Control and Development of a Robotic Cycling Trainer

Patrícia Valcarcel Lopes
Instituto Superior Técnico, Lisboa, Portugal

July 2021

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.

1. Introduction

Stroke is the second leading cause of death and a primary cause of disability worldwide with an increasing incidence due to the population aging. [1] 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.[1] In Europe, the reality is not different, stroke has affected 1.5 million people and generated 439,000 deaths.[2] This event has deep consequences in people's life 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.[3] 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.[4] In fact, post-stroke rehabilitation will soon increase, putting more pressure on healthcare budgets. [2] 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. [3] 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 . [3] Having this, it is expected that, for ethical reasons in addition to economic ones, it is imperative to increase rehabilitation options and effectiveness for stroke rehabilitation.[4]

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. [5] 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.[6] Considering these similarities with walking, cycling leg exercise can be seen as a solution and might be an alternative motor function rehabilitation method.[7] 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.[8] 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.[9]

Bearing in mind that stroke is a major problem with several people around the globe who consequently needs to be rehabilitated afterwards, it is understandable that there is a considerable need to develop novel rehabilitation techniques, in particular, to develop rehabilitation robots. The most mentioned advantage is that 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.[10] This helps the physiotherapist to concentrate on physical recovery during clinical therapy and to supervise multiple patients during treatments at the same time.[4]

Since 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 for gait rehabilitation.

This thesis aimed to further the development of the cycling trainer device. 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 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 the analysis of the system response with each controller and determine the best suited to the device.

Afterwards, a trajectory planning algorithm was required, which enabled 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 some of these trajectories was studied. Within the development of this project, it was also intended to develop a 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.

2. System Implementation and PID Control

2.1. The Haptic Cycling Trainer

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 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.

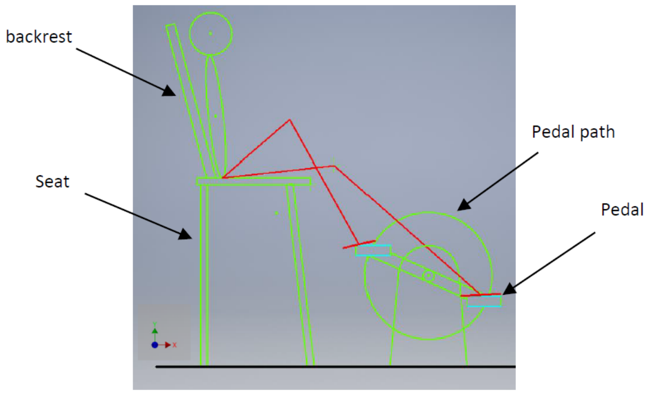

Figure 1: Schematic of the patient using the cycling ergometer.

In Figure 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 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 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.

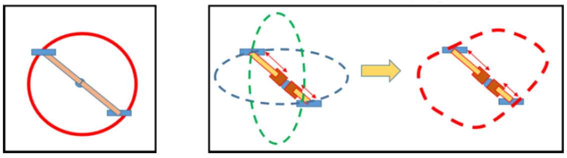

Figure 2: Representation of the key difference of the HaCT device and the current available cycle ergometer. On the left is represented the normal cycling functioning and in the right is represented how this system will operate.

The 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.

2.2. System 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. It was necessary 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.

2.3. 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. 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 1Hz and 2Hz.

PID Autotuning

The LabVIEW software already contains a palette of G-language building blocks known as Virtual Instruments (VI).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, to find the PID parameters it was defined a step signal of 0.2 m of amplitude, 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 1.

$$setpoint(t) = 0.1625 + 0.0775 \cdot sin(6.2832 \cdot t) \quad (1)$$

Tuning Based on The Genetic Algorithm

The 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.[11] The GA optimization was carried out in the MATLAB/Simulink environment, using the system's previous Simulink model in conjunction with an implemented script.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.[11] [12] 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. 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.[12] 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.[12] This process continues until the population converges to the global maximum or another stop criterion is reached.[13]

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.

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 2, Integral of Absolute Magnitude of the Error (IAE) as in Equation 3, Integral of the Square of the Error (ISE) present in Equation 4 , Integral of Time multiplied by Absolute Error (ITAE) in Equation 5 and Mean of the Square of the Error (MSE) in Equation 6.

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

$$IAE = \int_0^T |e(t)|\, dt \tag{3}$$

$$ISE = \int_0^T e(t)^2 dt \tag{4}$$

$$ITAE = \int_0^T t \cdot |e(t)|\, dt \tag{5}$$

$$MSE = \frac{1}{t} \int_0^T [e(t)]^2\, dt \tag{6}$$

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 created 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.

The second setpoint signal studied was a sine wave defined, once again, in the Simulink model used by the script with the fitness function. The setpoint for this set of simulations is characterized in Equation 7, with an angular frequency of approximately 1.5Hz which was ω=9.4248 rads/s.

$$setpoint(t) = 0.1625 + 0.0775 \cdot sin(9.4248 \cdot t) \tag{7}$$

2.4. Peformance tests

The controller we were trying to tune was 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 and the ones found using the GA optimization match our requirements. It was tested the Step-response characteristics, it was designed a frequency test and delay and RMSE calculations.

Step-response characteristics

The goal of this test was to study the rise time, settling time, overshoot, peak and peak time of each controller. 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 Autotunning 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

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.

To perform the first 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 1 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 (0.1 Hz), the target time (10 s) and frequency at target time (4.0 Hz).

With the second 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 reached 2 Hz.

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 RMSE it is done the squared root of the mean-squared error.

3. Pedal Trajectory and GUI

One of the most innovative features of this system is the therapist’s opportunity to freely design a pedal trajectory based on his experience that will contribute to efficient patient recovery. 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 3, 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 8.5 cm (represented by the smaller circle) and 24 cm (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.

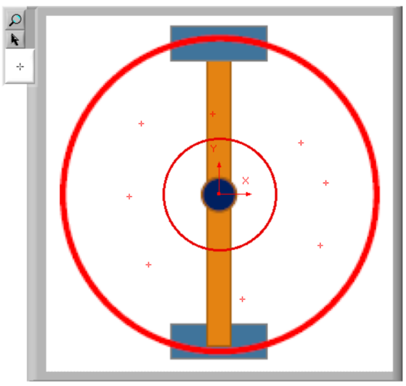

Figure 3: LabVIEW IMAQ display with representative image of pedal and its workspace.

3.1. 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.

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 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. 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.

In the vector manipulation stage, 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).

Is at this step, where the trajectory interpolation is performed, 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. To proceed with the trajectory interpolation it was used the LabVIEW block “Interpolate 1D VI” that performs 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.

If the user decides to use a predefined trajectory, three options are available: a circular shape, a butterfly-like shape, and the foot gait pattern.

3.2. Graphical User Interface

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.

The first tab is the Settings tab 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 and the tab to select the predefined trajectories.

The second tab has the Results in real-time. At the upper part of the tab, there are 4 numeric boxes representing the pedal velocity in RPM, the number of total cycles the pedal will perform within the simulation time, the total distance traveled by the patient (in meters) at the end of the simulation, and the current simulation time.

At the bottom, there are two graphical displays, the one on the left shows the pedal length versus time where can be seen the desired pedal length (calculated from the trajectory interpolation) and the system's response. On the right side, there is an XY graph displaying the interpolated trajectory in Cartesian coordinates, the data points selected by the user at the settings tab and, 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.

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

4. PID Control Results

The controllers discovered by the Autotuning Wizard and the GA optimization were tested as stated in Section 2.4, and the results are presented below.

4.1. LabVIEW Autotuning results

The Autotuning Wizard discovered two controllers: controller number 1 was found tuning with the step signal, and controller number 2 was discovered tuning with the sine wave given in Equation 1.

Step-response characteristics

Controller number 1 had the best results for overshoot (7.4%) and reached peak (0.2148 m), whereas controller number 2 had the best results for rise time (0.2020 s), settling time (0.4165 s), and peak time (0.3310 s).

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 obtained, controller 2 was the best match between the two controllers. This controller had a faster rise time and a shorter settling time. The overshoot of 9.35%, hitting a peak of 0.2187 m, was one of this controller's drawbacks. However, after settling, both controllers displayed oscillatory behavior, which is undesirable.

Frequency Test

In the **chirp signal** 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.

In the second test with the **frequency changed manually**, it was noticeable that, for frequencies in the range of 1.0 to 1.4 Hz, the model took a few milliseconds to converge to the SP signal, about 0.40 s, corresponding to the settling time found in the step signal test. The signal converged completely after these first milliseconds, maintaining the satisfactory behavior until the next frequency was set at the setpoint (SP) signal. Due to the similarity of both the System's and SP signal at the time of the frequency shift, the signal converged much quicker at 1.5 Hz; although, when adjusting the SP frequency in the other parts of the test, there was a spike in the SP signal, which took the system some time to settle again. Between 1.6 and 2 Hz, the device appeared to have difficulty following the test signal. First, due to a lack of time to settle, it was unable to fully converge to the SP signal, resulting in major overshoot at the sine wave peaks; then, as the frequency raised, it took longer to settle and started to have a phase displacement; and finally, as the frequency continued to increase, it took longer to settle and began to have a phase displacement. At the time the SP signal reached the 2Hz frequency, the system response (PV) was not capable of converging within 2 s.

Delay and RMSE calculation

Controller 1 had a 31 ms delay, while controller 2 had a 30 ms delay. The RMSE value for Controller 1 was 3.52$\cdot 10^{-2}$ whereas Controller 2 was 3.50$\cdot 10^{-2}$.

Based on these findings, it was possible to conclude that controller 2 had a better match to the sine wave, as shown by a lower delay value and a lower root mean squared error. Despite the fact that it had better performance in this test, the difference between this controller and the first is minor. The time delays of both controllers were in the range of 30 ms. Assuming a therapy session at 1Hz, 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 was considered to be inadequate.

4.2. Genetic Algorithm results

The GA optimization in MATLAB was used to find controllers 3 to 12 with two different 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 7.

Step-response Characteristics

All of the controllers had the same rise time of 0.9161 s, with controller 6 settling the fastest 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 (2.0296%) , while controller 11 had the most (9.8118%). Controller 6 was the best controller in terms of reached peak (0.2041 m) and peak time (0.2980 s), while controller 11 was the worst once more (0.2196 m and 0.3170 s, respectively). 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.

Frequency Test

In the **chirp signal** test, the controllers had a limit frequency quite close (ranging from 1.03 Hz to 1.17 Hz), but it is important to note that their answer in the first few seconds of simulation differed significantly. Controllers 3-7 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 s to converge and ringing, while controllers 9 and 11 experienced this behavior for around 1.5 s. 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.

In the second test, with the **frequency changed manually**, Controller 3 effectively followed the sine wave up to a frequency of 1.8 Hz, after which the device 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 1.8Hz signal. A closer look at the responses of controllers 8-12 revealed a different behavior. Controllers 8 and 10 had some difficulty settling to a 1.6 Hz wave, taking about 1 s to converge, then at 1.7 Hz it took 1.75 s 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 s to converge and 2 s at 1.7 Hz, during which it displayed the same behavior as the others, showing phase displacement and overshoot. Controller 11 required 1 s 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 s to settle, but it was no longer able to converge, exhibiting phase displacement and overshoot at the wave peaks.

Delay and RMSE calculation

There was 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 ms from the setpoint and the latter having delays ranging from 27 to 32 ms. Even though the second group of controllers outperformed the first in terms of delay analysis, it is essential to note that even a 20 ms delay with the setpoint is significant.

Regarding the RMSE results, the second group of controllers (controllers 8-12) had 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.

4.3. PID results discussion

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 s), 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.

The results in this section showed that the controllers found were 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 was the best way to choose the best option.

5. Trajectory Results

It was essential to further test the controllers' performance with the foot gait trajectory to choose the fittest controller for the system. After the testing, it was concluded that controller 6 was the best option for the model. Then it was necessary to study the system response with different trajectories at different frequencies.

5.1. Circular Trajectory

The circular trajectory was defined by selecting 9 points in the delimited pedal range as seen in Figure 4. For this test it was selected two frequencies: 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.

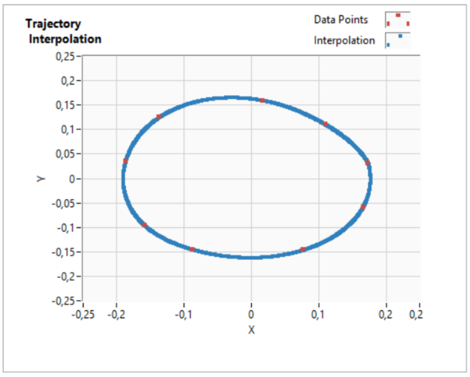

Figure 4: XY graph containing the data points selected to define the trajectory in blue and the interpolated trajectory in red.

Frequency of 0.5 Hz

In Figure 5 it is represented the system's response to the interpolated trajectory. Even though the simulation time was defined for 10 s, for simplicity it is represented in the graph the first 4 s 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.

The settling time represents 10% of the first cycle which is a satisfactory result, furthermore, the system seems to react adequately to differences in the pedal length. The error between the SP and

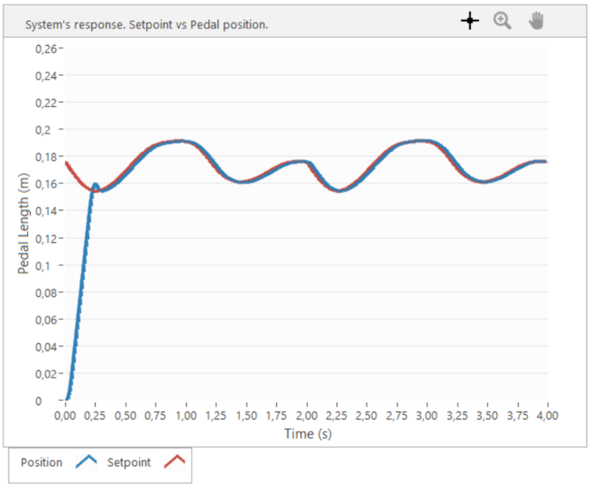

Figure 5: Graph of pedal length (meters) versus simulation time (seconds) at 0.5Hz. The setpoint is shown in red, while the system's response is shown in blue.

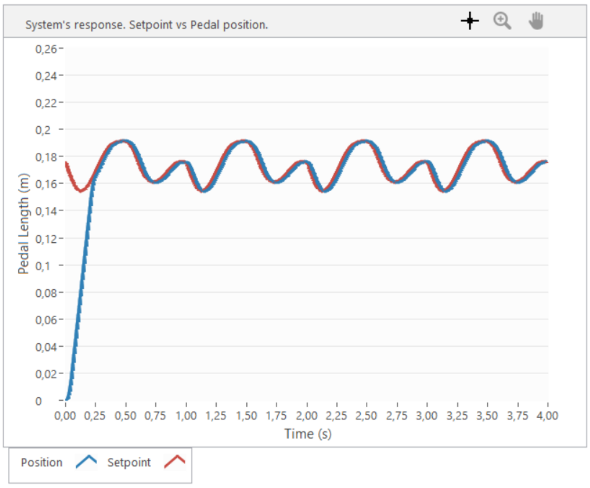

Figure 6: Pedal length (meters) versus simulation time (seconds) graph at 1Hz. In red is represented the setpoint (the defined trajectory) and in blue the system's response.

the PV is 1.60$\cdot 10^{-3}$, which is the smallest error discovered thus far. The lowest delay calculated between the two signals is zero, indicating an ideal response. Regarding the correlogram, 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.

Frequency of 1 Hz

Figure 6 depicts 4 cycles of the simulation. The first thing to notice is that the system takes 0.25 s 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, there is still a minor delay between the two signals. 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. There was a correlation of 1 at 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.

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

5.2. Foot Gait Trajectory

The foot gait trajectory was adapted from [14], 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.

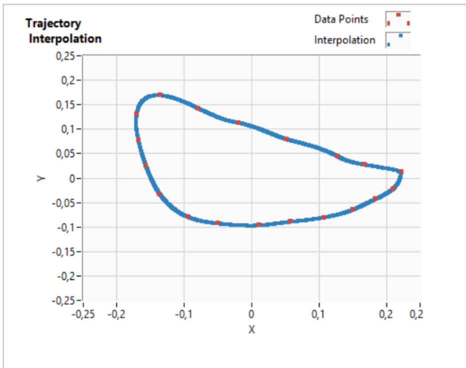

Figure 7: XY graph containing the data points selected to define the trajectory in blue and the interpolated trajectory in red.

First it was checked the results at 0.5 Hz, a adequate velocity for gait therapy, and then it was tested for 1 Hz, the system's maximum velocity.

Frequency of 0.5 Hz

In Figure 8, it is represented about two trajectory cycles of this system with the foot gait trajectory. Closer inspection of this figure shows that the system needs 0.30 s to settle, which is 0.15% of the first cycle that finishes at 2 s. 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 system presented a RMSE of 2.66$\cdot 10^{-2}$ and the minimum delay found between the SP and the PV is 1.50$\cdot 10^{-1}$ s. The correlogram obtained, goes accordingly with the quality of the rest of the results. It can be seen that 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-

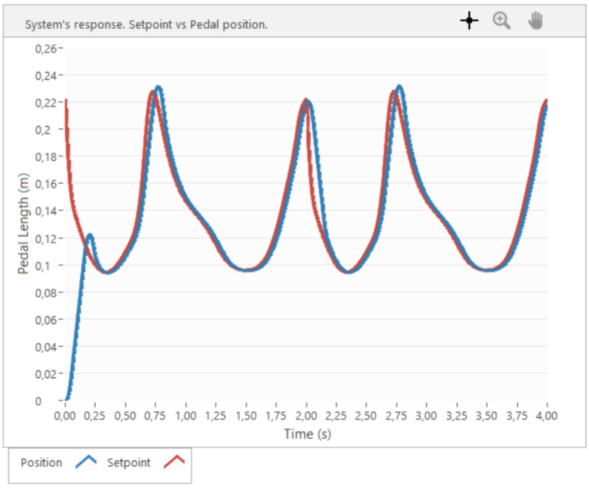

Figure 8: System response with controller 6 to the foot gait trajectory.

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.

Frequency of 1 Hz

Figure 9 presents the system response to the trajectory at 1Hz. The pedal length reaches the desired value in about 0.25 s of simulation, taking 25% of the first cycle. What is noteworthy about Figure 9 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.

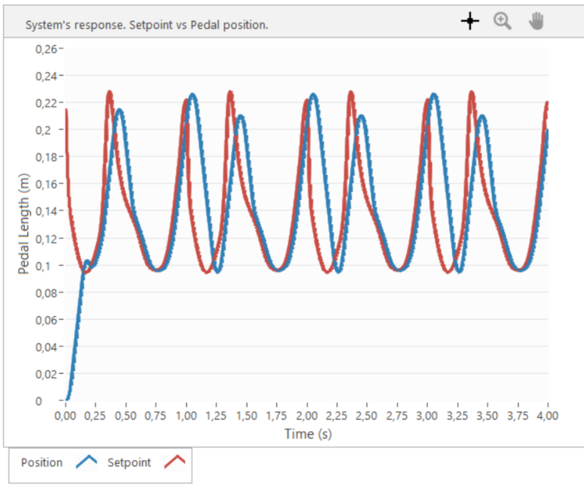

Figure 9: Pedal length (meters) versus simulation time (seconds) graph at 1Hz, with the setpoint and the system response represented.

The minimum delay found was of 3.90 s 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 with a lower frequency. The correlogram demonstrated, once again, the low quality of the system reaction to the trajectory. The symmetrically 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 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 shows that the best controller chosen, from the set of controllers discovered using diverse approaches, does not have a satisfying behaviour for more complex trajectories.

6. Conclusions

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 and tuned using the GA optimization.

The goal of creating a trajectory algorithm was accomplished as it was developed an algorithm to collect the data points selected from the user to create a trajectory for the pedal to follow. It was implemented a set of predefined trajectories, including one that mimics the foot gait trajectory. A clean and intuitive Graphical User Interface was designed 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.

The system reaction to two different trajectories, a circular and a foot gait trajectory, was tested. The system response to the circular trajectory was ideal, with the pedal being able to successfully follow the trajectory without delays and at 0.5Hz 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.

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. 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.

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. 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 the appropriate controller and its tuning and on new proposals regarding the trajectory planning algorithm and the GUI.

7. Acknowledgments

This document was prepared and made public as part of the author's MSc thesis in Biomedical Engineering at Instituto Superior Técnico as part of an institutional academic requirement. 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.

References

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

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

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

[12] N. Goldstein, "Process control," *BioCycle*, vol. 49, no. 8, pp. 1–6, 2008.

[13] 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.

[14] 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.