and 2 presented a delay of 31 milliseconds and 30 milliseconds, respectively, regarding the sine wave while the GA found controllers presented delays ranging from 27 to 42 milliseconds. Comparing the ones tuned with the step signal and those tuned with the sine wave, the first group performed the worst. Controllers 8-12, which were tuned with a sine wave, performed better and had lower delays, as predicted. Since these controllers were tuned with the sine wave, it is presumed that they will have better outcomes when following a signal with the same characteristics. Regarding the RMSE values, the controllers found with the Autotuning method had the worse result and consequently provided the worse fitting to the SP. Once again, controllers 8-12 demonstrated to be more fitted to the wave presenting reduced values of RMSE. For this section, the best controllers regarding the delay are controller 8 and 11, and regarding the RMSE value are controllers 10 and 12, all of these controllers were found using the GA optimization tuned with the sine wave.

Even though this series of tests was not definitive in the sense that it did not lead to a single best controller, it did allow for a clearer understanding of the output potential of the discovered controllers and determining whether or not they met the standards. All of these controllers produced satisfactory performance, meeting the minimum requirements for settling time (all less than 0.5 seconds), overshoot (all less than 10%), and successfully following signals at 1 Hz. It was assumed that controllers discovered through the GA optimization would have better performances when comparing with the ones found with the Autotuning wizard, once the former is a more complex approach taking into account the controller performance indexes. The results reinforced this assumption.

Considering the results of all tests, it was surprising that the controllers tuned with the step as the setpoint performed the best. Despite the fact that controllers 8-12 were tuned using the ideal setpoint (a sine wave with 1 Hz) they did not perform as well as controllers 3-7 in the output tests, only outperforming them in the delay and RMSE results. This outcome can be explained by the controllers tuned with the step being slightly faster than the others and therefore more capable of following the setpoint to higher frequencies.

The results in this chapter show that the controllers found are capable of meeting the system's minimum requirements, with the group of GA optimized controllers tuned with a step presenting the best performance; however, evaluating these with a specified pedal trajectory is the best way to choose the best option.

Table 5.9: Summarized results of the Manual Frequency test for all controllers 1 to 12.

<table>
<tr><th>Controller
Frequency (Hz)</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th></tr>
<tr><td>1.0</td><td colspan="2" rowspan="6">Settling time: 0.5 s</td><td colspan="10" rowspan="6">Settling time: around 0.5 s</td></tr>
<tr><td>1.1</td></tr>
<tr><td>1.2</td></tr>
<tr><td>1.3</td></tr>
<tr><td>1.4</td></tr>
<tr><td>1.5</td></tr>
<tr><td>1.6</td><td>Settling time: 1.75 s</td><td>Settling time:1 s</td><td colspan="5">Settling time: 0.5 s</td><td>Settling time: 1s</td><td>Settling time: 1 s</td><td>Settling time:
1s</td><td>Settling time:
1 s</td><td>Settling time:
0.5 s</td></tr>
<tr><td>1.7</td><td colspan="2">Does not settle
Peaks overshoot
Significant Phase Displacement</td><td colspan="5">Settling time: 0.5</td><td>Settling time:1.75s
Peaks overshoot</td><td>Settling time˜
2 s</td><td>Settling time:
1.75 s
Peaks overshoot˜</td><td>Does not settle
Peaks overshoot
Significant Phase displacement</td><td>Settling time:
1 s</td></tr>
<tr><td>1.8</td><td colspan="2">Settling time: 0.5 s
Peaks overshoot</td><td colspan="5">Settling time:
1 s</td><td colspan="5" rowspan="3">Does not settle
Peaks overshoot
Significant phase displacement</td></tr>
<tr><td>1.9</td><td colspan="2" rowspan="2">Does not settle
Peaks overshoot and significant phase displacement</td><td rowspan="2">Does not settle
Peaks overshoot and significant phase displacement</td><td colspan="4" rowspan="2">Residual phase displacement</td></tr>
<tr><td>2.0</td></tr>
</table>

Chapter 6

Trajectory Results

6.1 PID Selection

As described in Section 4.1.3, it was essential to further test the controllers with the foot gait trajectory to test their performance with a real trajectory instead of sinusoidal signals. The measured values for RMSE, and delay between the device response and the setpoint are available in Table 6.1.

Whereas the first group of controllers (controllers 3-7) tends to have reduced values for the error, the second group (controllers 8-12) has better performance regarding the presented delay between the SP and the PV. The controllers leading to the minimum error are controllers 4 and 6 with the RMSE of 2.66$\cdot 10^{-2}$ and the controller with the smallest delay is controller 8 with 110 milliseconds of minimum detected delay between the signal and the setpoint. There was the need to further analyze the system response with these 3 controllers. In Figures 6.1, 6.2 and 6.3 are presented the system response with the foot gait trajectory with controllers 4, 6 and 8 , respectively.

Both Figures 6.1 and 6.2 share a number of key features, as they both have responses quite closely with the setpoint. Both controllers allow the system to quickly converge to the setpoint, taking around 0.5 seconds to converge with little overshoot, and after settling the system does not present any ringing behavior. What is striking about both figures are the delays presented at the peaks, meaning that these controllers have difficulties in following sharp forms in the trajectory even though they are acceptable in terms of settling time.

Now, looking at Figure 6.3, which corresponds to the system response with controller 8, it has sig-

Table 6.1: Delay and RMSE results with the foot gait trajectory as setpoint for controllers found with GA optimization.

Controller	3	4	5	6	7	8	9	10	11	12
Delay (samples)	13	15	14	15	14	11	12	12	14	12
Delay (seconds)	1.30$\cdot 10^{-1}$	1.50$\cdot 10^{-1}$	1.40$\cdot 10^{-1}$	1.50$\cdot 10^{-1}$	1.40$\cdot 10^{-1}$	1.10$\cdot 10^{-1}$	1.20$\cdot 10^{-1}$	1.20$\cdot 10^{-1}$	1.40$\cdot 10^{-1}$	1.20$\cdot 10^{-1}$
RMSE	2.69 $\cdot 10^{-2}$	2.66$\cdot 10^{-2}$	2.68$\cdot 10^{-2}$	2.66$\cdot 10^{-2}$	2.68$\cdot 10^{-2}$	2.79$\cdot 10^{-2}$	2.86$\cdot 10^{-2}$	2.72$\cdot 10^{-2}$	2.89$\cdot 10^{-2}$	2.72$\cdot 10^{-2}$

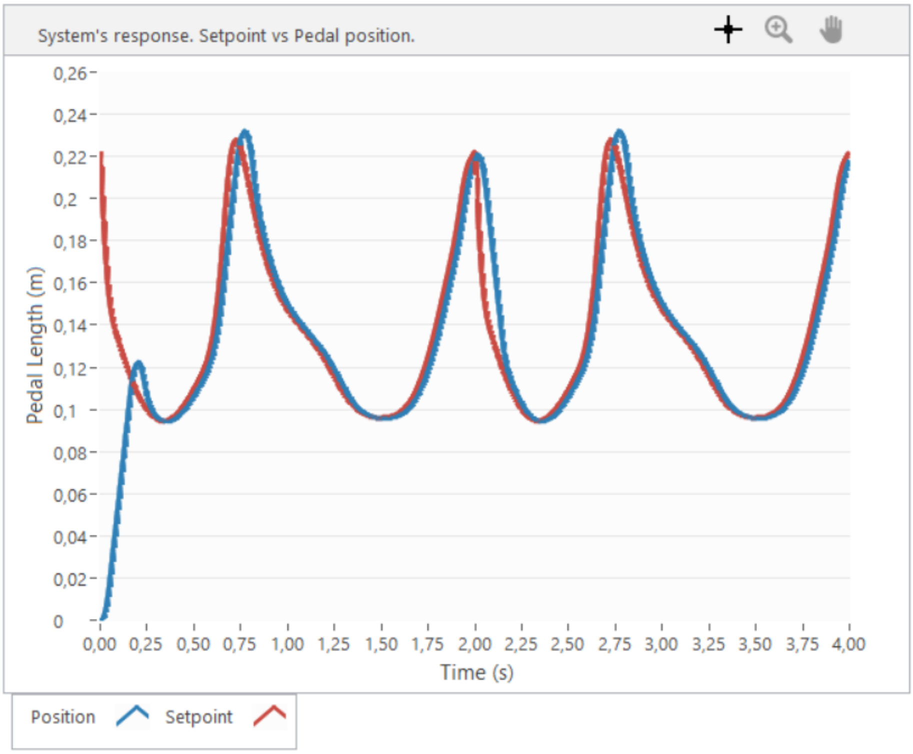


Figure 6.1: System response with controller 4 to the foot gait trajectory.

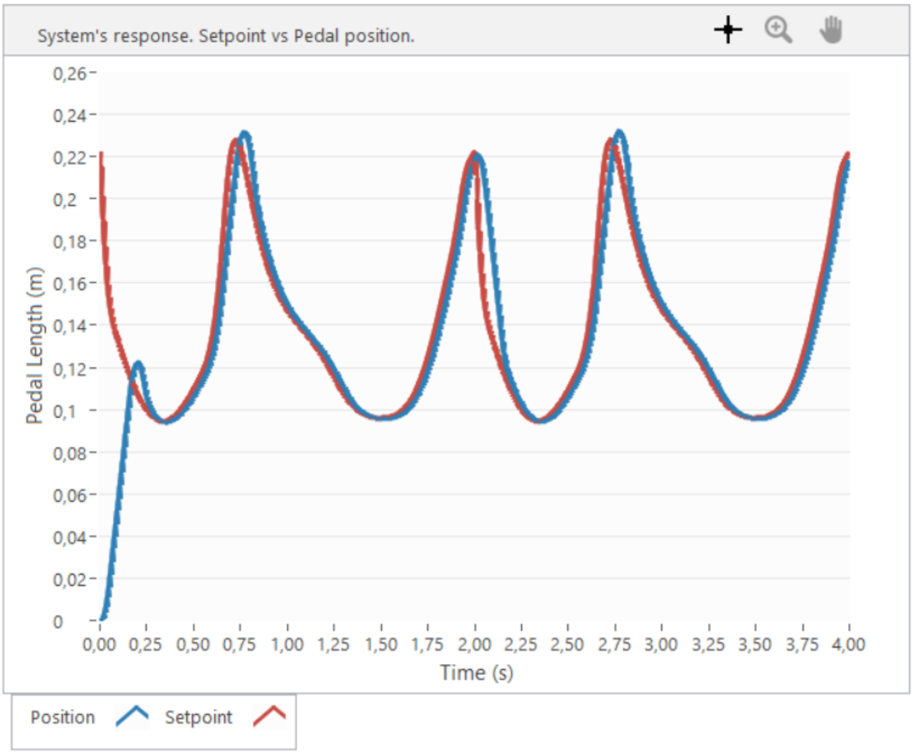


Figure 6.2: System response with controller 6 to the foot gait trajectory.

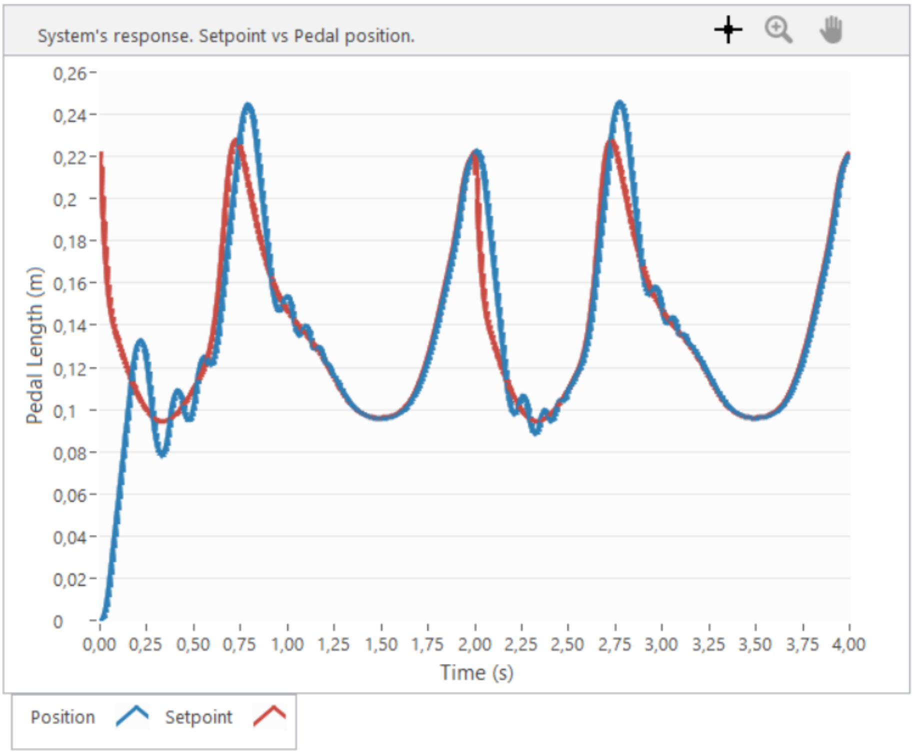


Figure 6.3: System response with controller 8 to the foot gait trajectory.

nificant differences when comparing with the last two analyzed controllers. Here, the system converges only after 0.75 seconds presenting a significant overshoot and ringing. At the peaks, there is present a delay, however bigger than the ones found with the other controllers, after the peaks the system once again shows difficulties in settling to the setpoint signal. After the convergence, nearly 1.25 seconds of simulation, it can be seen that there is almost no delay between the blue and the red line in the graphic justifying the smallest delay presented in Table 6.1. Apart from this positive feature of the system response, the overall response of this controller is not suitable.

From the data obtained in the step response characteristics test, provided in Table 5.6, it is revealed that controller 6 had the best results regarding the settling time, overshoot, and peak time. The fact that controller 6 was the fastest controller between all found is relevant to explain the proper response when subjected to a trajectory. Additionally, from Table 5.9, where it is presented the results from all controllers in the manual frequency shifting test, controller 6 was in the group of controllers with the best response, being able to follow up to 1.8 Hz without difficulties and only presenting phase displacement in sinusoidal signals with 1.9 and 2 Hz. These results prove that controller 6 is a better fit for the system than controller 4, and for this reason, it will be used in all remaining tests.

6.2 Trajectory tracking results

6.2.1 Circular Trajectory

The circular trajectory was defined by selecting 9 points in the delimited pedal range as seen in Figure 6.4. For this test two frequencies were selected: 0.5 Hz to test the system at an appropriate pace for walking therapy and 1 Hz to test the system at its maximum frequency.

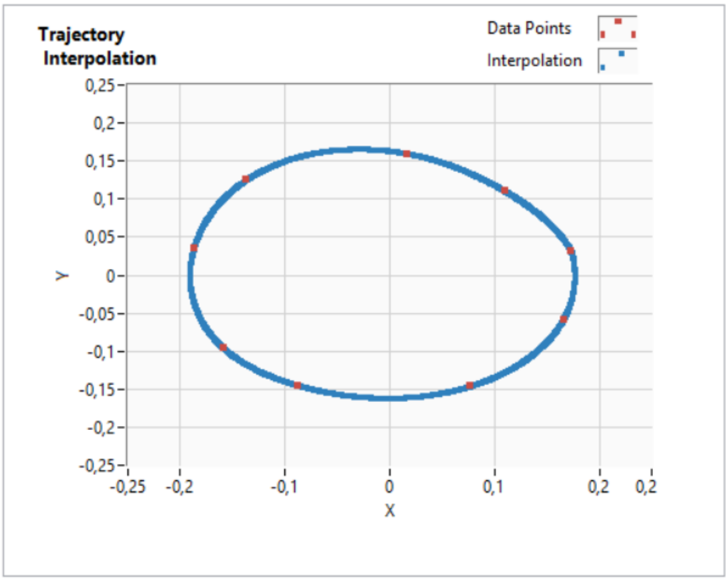


Figure 6.4: XY graph containing the data points selected to define the trajectory in blue and the interpolated trajectory in red.

6.2.1.A Frequency of 0.5 Hz

The simulation time was 10 seconds which resulted in 5 trajectory cycles within the simulation. In Figure 6.5 it is represented the system's response to the interpolated trajectory represented in Figure 6.4. The red line is the calculated values of L from the trajectory algorithm (desired values) and in blue is the system's response. Even though the simulation time was defined for 10 seconds, for simplicity it is represented in the graph the first 4 seconds of simulation as this is the more relevant part of the simulation, where the system is converging to the setpoint, and enough to represent the system's performance throughout the simulation.

Figure 6.5 shows nearly 2 cycles. The first cycle finishes at 2.50 seconds and it can be seen that the system takes around 0.25 seconds to converge with the SP. The settling time represents 10% of the first cycle which is a satisfactory result.

The system seems to react adequately to differences in the pedal length. The RMSE and the delay were calculated and the results are at Table 6.2. The error between the SP and the PV is 1.60$\cdot 10^{-3}$,

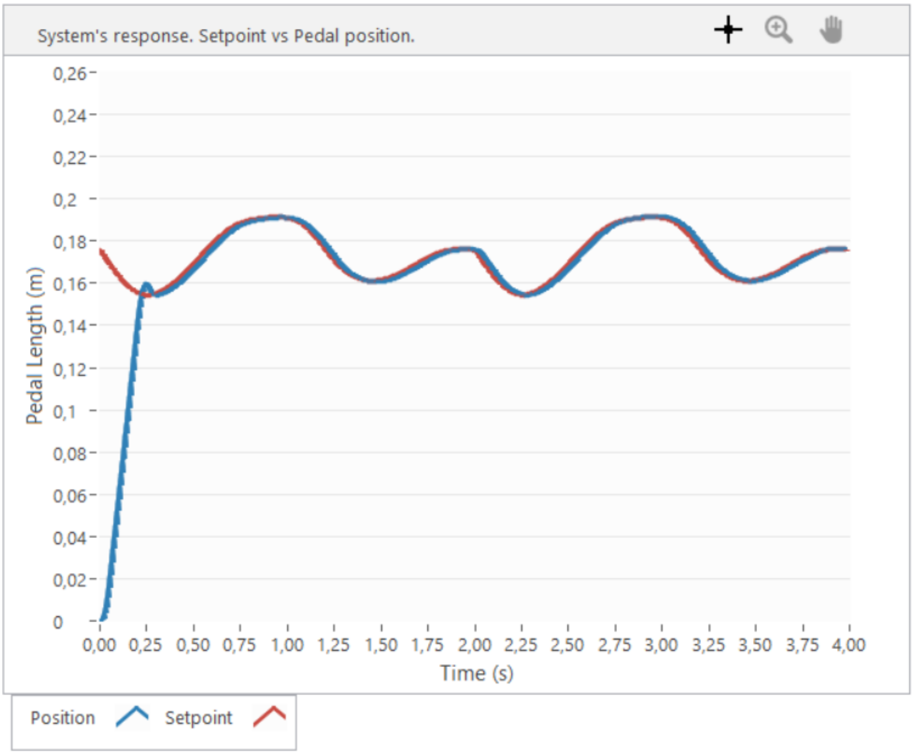


Figure 6.5: Graph of pedal length (meters) versus simulation time (seconds) at 0.5 Hz. The setpoint is shown in red, while the system's response is shown in blue.

Table 6.2: RMSE and delay results for the circular trajectory at 0.5 Hz.

Trajectory	Circular
RMSE	1.60$\cdot 10^{-3}$
Delay (s)	0

which is the smallest error discovered thus far. The lowest delay calculated between the two signals is zero, indicating an ideal response.

The cross-correlation of both signals is present in Figure 6.5, where the x-axis is the lag and the y axis the normalized cross-correlation values. There was a correlation of 1 at lag 0 which means that both signals are extremely similar and there is not a time delay between the signals, as the lag increases to both positive and negative extremes, the correlation value decreases smoothly. The symmetrical shape of the graph shows that there is not a random correlation between the signals and so that they are strongly related.

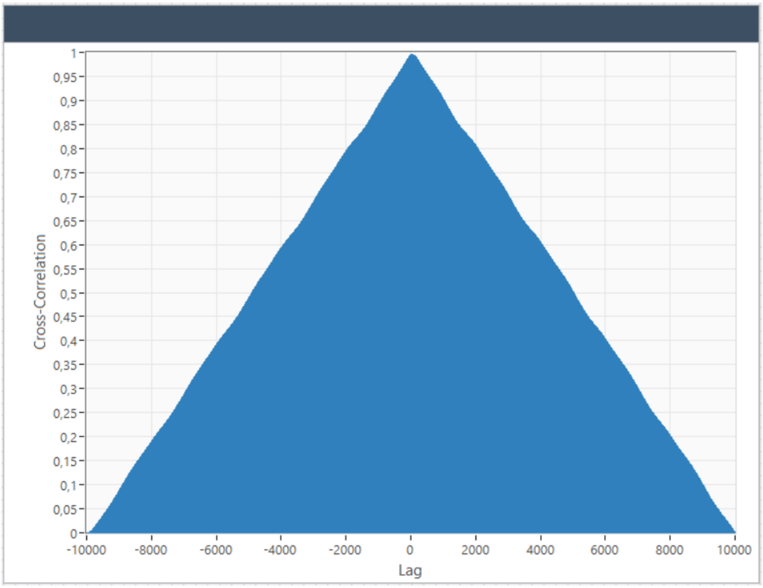


Figure 6.6: Correlogram of the system response to the circular trajectory at 0.5 Hz. In the y axis, the cross-correlation values are presented, and in the x axis the lag.

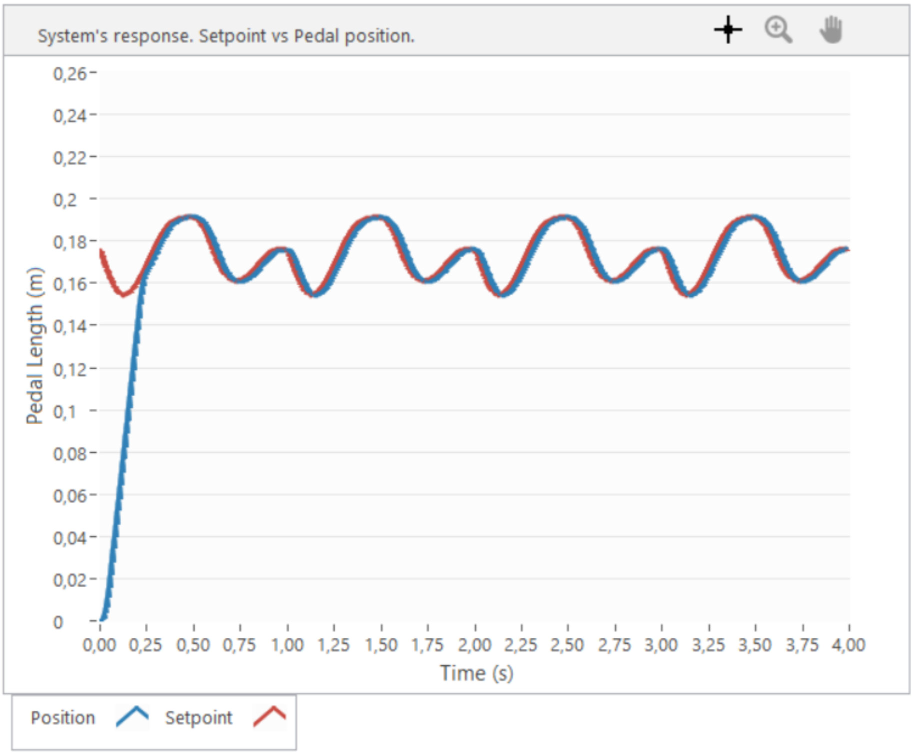


Figure 6.7: Pedal length (meters) versus simulation time (seconds) graph at 1 Hz, with the setpoint and the system's response represented.

Table 6.3: RMSE and delay results for the circular trajectory at 1 Hz.

Trajectory	Circular
RMSE	3.20$\cdot 10^{-3}$
Delay (s)	0

6.2.1.B Frequency of 1 Hz

In this test, the simulation time was 10 seconds once again but it resulted in 10 trajectory cycles. Figure 6.7 depicts 4 cycles, with the first cycle completing in one second of simulation. The first thing to notice is that the system takes 0.25 seconds to reach the setpoint which is nearly 50% of the first cycle lost as the system tries to converge.

Even though the system reacts adequately to sudden differences in the pedal length (as presented at 0.9 seconds of simulation) there is still a minor delay between the two signals. For this reason, the parameters presented in Table 6.3 were calculated, allowing us to better assess the trajectory tracking performance of the cycling system. The calculated value for the RMSE was 3.7$\cdot 10^{-3}$ indicating that the error between the system response and the trajectory data is residual, while the minimum delay found was 0 which is the optimum result.

The cross-correlation of both signals is depicted in Figure 6.8. There was a correlation of 1 at lag

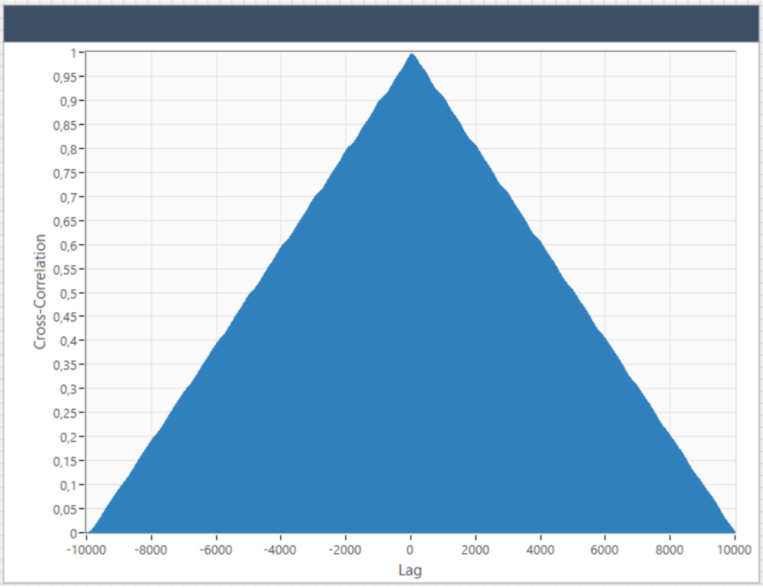


Figure 6.8: Correlogram of the system response to the circular trajectory at 1 Hz. In the y axis it is represented the cross-correlation values and in the x axis the lag.

0, demonstrating that both signals are extremely similar and that there is no time delay between them. As the lag increases to both positive and negative extremes, the correlation value declines smoothly, similar to the result obtained at a lower frequency. The symmetrical structure of the graph indicates, once again, that the correlation between the signals is not random and that they are tightly related.

Overall, these findings show that the device works adequately for simple trajectories, such as a circle, at a suitable walking speed for therapy of 0.5 Hz and for the maximum desired speed of 1 Hz.

6.2.2 Foot Gait Trajectory

The foot gait trajectory, defined in Section 4.1.2, was already employed to find the optimum controller between the ones found at Chapter 5. Here, this trajectory is analysed in other perspective, to evaluate the quality of the system response when following this trajectory at different frequencies. Firstly, the results at 0.5 Hz, the adequate velocity for gait therapy, were checked and then it was tested for 1 Hz, the system's maximum velocity.

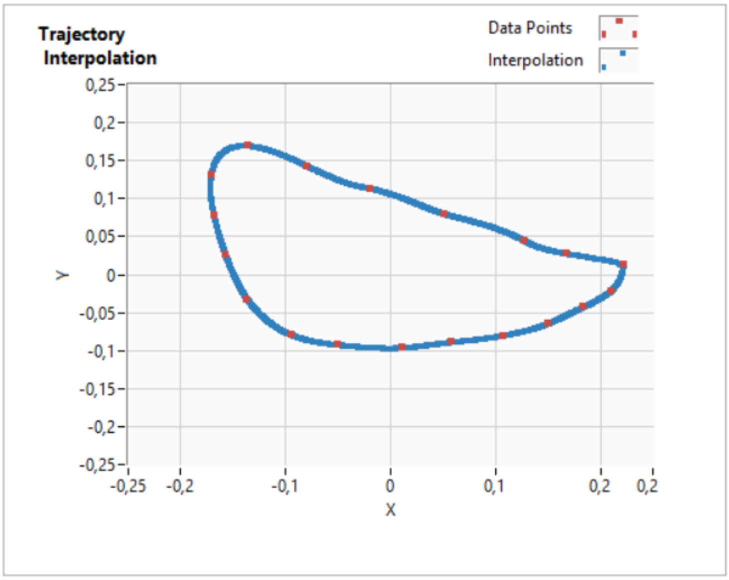


Figure 6.9: XY graph containing the data points selected to define the trajectory in blue and the interpolated trajectory in red.

6.2.2.A Frequency of 0.5 Hz

In Section 6.1 this trajectory was tested with several controllers at 0.5 Hz which contributed to the selection of the actual controller as the optimum. For this reason, the system is validated for smaller frequencies with this controller and with this trajectory. In Figure 6.2, from Section 6.1, nearly two trajectory cycles of this system with the foot gait trajectory at 0.5 Hz are represented. Closer inspection of this figure shows that the system needs 0.30 seconds to settle, which is 0.15% of the first cycle that finishes at 2 seconds. This result is not ideal, nonetheless it can be accepted. The delay presented at the peaks were the worst aspect of the system response, demonstrating difficulties to follow this trajectory at this frequency.

The RMSE value and the delay were already calculated and presented at Table 6.1. The system presented a RMSE of 2.66$\cdot 10^{-2}$ and the minimum delay found between the SP and the PV is 1.50$\cdot 10^{-1}$ seconds.

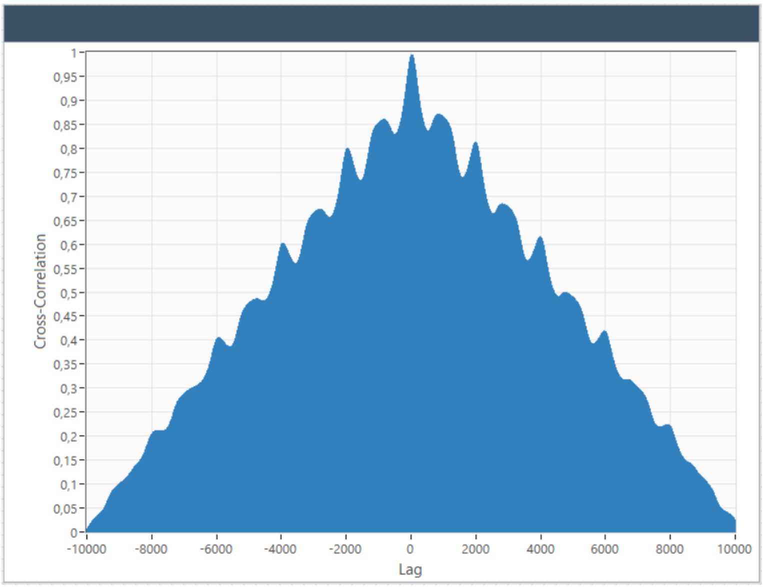


Figure 6.10: Correlogram of the system response with the foot gait trajectory at 0.5 Hz as input. In the y axis, the cross-correlation values are represented, and in the x axis the lag.

The correlogram obtained, presented in Figure 6.10, goes accordingly with the quality of the rest of the results. It can be seen that at lag 0 the cross-correlation is almost 1 however it is a narrow spike indicating that both signals have only some areas that are similar. The oscillating pattern, as the cross-correlation values does not diminish smoothly, is caused by the combination of extremely similar and divergent regions of the two signals. When both signals are divergent (at the peaks), the cross-correlation is lower, and the opposite is true when the signals converge.

6.2.2.B Frequency of 1 Hz

Figure 6.11 presents the system response to the trajectory at 1 Hz.

The simulation time was set to 10 seconds, nevertheless, for simplicity, it is only shown the first 4 seconds of simulation allowing the analysis of the system settling and the stabilized response. The pedal length reaches the desired value in approximately 0.25 seconds of simulation, taking 25% of the first cycle. What is noteworthy about Figure 6.11 is the discrepancy presented at the peaks, that coincide with the sharp areas in the trajectory. Because of the high frequency of the signal, the trajectory quickly changes in these sharp areas and the system is not fast enough to converge without causing a delay. This result is confirmed by the data presented at Table 6.4, with the minimum delay found of 3.90 seconds which is a significant amount. The RMSE result was of 4.21$\cdot 10^{-2}$ which was a rather worse result than the obtained with this controller and with the same trajectory in Section 6.1 but with a lower frequency. However, taking into consideration the previous results, even though at half velocity of the one in this section, the controller had already indicated that it was not fast enough in following the foot

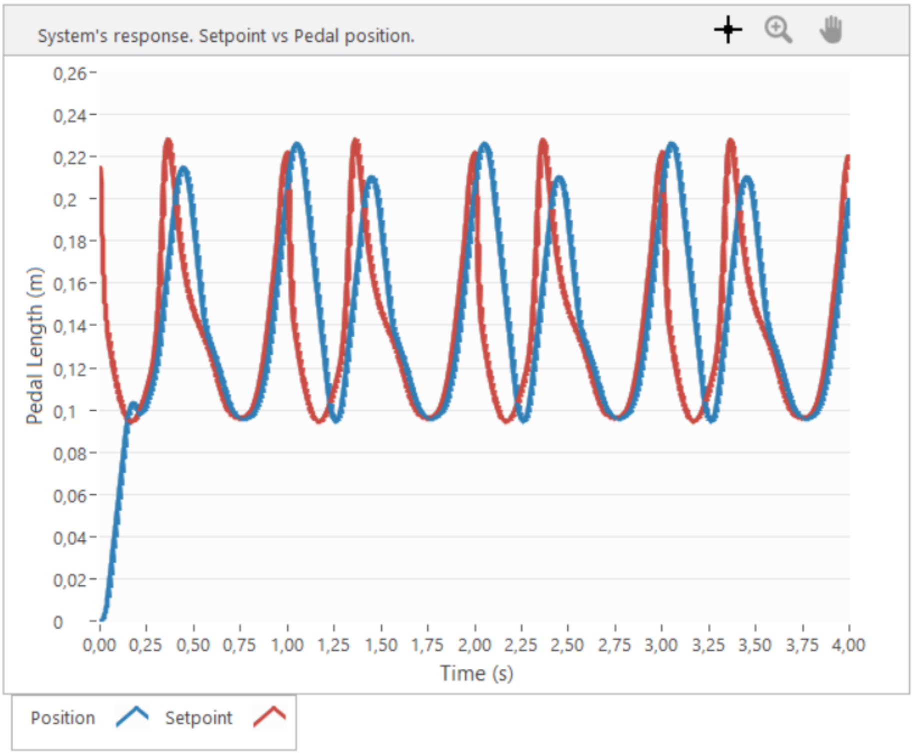


Figure 6.11: Pedal length (meters) versus simulation time (seconds) graph at 1 Hz. In red is represented the setpoint and in blue the system's response.

Table 6.4: Delay and RMSE result for foot gait trajectory at 1 Hz.

Trajectory	Foot Gait
RMSE	4.21$\cdot 10^{-2}$
Delay (s)	3.9$\cdot 10^{-1}$

gait trajectory as it is necessary.

The correlogram in Figure 6.12 demonstrates the low quality of the system reaction to the trajectory. At lag 0, the cross-correlation is approximately one, although it is a narrow spike, indicating that both signals share only a few areas. The symmetrical decrease of the cross-correlation values, when the lag goes to extremes, shows the fragile similarity between the two signals. However, the lack of smoothness of the correlogram's shape displays the oscillatory unanimity of the system response with the defined trajectory.

In summary, the results for the system with the foot gait trajectory at 0.5 and 1 Hz show that the best controller chosen, from the set of controllers discovered using diverse approaches, does not have a satisfying behaviour for complex trajectories.

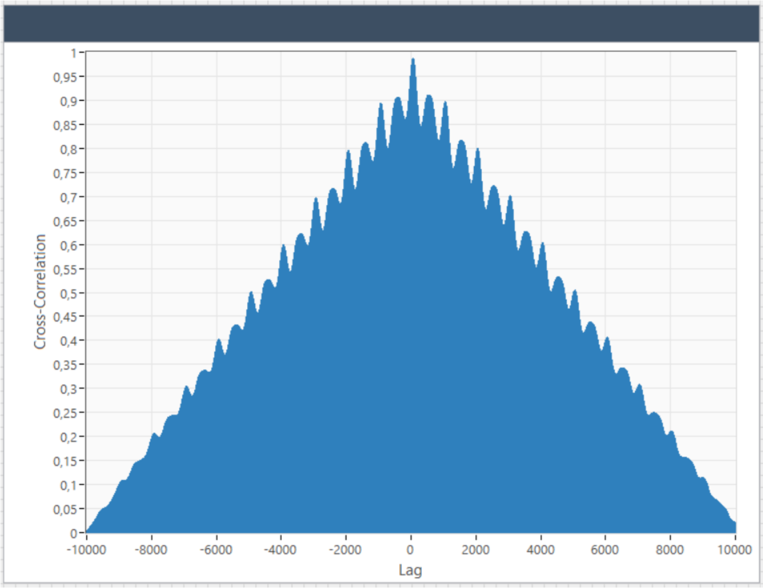


Figure 6.12: Correlogram of the system response with the foot gait trajectory as input. In the y axis, the cross-correlation values is represented, and in the x axis the lag.

Chapter 7

Conclusion

7.1 Discussion

The proposed goal for this thesis was to further improve the Haptic Cycling Trainer by transferring the previously established system from the Simulink environment to LabVIEW software once it was more suitable for developing the user interface, as well as allowing the use of the CompactRIO controller to link the plant to the device's hardware. Furthermore, the aim of finding a suitable controller using various methodologies was established, as was the goal of implementing a trajectory planning algorithm for the device using data provided by the user and implementing a Graphical User Interface.

In Chapter 2 was described the problematic of stroke and patient rehabilitation and its impact on people's lives and countries economy. It was discussed how important it is to develop new technologies to enhance the rehabilitation process and patient's quality of life after the stroke event. This background analysis allowed to pinpoint the gaps in current technology and to highlight the potential of the cycling exercise in the early stages of stroke recovery.

The description of the further development of the Haptic Cycling Trainer starts in Chapter 3. The first aim of this thesis, the implementation of the previous project developed in Simulink to the LabVIEW environment, was attained as the system was correctly implemented in the desired software. After this process, it was made an additional update of the system by implementing a position controller and an analysis of the suitable control. It was used two different approaches to tune the PID controller, one using the Autotuning Wizard from LabVIEW and the other using a Genetic Algorithm with the Optimization Toolbox from MATLAB. In general, the controllers tuned with the step signal had better performance and at the end of the study, it was chosen as the best-fit controller one discovered using the step as the setpoint. The findings of this investigation, such as the conclusion that controllers found with the GA optimization had a better performance, complement those of earlier studies which stated that the

Genetic Algorithm is a valuable tool to solve complex questions and, in specific, for PID tuning. [96] [94]

The goal of creating a trajectory algorithm was accomplished in Chapter 4. an algorithm was developed to collect the data points selected from the user to create a trajectory for the pedal to follow. A set of predefined trajectories was implemented, including one that mimics the foot gait trajectory. A clean and intuitive Graphical User Interface was designed in Section 4.2, where the user is able to select the method for the trajectory definition (between manual or predefined trajectories), to select the data points for the desired trajectory, select the frequency and simulation time. The GUI also allows to access the simulation results, track the system response, observe in real-time the pedal movement and compare with the desired trajectory while having information about the number of cycles performed and total distance traveled by the patient at the end of the simulation. As this is an ongoing project, the GUI also was designed with an area specially designated for future work in the project which allows monitoring other variables of the system.

Lastly, some tests were performed, in Chapter 6, to check the system response to two trajectories, the circular and the foot gait trajectory. The system response to the circular trajectory was ideal, with the pedal being able to successfully follow the trajectory without delays and at 0.5 Hz and at the limit frequency of 1 Hz. Nonetheless, with the foot gait trajectory, the system did not have a good enough performance for both frequencies tested, with major delays at the sharp areas of the trajectory meaning that the controller is not the fittest to the system. This result clearly indicates that even choosing the best controller between the group of 12 controllers, the system response to some trajectories is not satisfactory which raises questions about the tuning method and even if a simple PID controller is enough to control this complex system or if it will be necessary to implement a new upcoming control methodology.

Overall, it is possible to state that the main objectives for this thesis were met. The findings from this study made several contributions to the project development. To begin with, it allowed the entire system to be implemented and nearly ready for testing with the CompactRIO controller and the physical components of the system, testing that was planned to be done in this thesis but was not allowed due to COVID-19 pandemic situation. It was discovered that a PID controller is unlikely to be sufficient to control the device, implying that more comprehensive controlling methodologies are needed. A trajectory algorithm was created allowing the user to create the desired trajectories. The device now has a user interface that allows the user to choose the simulation settings and view the results. The present work has gone some way towards enhancing the development of this project, putting this system one step closer to be used in gait rehabilitation. The Haptic Cycling Trainer has the potential to be extremely useful for both therapists and patients, allowing for early-stage rehabilitation and improving patients' gait and independence while remaining accessible and affordable. In this way, it is expected for this device to overcome the current limitation of gait rehabilitation and make a positive impact in people's life after the stroke event.

7.2 Future Steps

Keeping in mind that this work is part of an ongoing project that is still in its early stages of development, there is still a lot of work to be undertaken. Now that this study has been completed, it is possible to analyze the thesis development as a whole, identify some work limitations, and state what could be done in future project development. Many modifications, simulations, and studies have been postponed due to material supply delays (due to COVID-19 pandemics) and a lack of time (i.e. the experiments with real data are usually very time-consuming). Future studies will focus on a more in-depth examination of specific processes, new proposals to explore different methods or pure interest.

From the first part of this project, the controller analysis, it is unfortunate that the chosen PID did not perform as expected, based on the results in Chapter 5 and Section 6.1, when tested with the foot gait trajectory in Section 6.2. This leads us to question if a PID controller was the best way to manage the system and if the testing could be performed differently. Further research on control strategies should be carried out to establish which would be a better approach for the system. Nonetheless, as mentioned in the device descriptions, it is intended for this device to apply different loads along the trajectory, implying that an impedance control should be applied to the system in conjunction with the already implemented position controller. It would be interesting to incorporate a hybrid control in this device, which has the advantage of allowing the user to apply a specific resistance force along a predetermined trajectory but not allowing voluntary active movements from the patient. It also allows the patient to experience different mobility patterns and aid in strengthening the patient's muscles. [46] [57] These features will be critical when using this system in rehabilitation. Another interesting control approach may be the use of bio-signals-based controls, which would enable this system to be operated in a more natural way by using the patient's EMG, which could be used as feedback in the case of stroke survivors. An EMG-triggered control may be introduced, as suggested by Krebs et al. [101], in which the patient can move the limbs without assistance at the beginning of the therapy session and robotic assistance is triggered when the EMG value exceeds a certain threshold.

The trajectory planning algorithm performed well in Chapter 6, but it had some limitations that should be addressed in the future. Modifications to the algorithm should be made to allow for the use of variable velocity during the simulation time. Following the implementation of impedance control, the algorithm should be able to process information about various loads applied in the pedal at various stages of the trajectory or simulation. Furthermore, an initialization routine must be introduced to allow the pedal to reach an initial position to begin the rehabilitation session while also converging with the trajectory implemented by the therapist. This study was created in 2020 during the COVID-19 pandemic, and it was restricted in many respects. It was planned at the beginning of the thesis to test the device and the trajectories by attaching the LabVIEW project to the CompactRIO controller, the DC motors, and the pedals with the ballscrew system. However, due to the material being trapped at the supplier's facilities and the restricted amount of time available in the robotics laboratory, it was not possible to continue with this part of the research. In this scenario, future research should certainly put the work learned in this project to the test.

As a result of incorporating the previously mentioned features into the system, the Graphical User Interface must be updated. It would be useful to investigate a new approach for the therapist to plan the desired trajectory, once additional data, such as different velocities and loads at different stages of the trajectory, are provided. It would also be interesting to provide a 3D representation of the pedal in real-time rather than the currently implemented 2D display, as well as to display details about the patient's performance during the therapy session. It is critical that therapists review the GUI and provide feedback on the functionalities introduced, including the easiness of information provision and results display. This input could be extremely beneficial, allowing for the implementation of additional relevant functionalities to the rehabilitation process along with determining which way this device could be of greater benefit to therapists.

Finally, to evaluate whether this device is useful in stroke gait rehabilitation and to collect scientific evidence, it must be tested on healthy and unhealthy subjects.

Bibliography

[1] I. Díaz, J. J. Gil, and E. Sánchez, “Lower-Limb Robotic Rehabilitation: Literature Review and Challenges,” *Journal of Robotics*, vol. 2011, no. i, pp. 1–11, 2011.

[2] M. Dzahir and S.-i. Yamamoto, “Recent Trends in Lower-Limb Robotic Rehabilitation Orthosis: Control Scheme and Strategy for Pneumatic Muscle Actuated Gait Trainers,” *Robotics*, vol. 3, no. 2, pp. 120–148, apr 2014.

[3] J. F. Veneman, R. Kruidhof, E. E. G. Hekman, R. Ekkelenkamp, E. H. F. Van Asseldonk, and H. van der Kooij, “Design and Evaluation of the LOPES Exoskeleton Robot for Interactive Gait Rehabilitation,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 3, pp. 379–86, 2007.

[4] I. Khanna, A. Roy, M. M. Rodgers, H. I. Krebs, R. M. MacKo, and L. W. Forrester, “Effects of unilateral robotic limb loading on gait characteristics in subjects with chronic stroke,” *Journal of NeuroEngineering and Rehabilitation*, vol. 7, no. 1, p. 23, 2010.

[5] S. K. Banala, S. K. Agrawal, and J. P. Scholz, “Active leg exoskeleton (ALEX) for gait rehabilitation of motor-impaired patients.” *IEEE 10th International Conference on Rehabilitation Robotics*, pp. 401–407, 2007.

[6] S. Hussain, S. Q. Xie, and P. K. Jamwal, “Robust nonlinear control of an intrinsically compliant robotic gait training orthosis,” *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, vol. 43, no. 3, pp. 655–665, 2013.

[7] M. Iosa, G. Morone, M. Bragoni, D. De Angelis, V. Venturiero, P. Coiro, L. Pratesi, and S. Paolucci, “Driving electromechanically assisted Gait Trainer for people with stroke.” *Journal of rehabilitation research and development*, vol. 48, no. 2, pp. 135–46, 2011.

[8] L. Marchal-Crespo and R. Riener, *Robot-assisted gait training*. Elsevier Ltd., 2018.

[9] G. Chen, C. K. Chan, Z. Guo, and H. Yu, "A review of lower extremity assistive robotic exoskeletons in rehabilitation therapy," *Critical Reviews in Biomedical Engineering*, vol. 41, no. 4-5, pp. 343–363, 2013.

[10] M. Bouri, B. L. Gall, and R. Clavel, "A new concept of parallel robot for rehabilitation and fitness : The Lambda," *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 2503–2508, 2009.

[11] M. Girone, G. Burdea, M. Bouzit, V. Popescu, and J. E. Deutsch, "A Stewart Platform-Based System for Ankle Telerehabilitation," *Autonomous Robots*, vol. 10, no. 2, pp. 203–212, 2001.

[12] N. G. Tsagarakis, J. S. Dai, D. G. Caldwell, and I. Italiano, "A High-performance Redundantly Actuated Parallel Mechanism for," *The International Journal of Robotics Research*, vol. 28, no. 9, pp. 1226–1227, 2009.

[13] J. Nikitczuk, B. Weinberg, P. K. Canavan, and C. Mavroidis, "Active Knee Rehabilitation Orthotic Device With Variable Damping Characteristics Implemented via an Electrorheological Fluid," *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 6, pp. 952–960, 2011.

[14] Y. Ding, M. Sivak, B. Weinberg, C. Mavroidis, and M. K. Holden, "NUVABAT: Northeastern University Virtual Ankle and Balance Trainer," in *2010 IEEE Haptics Symposium, HAPTICS 2010*, 2010, pp. 509–514.

[15] B. J. Ruthenberg, N. A. Wasylewski, and J. E. Beard, "An experimental device for investigating the force and power requirements of a powered gait orthosis," *Journal of Rehabilitation Research and Development*, vol. 34, no. 2, pp. 203–213, 1997.

[16] G. Belforte, L. Gastaldi, and M. Sorli, "Pneumatic active gait orthosis," *Mechatronics*, vol. 11, no. 3, pp. 301–323, 2001.

[17] G. S. Sawicki and D. P. Ferris, "A pneumatically powered knee-ankle-foot orthosis (KAFO) with myoelectric activation and inhibition," *Journal of NeuroEngineering and Rehabilitation*, vol. 6, no. 1, p. 23, dec 2009.

[18] H. Schmidt, D. Sorowka, S. Hesse, and R. Bemhardt, "Development aspects of a robotised gait trainer for neurological rehabilitation," vol. 2, 02 2001, pp. 1340 – 1343 vol.2.

[19] M. Katan and A. Luft, "Global Burden of Stroke," *Seminars in Neurology*, vol. 38, no. 2, pp. 208–211, 2018.

[20] E. Stevens, D. E. Emmett, D. Y. Wang, P. C. McKevitt, and P. C. Wolfe, "The Burden of Stroke in Europe," Stroke Alliance for Europe, London, Tech. Rep.

[21] D. G. da Saúde, "Programa Nacional para as Doenças Cérebro-Cardiovasculares," Lisboa, Tech. Rep., 2017.

[22] G. Morone *et al.*, "Robot-assisted gait training for stroke patients: Current state of the art and perspectives of robotics," *Neuropsychiatric Disease and Treatment*, vol. 13, pp. 1303–1311, 2017.

[23] H. M. Dewey, L. J. Sherry, and J. M. Collier, "Stroke rehabilitation 2007: What should it be?" *International Journal of Stroke*, vol. 2, no. 3, pp. 191–200, 2007.

[24] D. B. Samir, "Stroke rehabilitation." *CONTINUUM: Lifelong Learning in Neurology*, vol. 23, no. 1, pp. 238–253, 2017.

[25] S. Paolucci *et al.*, "Early versus delayed inpatient stroke rehabilitation: A matched comparison conducted in Italy," *Archives of Physical Medicine and Rehabilitation*, vol. 81, no. 6, pp. 695–700, 2000.

[26] C. C. Raasch and F. E. Zajac, "Locomotor strategy for pedaling: Muscle groups and biomechanical functions," *Journal of Neurophysiology*, vol. 82, no. 2, pp. 515–525, 1999.

[27] S. J. Kim, H. Y. Cho, Y. L. Kim, and S. M. Lee, "Effects of stationary cycling exercise on the balance and gait abilities of chronic stroke patients," *Journal of Physical Therapy Science*, vol. 27, no. 11, pp. 3529–3531, 2015.

[28] D. Barbosa, C. P. Santos, and M. Martins, "The application of cycling and cycling combined with feedback in the rehabilitation of stroke patients: A review," *Journal of Stroke and Cerebrovascular Diseases*, vol. 24, no. 2, pp. 253–273, 2015.

[29] S. Hesse, J. Mehrholz, and C. Werner, "Robot-Assisted Upper and Lower Limb Rehabilitation After Stroke Walking and Arm/Hand Function," *Deutsches Ärzteblatt International[U+F8E6] [U+F8E6]Dtsch Arztebl Int*, vol. 105, no. 18, pp. 330–336, 2008.

[30] A. S. Ryan, C. L. Dobrovolny, K. H. Silver, G. V. Smith, and R. F. Macko, "Cardiovascular fitness after stroke: Role of muscle mass and gait deficit severity," *Journal of Stroke and Cerebrovascular Diseases*, vol. 9, no. 4, pp. 185–191, 2000.

[31] "Hemorrhagic Strokes (Bleeds) — American Stroke Association," accessed on: 2020-3-30. [Online]. Available: https://www.stroke.org/en/about-stroke/types-of-stroke/hemorrhagic-strokes-bleeds

[32] "Ischemic Strokes (clots) — American Stroke Association," accessed on: 2020-3-30. [Online]. Available: https://www.stroke.org/en/about-stroke/types-of-stroke/ischemic-stroke-clots

[33] D. Mozaffarian *et al.*, *Heart disease and stroke statistics-2016 update a report from the American Heart Association*, 2016, vol. 133, no. 4.

[34] Y. Béjot, H. Bailly, J. Durier, and M. Giroud, "Epidemiology of stroke in Europe and trends for the 21st century," *Presse Medicale*, vol. 45, no. 12, pp. e391–e398, 2016.

[35] D. R. Louie and J. J. Eng, "Powered robotic exoskeletons in post-stroke rehabilitation of gait: A scoping review," *Journal of NeuroEngineering and Rehabilitation*, vol. 13, no. 1, pp. 1–10, 2016.

[36] T. Yang, X. Gao, R. Gao, F. Dai, and J. Peng, "A novel activity recognition system for alternative control strategies of a lower limb rehabilitation robot," *Applied Sciences (Switzerland)*, vol. 9, no. 19, 2019.

[37] N. Liu *et al.*, "Randomized controlled trial of early rehabilitation after intracerebral hemorrhage stroke: Difference in outcomes within 6 months of stroke," *Stroke*, vol. 45, no. 12, pp. 3502–3507, 2014.

[38] A. Kamps and K. Schüle, "Cyclic movement training of the lower limb in stroke rehabilitation," *Neurologie und Rehabilitation*, vol. 11, no. 5, pp. 259–269, 2005.

[39] P. Y. Lin, J. J. J. Chen, and S. I. Lin, "The cortical control of cycling exercise in stroke patients: An fNIRS study," *Human Brain Mapping*, vol. 34, no. 10, pp. 2381–2390, 2013.

[40] A. Shariat, M. G. Najafabadi, N. N. Ansari, J. A. Cleland, M. A. Singh, A. H. Memari, R. Honarpishe, A. Hakakzadeh, M. S. Ghaffari, and S. Naghdi, "The effects of cycling with and without functional electrical stimulation on lower limb dysfunction in patients post-stroke: A systematic review with meta-analysis," *NeuroRehabilitation*, vol. 44, no. 3, pp. 389–412, 2019.

[41] M. Katz-Leurer, I. Sender, O. Keren, and Z. Dvir, "The influence of early cycling training on balance in stroke patients at the subacute stage. Results of a preliminary trial," *Clinical Rehabilitation*, vol. 20, no. 5, pp. 398–405, 2006.

[42] D. J. Reinkensmeyer *et al.*, "Tools for understanding and optimizing robotic gait training," *Journal of Rehabilitation Research and Development*, vol. 43, no. 5, pp. 657–670, 2006.

[43] J. Penalver-Andres *et al.*, "Do we need complex rehabilitation robots for training complex tasks?" *2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*, vol. 5, no. Group 4, pp. 1085–1090, 2019.

[44] G. Morone, S. Masiero, C. Werner, and S. Paolucci, "Editorial Advances in Neuromotor Stroke Rehabilitation," *BioMed Research International*, vol. 2014, 2014.

[45] R. S. Calabrò *et al.*, "Robotic gait rehabilitation and substitution devices in neurological disorders: where are we now?" *Neurological Sciences*, vol. 37, no. 4, pp. 503–514, 2016.

[46] S. L. Chaparro-Cárdenas, A. A. Lozano-Guzmán, J. A. Ramirez-Bautista, and A. Hernández-Zavala, "A review in gait rehabilitation devices and applied control techniques," *Disability and Rehabilitation: Assistive Technology*, vol. 13, no. 8, pp. 819–834, nov 2018.

[47] G. Colombo, M. Joerg, . R. Schreier, and V. Dietz, "Treadmill training of paraplegic patients using a robotic orthosis," Tech. Rep. 6.

[48] S. Jezernik, G. Colombo, T. Keller, H. Frueh, and M. Morari, "Robotic Orthosis Lokomat: A Rehabilitation and Research Tool," *Neuromodulation*, vol. 6, no. 2, pp. 108–115, 2003.

[49] S. K. Agrawal *et al.*, "Assessment of motion of a swing leg and gait rehabilitation with a gravity balancing exoskeleton," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 3, pp. 410–420, 2007.

[50] S. K. Banala, S. K. Agrawal, S. H. Kim, and J. P. Scholz, "Novel gait adaptation and neuromotor training results using an active leg exoskeleton," *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 2, pp. 216–225, 2010.

[51] S. K. Banala, S. H. Kim, S. K. Agrawal, and J. P. Scholz, "Robot assisted gait training with active leg exoskeleton (ALEX)," *Proceedings of the 2nd Biennial IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, BioRob 2008*, vol. 17, no. 1, pp. 653–658, 2008.

[52] S. Hussain, "State-of-the-art robotic gait rehabilitation orthoses: Design and control aspects," *NeuroRehabilitation*, vol. 35, no. 4, pp. 701–709, 2014.

[53] S. Freivogel, J. Mehrholz, T. Husak-Sotomayor, and D. Schmalohr, "Gait training with the newly developed 'LokoHelp'-system is feasible for non-ambulatory patients after stroke, spinal cord and brain injury. A feasibility study," *Brain Injury*, vol. 22, no. 7-8, pp. 625–632, 2008.

[54] J. Patton *et al.*, "KineAssist: Design and Development of a Robotic Overground Gait and Balance Therapy Device," *Topics in Stroke Rehabilitation*, vol. 15, no. 2: Design of Products and Environments for People with Stroke, pp. 131–139, 2008.

[55] "Measurement of Human Ankle Stiffness Using the Anklebot," in *2007 IEEE 10th International Conference on Rehabilitation Robotics, ICORR'07*, 2007, pp. 356–363.

[56] K. Bharadwaj and T. G. Sugar, "Kinematics of a robotic gait trainer for stroke rehabilitation," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2006, 2006, pp. 3492–3497.

[57] W. Meng, Q. Liu, Z. Zhou, Q. Ai, B. Sheng, and S. S. Xie, “Recent development of mechanisms and control strategies for robot-assisted lower limb rehabilitation,” *Mechatronics*, vol. 31, pp. 132–145, 2015.

[58] X. Zhang, Z. Yue, and J. Wang, “Robotics in Lower-Limb Rehabilitation after Stroke,” *Behavioural Neurology*, vol. 2017, 2017.

[59] “A Review on Lower Limb Rehabilitation Exoskeleton Robots,” *Chinese Journal of Mechanical Engineering (English Edition)*, vol. 32, no. 1, 2019.

[60] E. Sariyildiz, G. Chen, and H. Yu, “An Acceleration-Based Robust Motion Controller Design for a Novel Series Elastic Actuator,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 3, pp. 1900–1910, mar 2016.

[61] L. Marchal-Crespo and D. J. Reinkensmeyer, “Review of control strategies for robotic movement training after neurologic injury,” *Journal of NeuroEngineering and Rehabilitation*, vol. 6, no. 1, 2009.

[62] 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,” *Medical Engineering and Physics*, vol. 36, no. 12, pp. 1555–1566, 2014.

[63] J. C. Ibarra and A. A. Siqueira, “Impedance control of rehabilitation robots for lower limbs. Review,” *Proceedings - 2nd SBR Brazilian Robotics Symposium, 11th LARS Latin American Robotics Symposium and 6th Robocontrol Workshop on Applied Robotics and Automation, SBR LARS Robocontrol 2014 - Part of the Joint Conference on Robotics and Intelligent Systems,*, pp. 235–240, 2015.

[64] B. Koopman, E. H. Van Asseldonk, and H. Van Der Kooij, “Selective control of gait subtasks in robotic gait training: Foot clearance support in stroke survivors with a powered exoskeleton,” *Journal of NeuroEngineering and Rehabilitation*, vol. 10, no. 1, pp. 1–21, jan 2013.

[65] E. T. Wolbrecht, V. Chan, V. Le, S. C. Cramer, D. J. Reinkensmeyer, and J. E. Bobrow, “Real-time computer modeling of weakness following stroke optimizes robotic assistance for movement therapy,” *Proceedings of the 3rd International IEEE EMBS Conference on Neural Engineering*, pp. 152–158, 2007.

[66] M. Mihelj, T. Nef, and R. Riener, “A novel paradigm for patient-cooperative control of upper-limb rehabilitation robots,” *Advanced Robotics*, vol. 21, no. 8, pp. 843–867, aug 2007.

[67] C. Castellini, P. Van Der Smagt, G. Sandini, and G. Hirzinger, “Surface EMG for force control of mechanical hands,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2008, pp. 725–730.

[68] T. Eiammanussakul and V. Sangveraphunsiri, “A Lower Limb Rehabilitation Robot in Sitting Position with a Review of Training Activities,” *Journal of Healthcare Engineering*, vol. 2018, no. Rehabilitation Robotics and Systems Special Issue, pp. 1–18, 2018.

[69] M. F. Bruni, C. Melegari, M. C. De Cola, A. Bramanti, P. Bramanti, and R. S. Calabrò, “What does best evidence tell us about robotic gait rehabilitation in stroke patients: A systematic review and meta-analysis,” *Journal of Clinical Neuroscience*, vol. 48, pp. 11–17, 2018.

[70] J. Mehrholz, S. Thomas, C. Werner, J. Kugler, M. Pohl, and B. Elsner, “Electromechanical-assisted training for walking after stroke,” Tech. Rep. 5, may 2017.

[71] G. Morone *et al.*, “Who May Have Durable Benefit From Robotic Gait Training?” *Stroke*, vol. 43, no. 4, pp. 1140–1142, apr 2012.

[72] S. Maeshima *et al.*, “Efficacy of a hybrid assistive limb in post-stroke hemiplegic patients: A preliminary report,” *BMC Neurology*, vol. 11, sep 2011.

[73] N. N. Byl, “Mobility training using a bionic knee orthosis in patients in a post-stroke chronic state: A case series,” *Journal of Medical Case Reports*, vol. 6, 2012.

[74] D. E. Uçar, N. Paker, and D. Buğdayci, “Lokomat: A therapeutic chance for patients with chronic hemiplegia,” *NeuroRehabilitation*, vol. 34, no. 3, pp. 447–453, 2014.

[75] A. C. Lo and E. W. Triche, “Improving gait in multiple sclerosis using robot-assisted, body weight supported treadmill training,” *Neurorehabilitation and Neural Repair*, vol. 22, no. 6, pp. 661–671, nov 2008.

[76] C. Vaney *et al.*, “Robotic-assisted step training (Lokomat) not superior to equal intensity of overground rehabilitation in patients with multiple sclerosis,” *Neurorehabilitation and Neural Repair*, vol. 26, no. 3, pp. 212–221, mar 2012.

[77] A. Domingo and T. Lam, “Reliability and validity of using the Lokomat to assess lower limb joint position sense in people with incomplete spinal cord injury,” *Journal of NeuroEngineering and Rehabilitation*, vol. 11, no. 1, p. 167, dec 2014.

[78] K. Van Kammen, A. Boonstra, H. Reinders-Messelink, and R. Den Otter, “The combined effects of body weight support and gait speed on gait related muscle activity: A comparison between

walking in the lokomat exoskeleton and regular treadmill walking," *PLoS ONE*, vol. 9, no. 9, sep 2014.

[79] C. Krewer, F. Müller, B. Husemann, S. Heller, J. Quintern, and E. Koenig, "The influence of different Lokomat walking conditions on the energy expenditure of hemiparetic patients and healthy subjects," *Gait and Posture*, vol. 26, no. 3, pp. 372–377, sep 2007.

[80] T. G. Hornby, D. D. Campbell, J. H. Kahn, T. Demott, J. L. Moore, and H. R. Roth, "Enhanced gait-related improvements after therapist- versus robotic-assisted locomotor training in subjects with chronic stroke: A randomized controlled study," *Stroke*, vol. 39, no. 6, pp. 1786–1792, jun 2008.

[81] M. D. Lewek, T. H. Cruz, J. L. Moore, H. R. Roth, Y. Y. Dhaher, and T. G. Hornby, "Allowing intralimb kinematic variability during locomotor training poststroke improves kinematic consistency: A subgroup analysis from a randomized clinical trial," *Physical Therapy*, vol. 89, no. 8, pp. 829–839, aug 2009.

[82] A. Picelli *et al.*, "Robot-assisted gait training in patients with parkinson disease: A randomized controlled trial," *Neurorehabilitation and Neural Repair*, vol. 26, no. 4, pp. 353–361, may 2012.

[83] O. Stoller, M. Schindelholz, L. Bichsel, and K. J. Hunt, "Cardiopulmonary responses to robotic end-effector-based walking and stair climbing," *Medical Engineering and Physics*, vol. 36, no. 4, pp. 425–431, apr 2014.

[84] M. Goffredo *et al.*, "Stroke gait rehabilitation: A comparison of end-effector, overground exoskeleton, and conventional gait training," *Applied Sciences (Switzerland)*, vol. 9, no. 13, jul 2019.

[85] I. Schwartz and Z. Meiner, "Robotic-Assisted Gait Training in Neurological Patients: Who May Benefit?" *Annals of Biomedical Engineering*, vol. 43, no. 5, pp. 1260–1269, may 2015.

[86] F. Molteni, G. Gasperini, G. Cannaviello, and E. Guanziroli, "Exoskeleton and End-Effector Robots for Upper and Lower Limbs Rehabilitation: Narrative Review," *PM and R*, vol. 10, no. 9, pp. S174–S188, 2018.

[87] M. P. Van Nunen, K. H. Gerrits, M. Konijnenbelt, T. W. Janssen, and A. De Haan, "Recovery of walking ability using a robotic device in subacute stroke patients: A randomized controlled study," *Disability and Rehabilitation: Assistive Technology*, vol. 10, no. 2, pp. 141–148, mar 2015.

[88] W. Wang *et al.*, "Neural Interface Technology for Rehabilitation: Exploiting and Promoting Neuroplasticity," pp. 157–178, feb 2010.

[89] F. Nocchi *et al.*, “Brain network involved in visual processing of movement stimuli used in upper limb robotic training: An fMRI study,” *Journal of NeuroEngineering and Rehabilitation*, vol. 9, no. 1, p. 49, jul 2012.

[90] Z. Yue, X. Zhang, and J. Wang, “Hand Rehabilitation Robotics on Poststroke Motor Recovery,” *Behavioural Neurology*, vol. 2017, 2017.

[91] M. Pazzaglia, G. Galli, G. Scivoletto, and M. Molinari, “A Functionally Relevant Tool for the Body following Spinal Cord Injury,” *PLoS ONE*, vol. 8, no. 3, p. e58312, mar 2013.

[92] A. Mirelman, B. L. Patritti, P. Bonato, and J. E. Deutsch, “Effects of virtual reality training on gait biomechanics of individuals post-stroke,” *Gait and Posture*, vol. 31, no. 4, pp. 433–437, apr 2010.

[93] C. Knospe, “Pid control,” *IEEE Control Systems Magazine*, vol. 26, no. 1, pp. 30–31, 2006.

[94] S. Katoch, S. S. Chauhan, and V. Kumar, *A review on genetic algorithm: past, present, and future*. Multimedia Tools and Applications, 2020.

[95] D. S. Pereira and J. O. Pinto, “Genetic Algorithm based system identification and PID tuning for optimum adaptive control,” *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, vol. 1, no. 67, pp. 801–806, 2005.

[96] P. Y. Huang and Y. Y. Chen, “Design of PID controller for precision positioning table using genetic algorithms,” *Proceedings of the IEEE Conference on Decision and Control*, vol. 3, no. December, pp. 2513–2514, 1997.

[97] N. Goldstein, “Process control,” *BioCycle*, vol. 49, no. 8, pp. 1–6, 2008.

[98] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modeling, Planning, and Control*, 1st ed. Springer-Verlag London, 2009, no. 1.

[99] J. H. Mathews and K. D.Fink, *Numerical Methods Using MATLAB*, 4th ed. Upper Saddle River, NJ: Prentice-Hall, Inc., 2004. [Online]. Available: www.journal.uta45jakarta.ac.id

[100] M. Grasselli and D. Pelinovsky, *Numerical Mathematics*. CA: Jones and Bartlett Publishers, Inc., 2008. [Online]. Available: www.journal.uta45jakarta.ac.id

[101] H. I. Krebs *et al.*, “Rehabilitation robotics: Performance-based progressive robot-assisted therapy,” *Autonomous Robots*, vol. 15, no. 1, pp. 7–20, jul 2003.

Appendix A

Appendices

A.1 Performance Tests: sine wave 1 Hz

The results of the first set of controllers found with the sine wave with 1 Hz of frequency as mentioned in Section 3.4.2.B. These tests were performed as explained in Section 3.4.3.

Setpoint	controller	Minimization Function	Fitness Value	Proportional Gain (Kp)	Integral Gain (Ki)	Derivative Gain (Kd)
Sine Wave	A	ITSE	$5.0649{\cdot}10^{-4}$	$9.8441{\cdot}10^{3}$	5.1668	48.8645
	B	IAE	$4.6100{\cdot}10^{-2}$	$9.7545{\cdot}10^{3}$	4.5751	48.6237
	C	ISE	$4.6000{\cdot}10^{-3}$	$9.8386{\cdot}10^{3}$	6.4136	49.8423
	D	ITAE	$5.1570{\cdot}10^{-2}$	$9.9500{\cdot}10^{3}$	4.1602	49.4978
	E	MSE	$4.5999{\cdot}10^{-7}$	$9.9533{\cdot}10^{3}$	5.5371	49.9760

Table A.1: PID parameters found using the GA with a sine wave with 1 Hz as setpoint.

Controller	A	B	C	D	E
Rise Time (s)	0.1913	0.1912	0.1913	0.1913	0.1912
Settling Time (s)	0.5288	0.5263	0.5288	0.5296	0.5300
Overshoot (%)	8.4048	8.4324	8.4028	8.4064	8.4043
Peak (m)	0.2114	0.2114	0.2114	0.2114	0.2114
Peak Time (s)	0.3070	0.3070	0.3070	0.3070	0.3070

Table A.2: Step Response characteristics of controllers found with the GA algorithm with a sine wave with 1 Hz.

All controllers were able to follow the chirp signal up to 1.07 Hz.

A.2 Performance Tests: Autotuning Chirp Signal Test

In this section is represented the system response of Controller 2 in the performance test mentioned in Section 5.1.2 with the chirp signal.

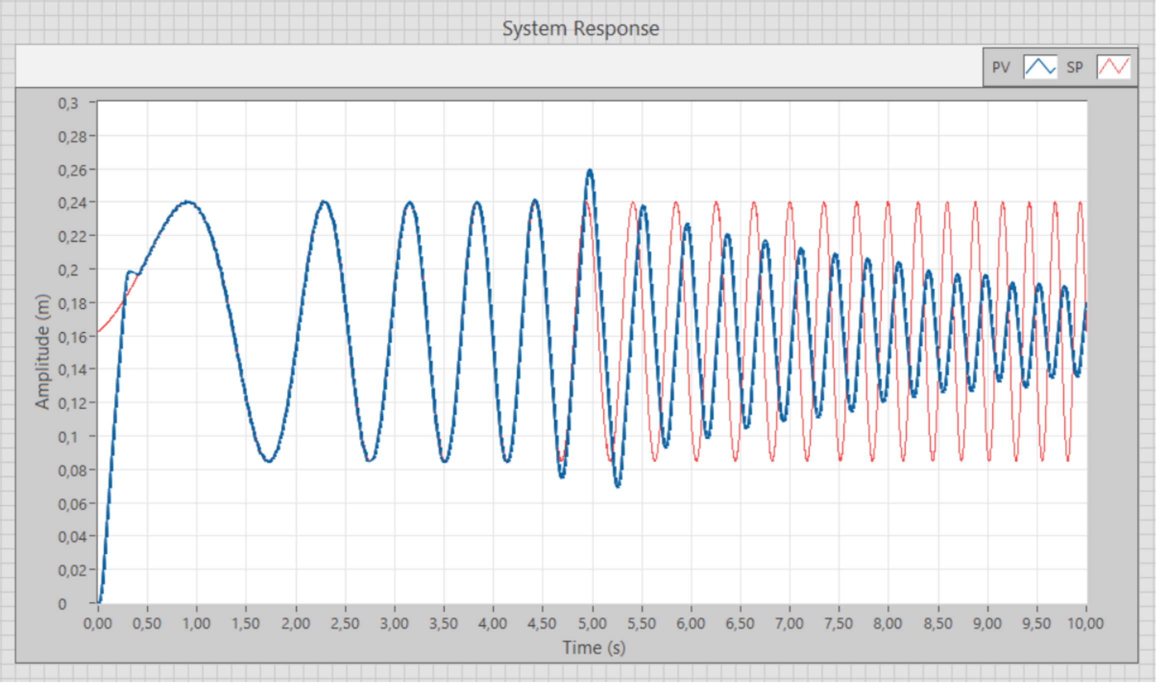
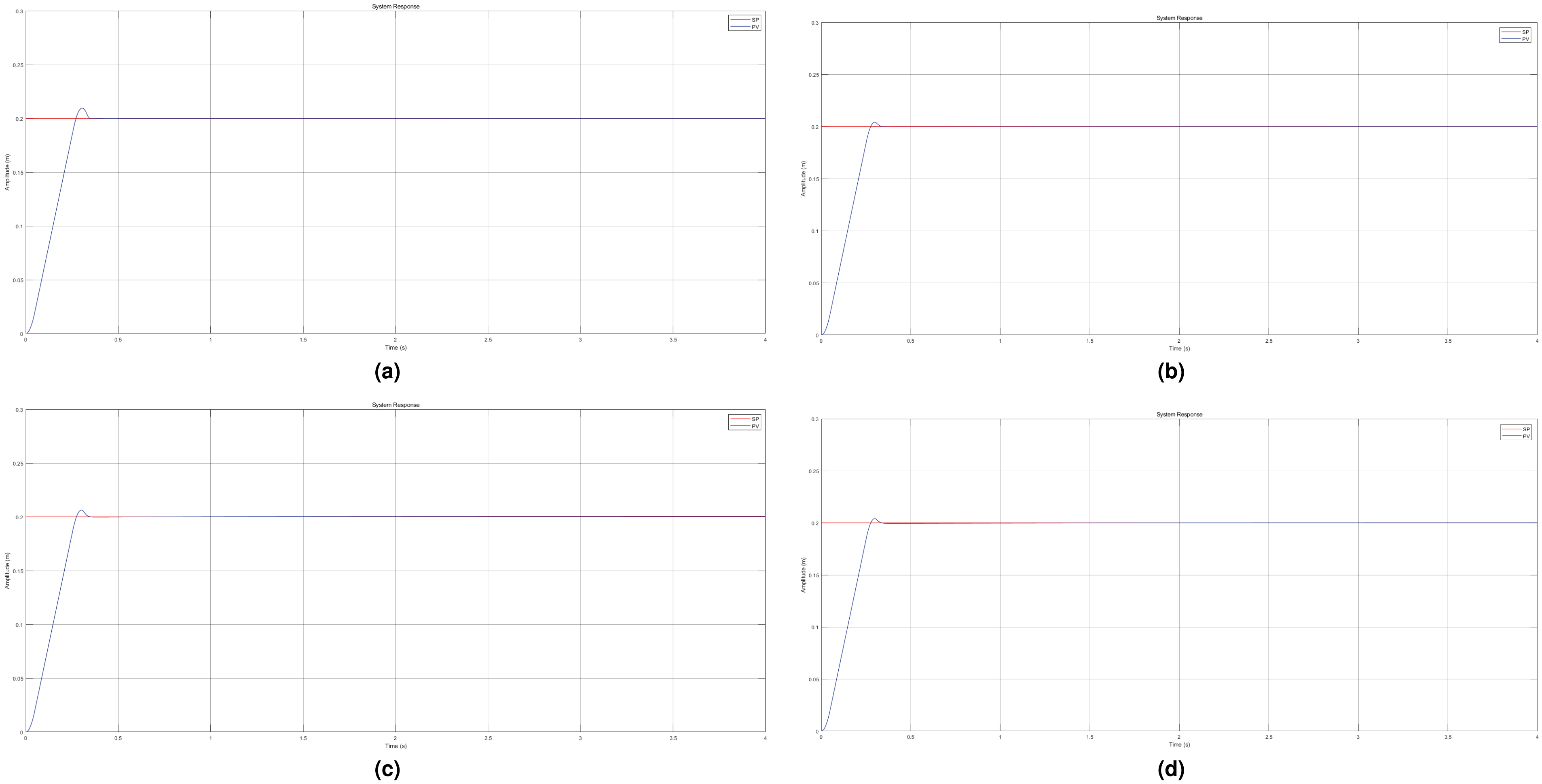


Figure A.1: Step response of Controller 2.

A.3 GA Step Response

In this appendix are shown the step responses of Controllers 3 to 12. This test was carried out according to the instructions in Section 3.4.3, and the findings were analyzed in Section 5.2.1



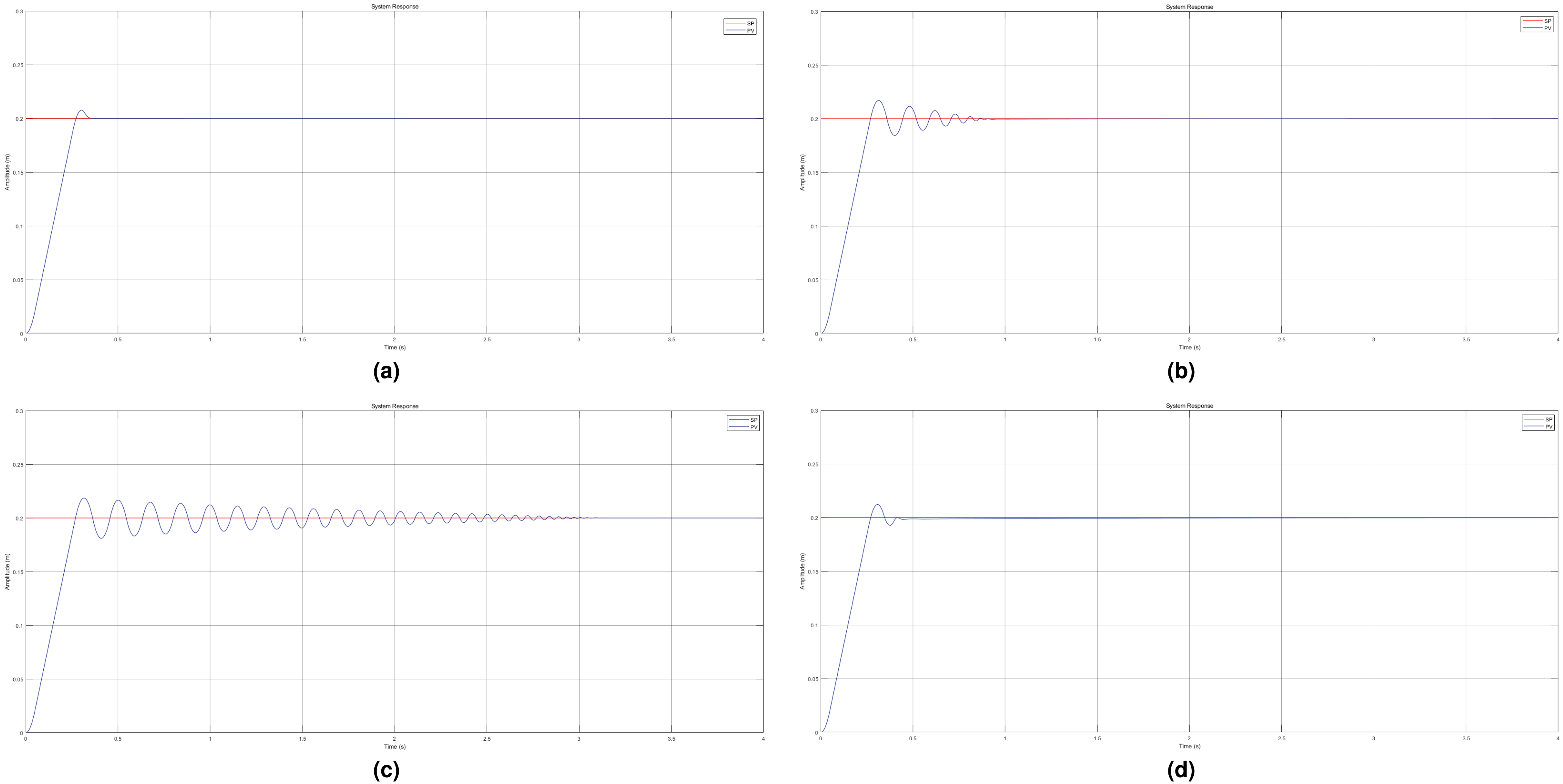
(a)

(b)

(c)

(d)

Figure A.2: (a) GA step response Controller 3; **(b)** GA step response Controller 4; **(c)** GA step response Controller 5; **(d)** GA step response Controller 6.



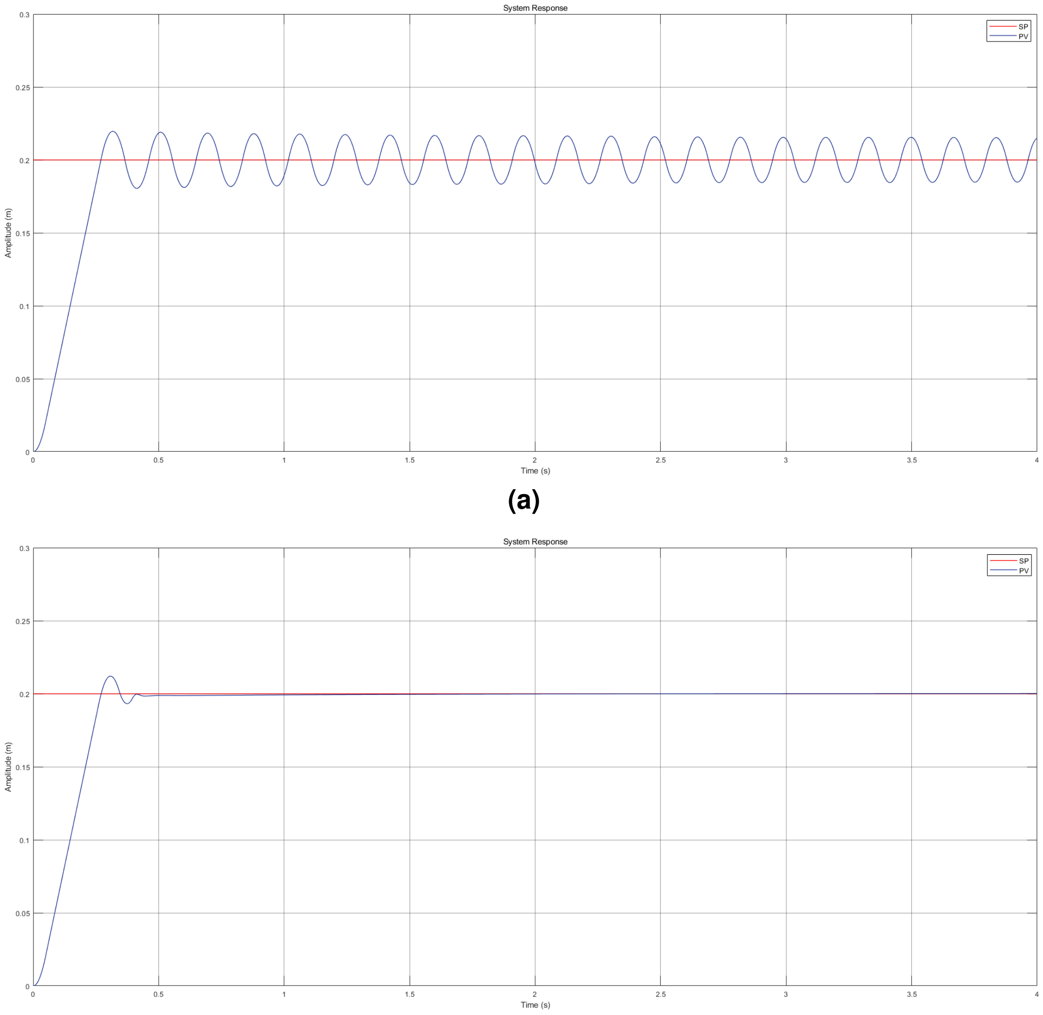
(a)

(b)

(c)

(d)

Figure A.3: (a) GA step response Controller 7; **(b)** GA step response Controller 8; **(c)** GA step response Controller 9; **(d)** GA step response Controller 10.



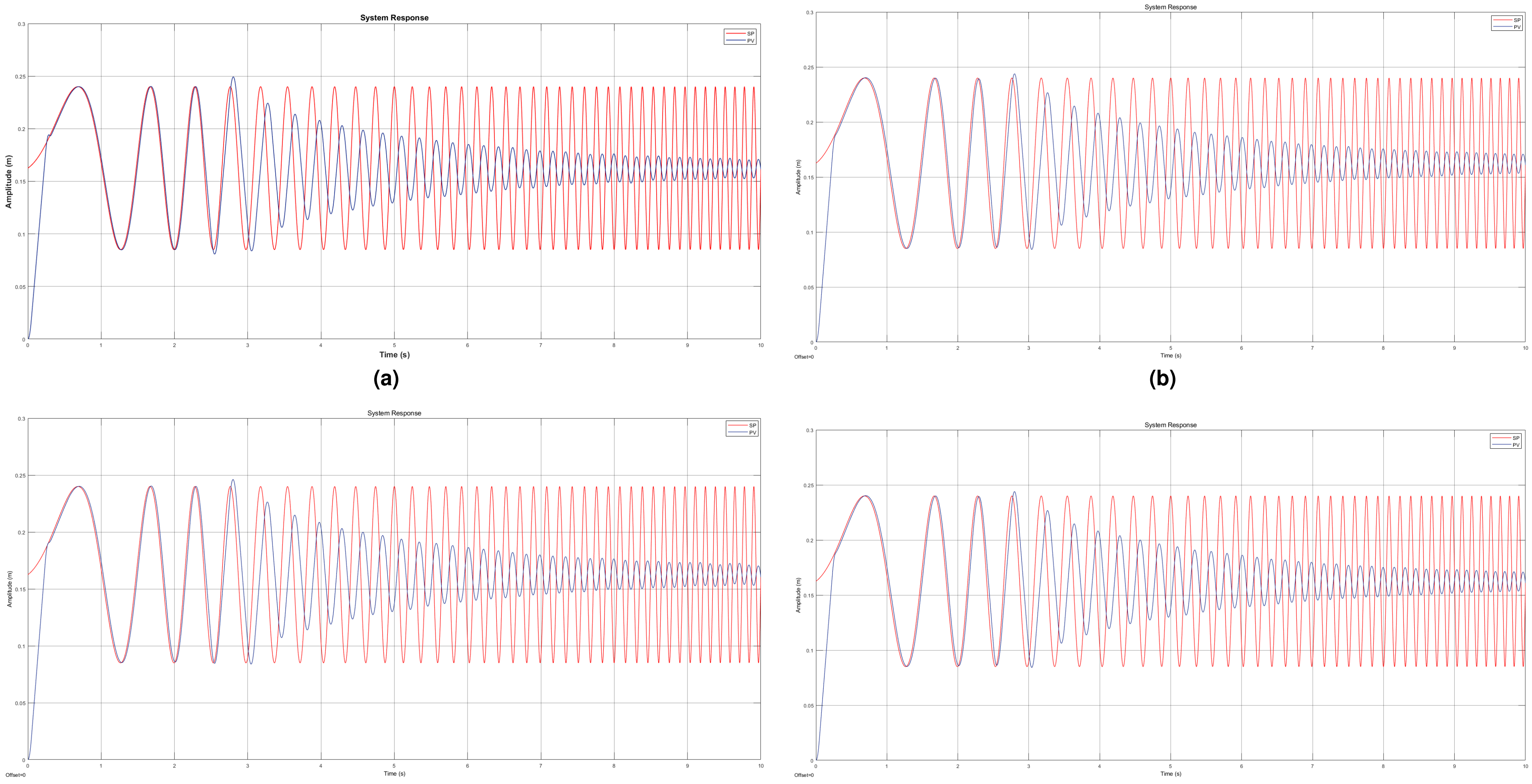
(a)

(b)

Figure A.4: (a) GA step response Controller 11; **(b)** GA step response Controller 12.

A.4 Frequency test: GA Chirp Signal

The results of Controllers 3 to 12 from the chirp signal test mentioned in Section 3.4.3 are shown in the following figures, and these results are further examined in Section 5.2.2.



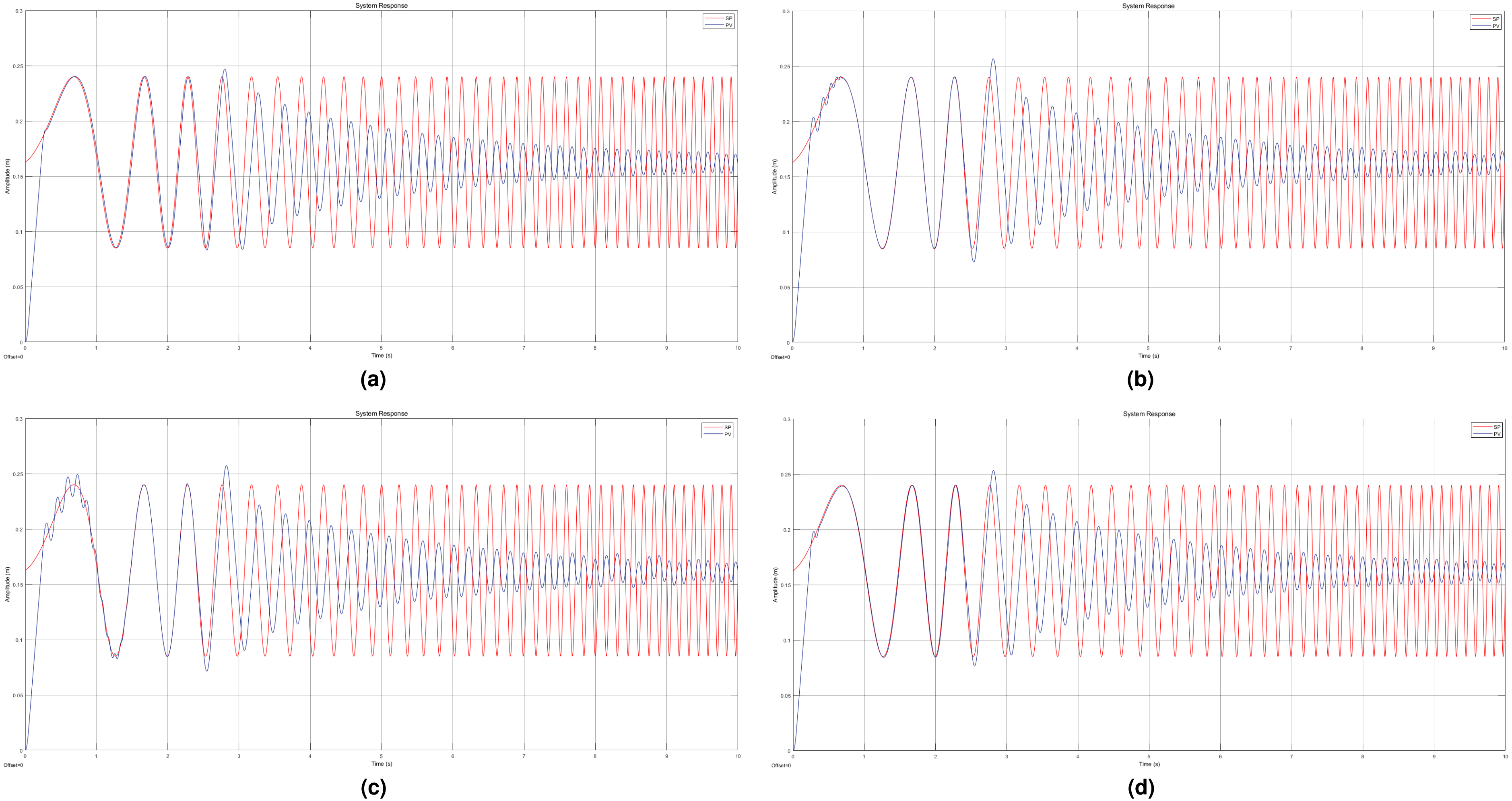
(a)

(b)

(c)

(d)

Figure A.5: (a) Chirp signal test performance of Controller 3; **(b)** Chirp signal test performance of Controller 4; **(c)** Chirp signal test performance of Controller 5; **(d)** Chirp signal test performance of Controller 6.



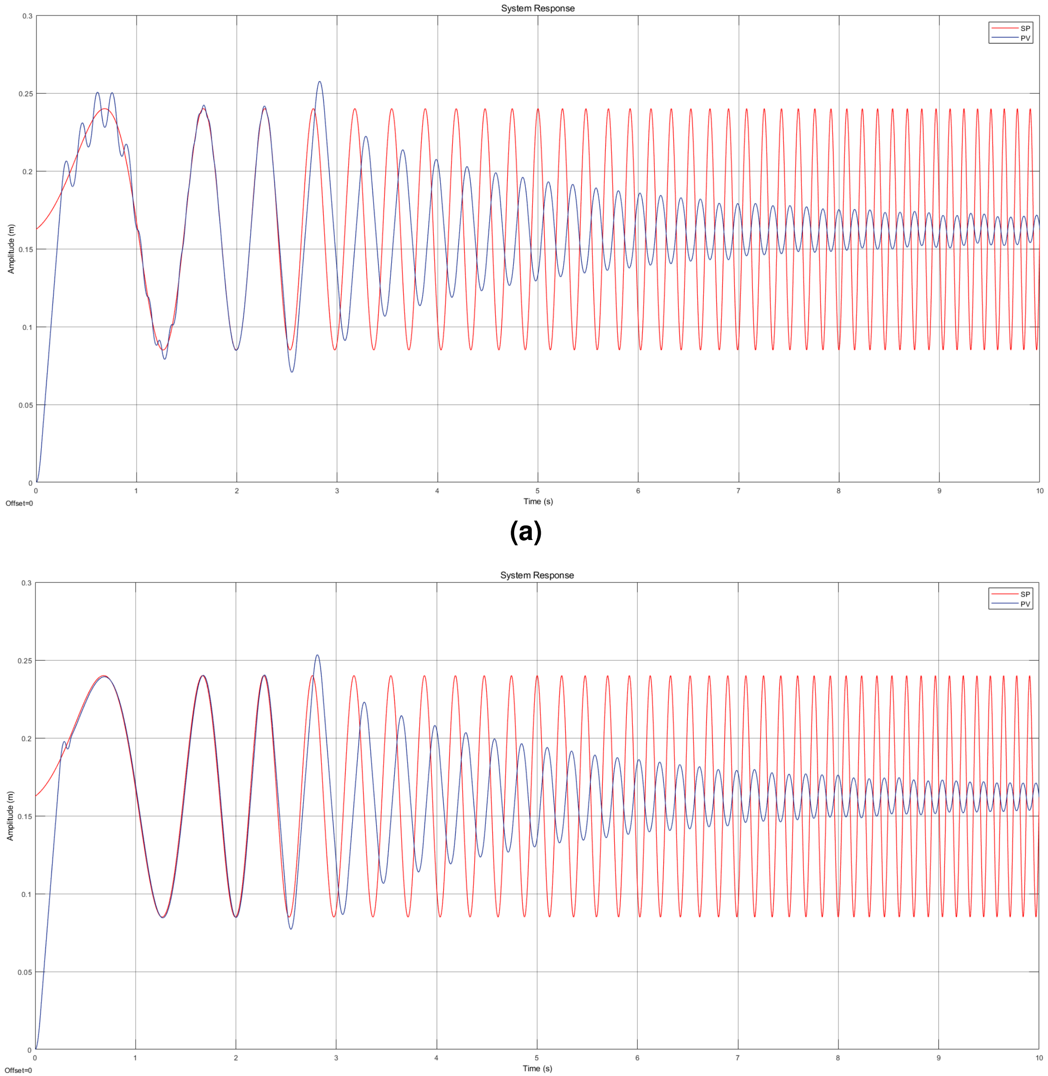
(a)

(b)

(c)

(d)

Figure A.6: **(a)** Chirp signal test performance of Controller 7; **(b)** Chirp signal test performance of Controller 8; **(c)** Chirp signal test performance of Controller 9; **(d)** Chirp signal test performance of Controller 10.



(a)

(b)

Figure A.7: (a) Chirp signal test performance of Controller 11; **(b)** Chirp signal test performance of Controller 12.

A.5 Frequency test: GA Manual Chirp test

The results of the manual chirp test for Controllers 3 to 12 are shown in the following Images; it was carried out as defined in Section 3.4.3, and the results are analyzed in Section 5.2.2.

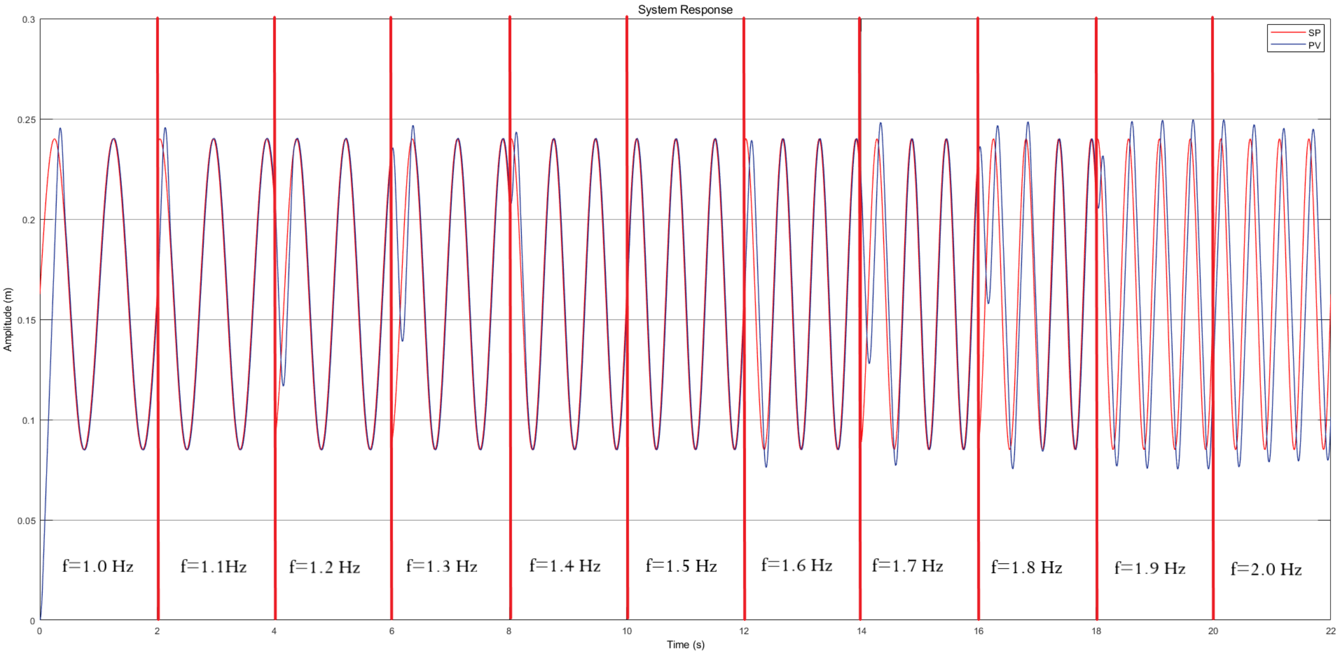


Figure A.8: Manual chirp signal test performance of Controller 3

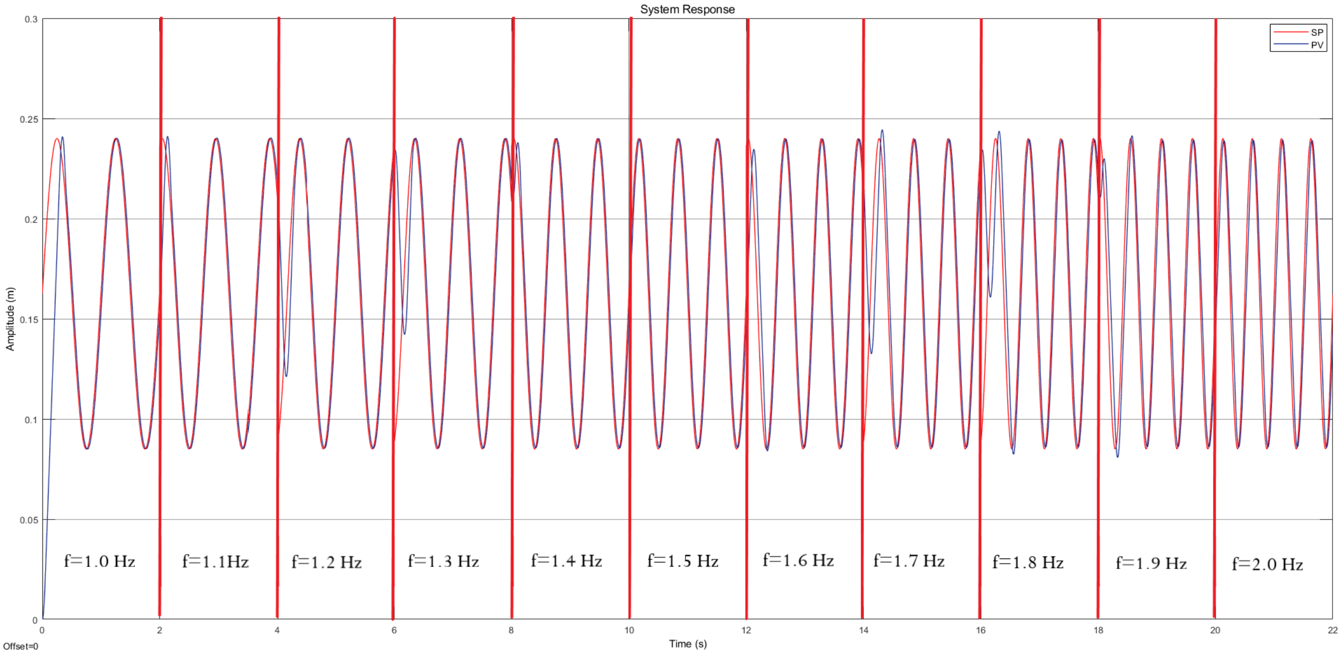


Figure A.9: Manual chirp signal test performance of Controller 4

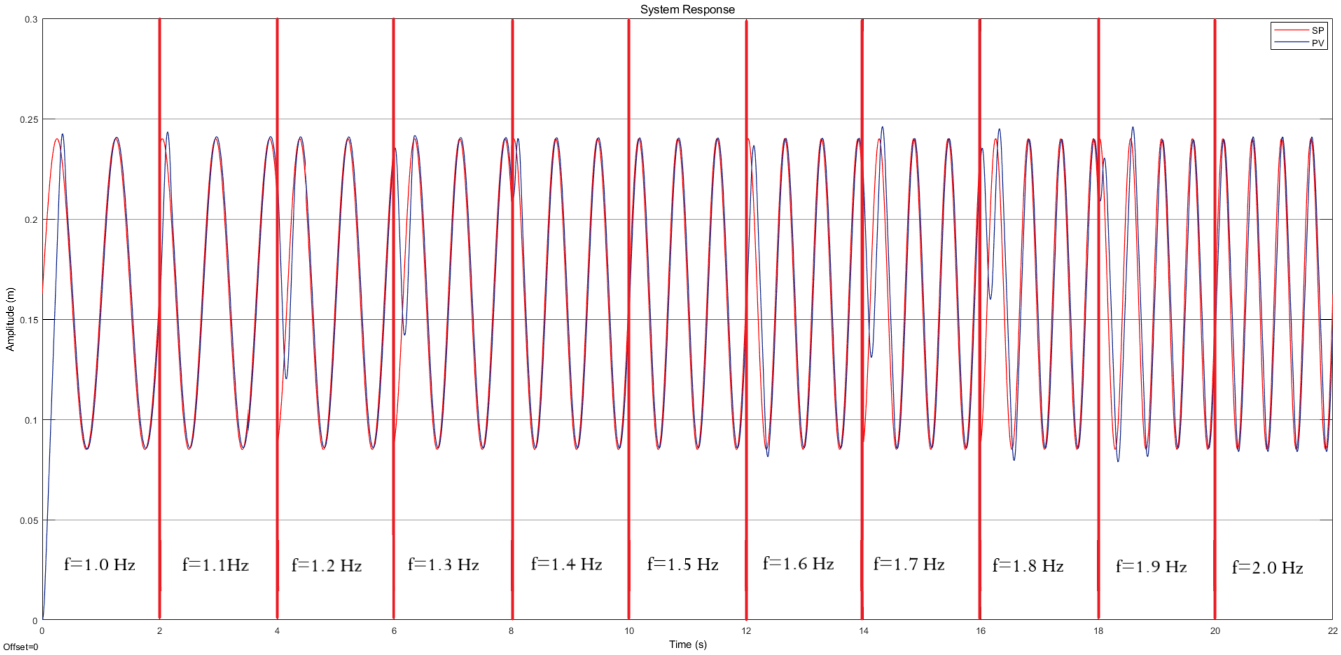


Figure A.10: Manual chirp signal test performance of Controller 5

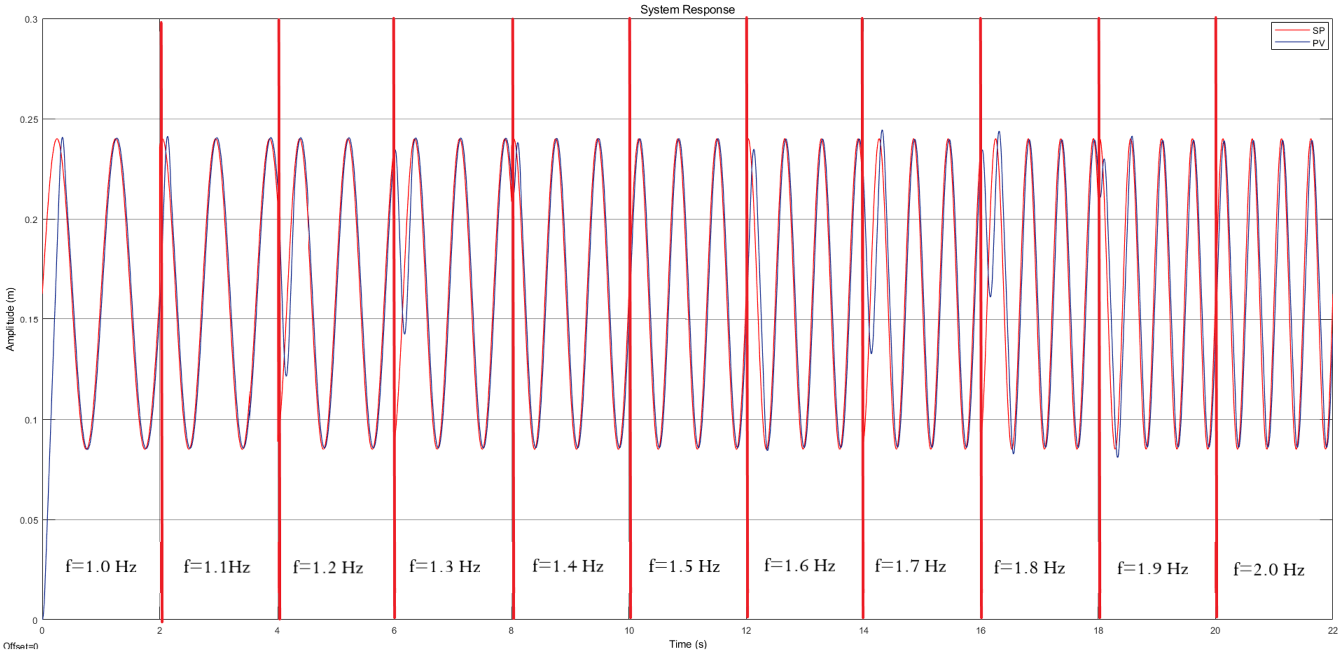


Figure A.11: Manual chirp signal test performance of Controller 6

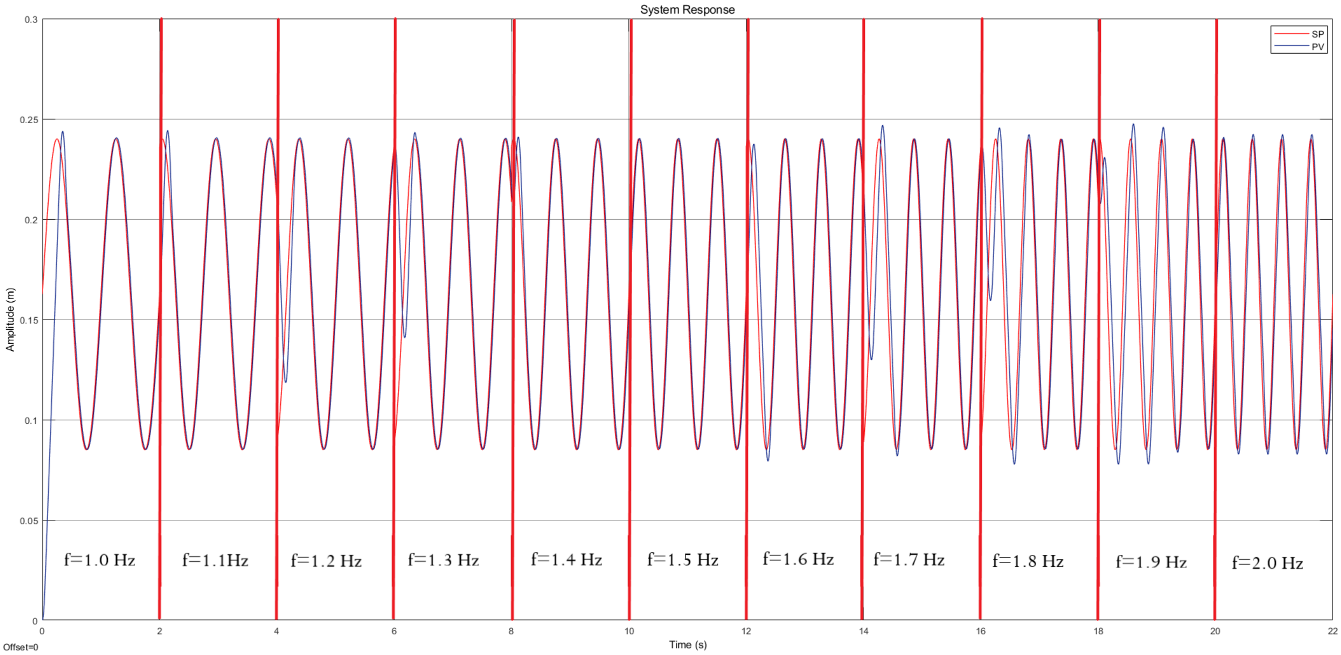


Figure A.12: Manual chirp signal test performance of Controller 7

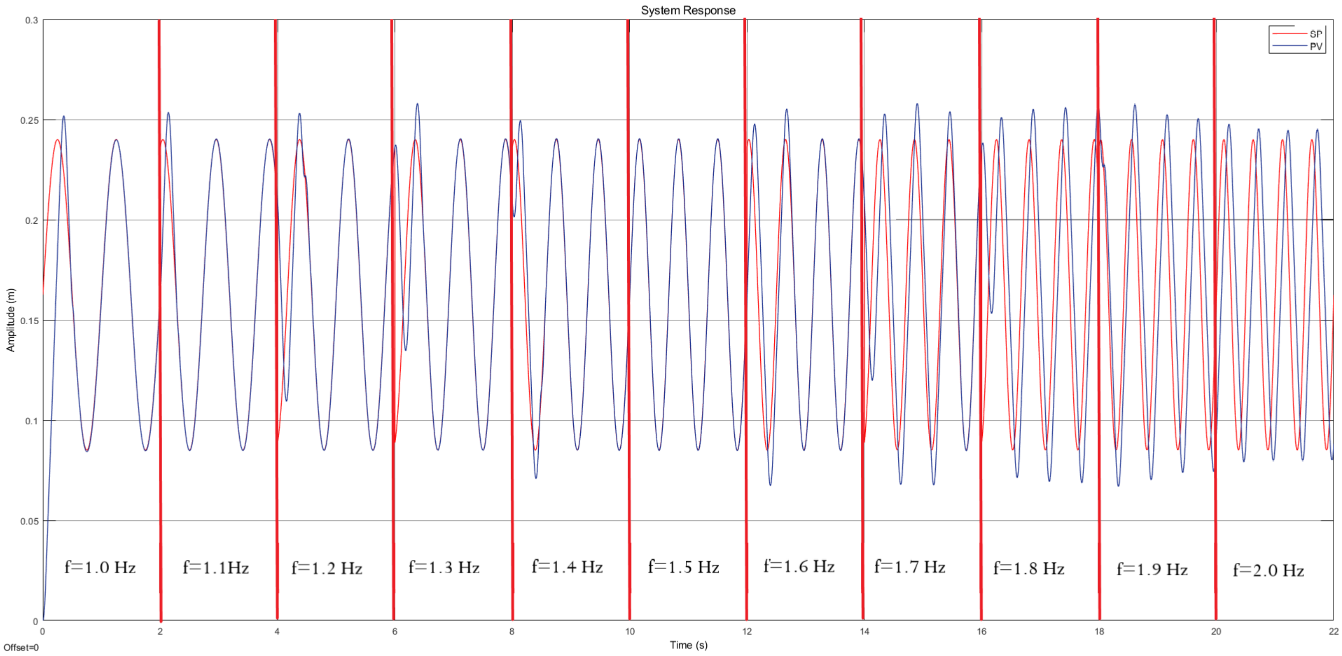


Figure A.13: Manual chirp signal test performance of Controller 8

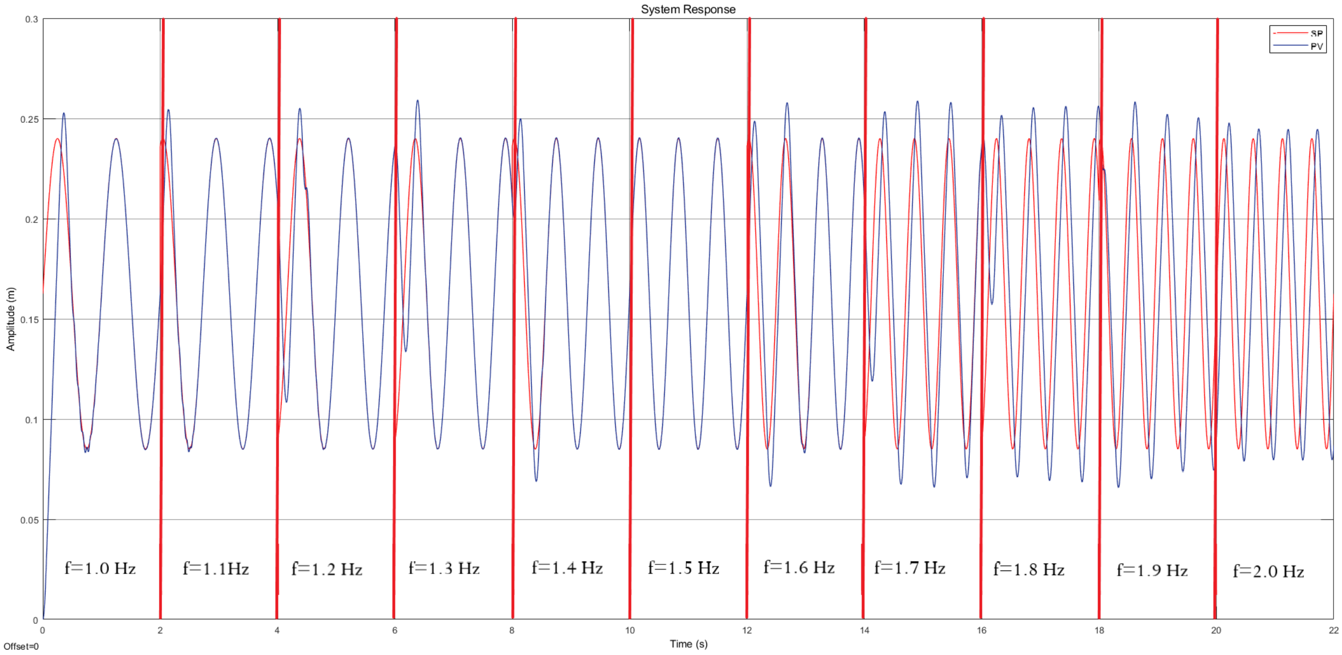


Figure A.14: Manual chirp signal test performance of Controller 9

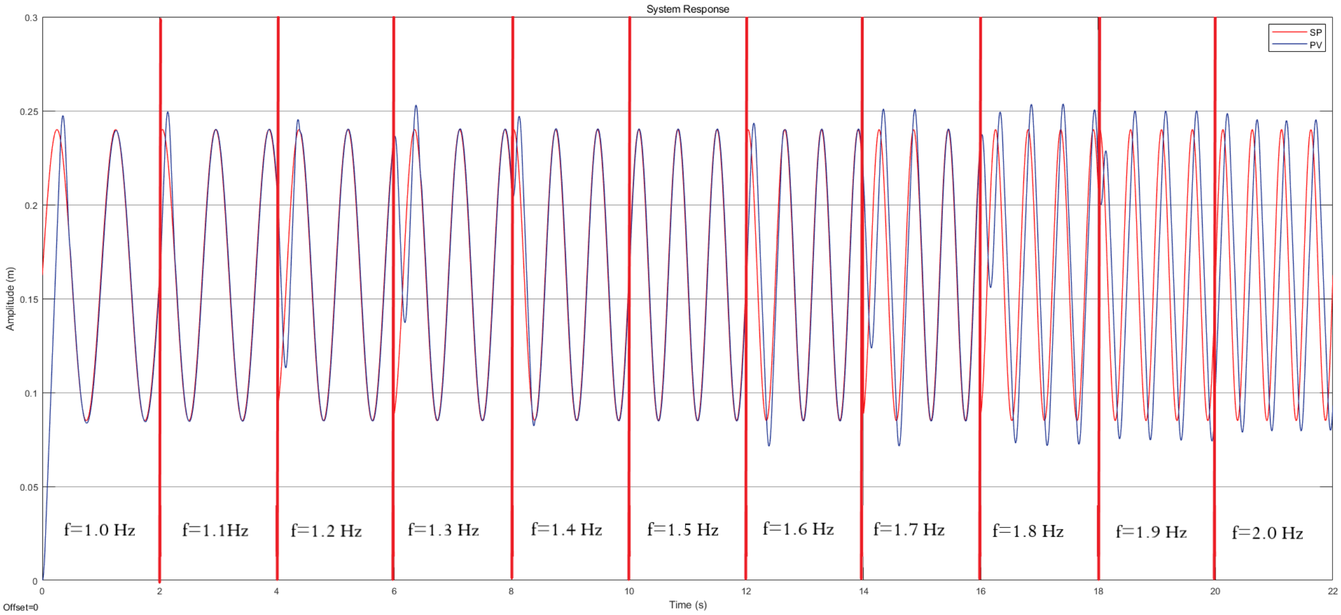


Figure A.15: Manual chirp signal test performance of Controller 10

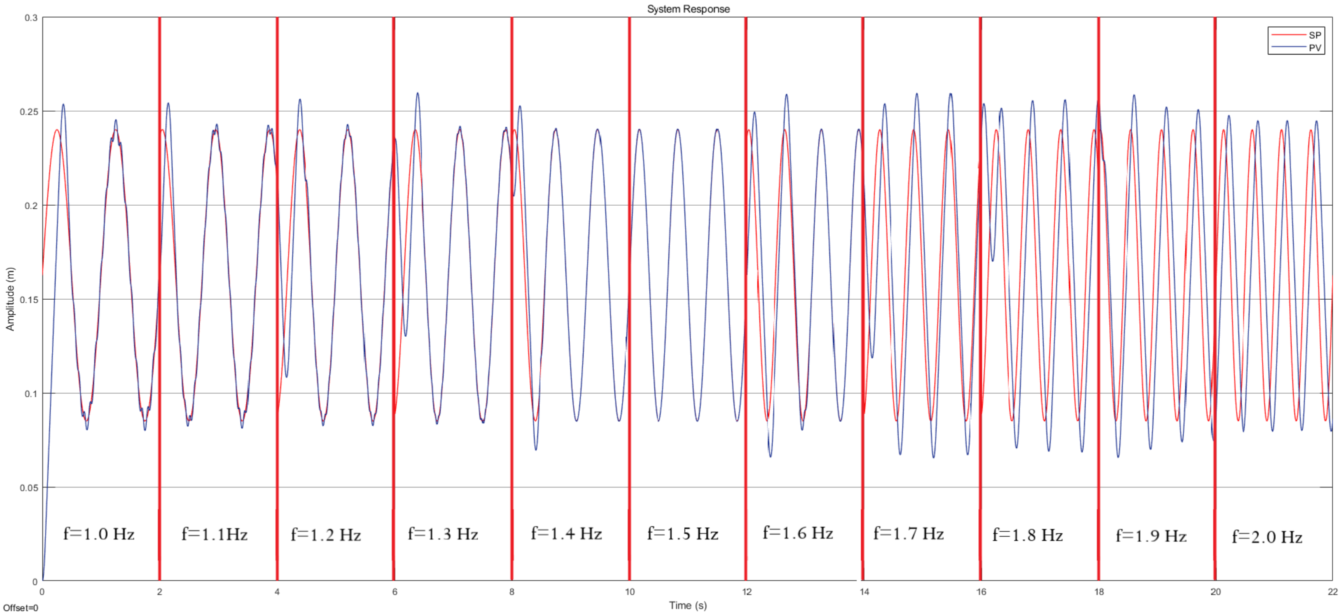


Figure A.16: Manual chirp signal test performance of Controller 11

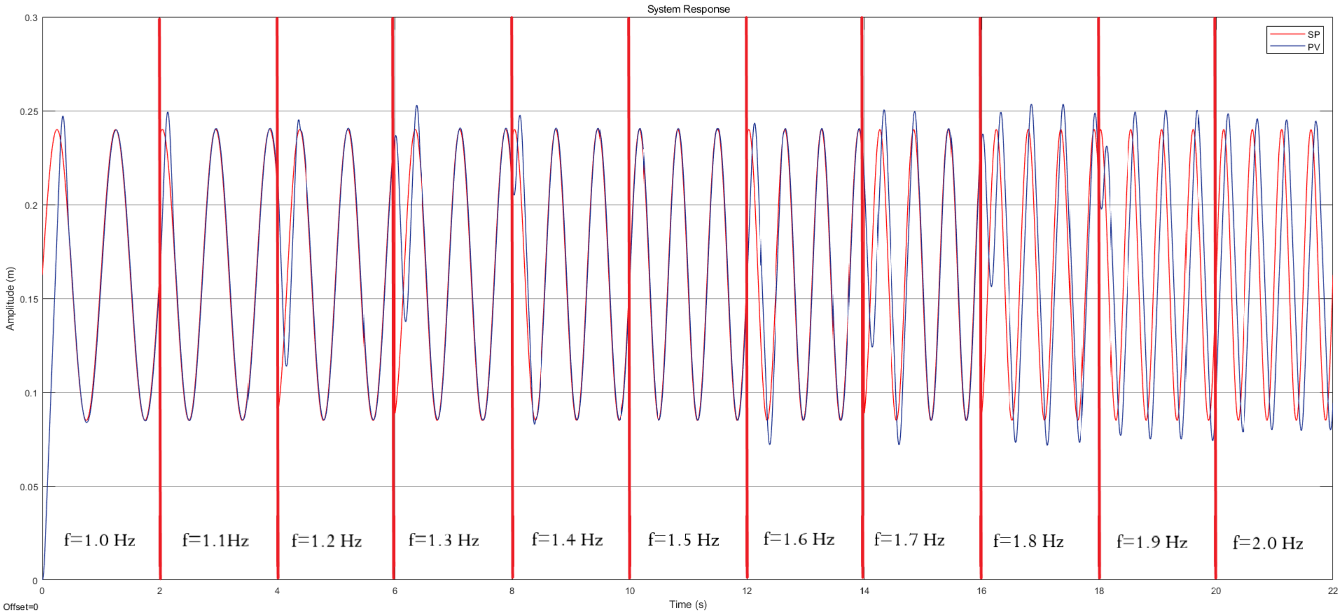


Figure A.17: Manual chirp signal test performance of Controller 12