UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering,

Mathematics & Computer Science

Validation of a machine learning approach and control strategy for a rehabilitation robot to train the upper extremity in stroke patients.

**Alvaro Bustillo
Rodriguez**

M.Sc. Thesis

November 2018

Supervisors:
dr. ir. B.J van Beijnum
dr. ir. M. Wessels
prof. dr. ir. H. Hermens
dr. ir. M.Abayazid

Biomedical Signals and Systems Group
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente

P.O. Box 217

7500 AE Enschede

The Netherlands

Preface

This dissertation marks the end of my student life, which has led me to live in six different countries and to study in four different universities. This phase of my life has been exciting and uncertain, full of experiences, new languages and friends all over the world. However – and as much as I enjoy being a student-, I look forward to the next stage.

All in all, I enjoyed my academic life at the University of Twente. Regarding my master thesis, I would like to thank Bert-Jan for accepting to be my daily supervisor when times were uncertain, as well as for his great supervision. I would also like to thank Martijn for his valuable input and support during the experimental procedures. Similarly, I thank Momen Abayazid and Hermie Hermens for taking part in the supervising committee.

I could not have enjoyed my student life to its full extent if it weren't for all the friends I made throughout it. I am grateful to have met you all and I cherish our experiences together. In special, I would like to give a big shutout to my friends back in Spain, who motivate me despite the distance and to my friends here in Enschede, who made the city feel like home.

The biggest acknowledgment however, I reserve for my family. I could not have gotten to where I am in life without their support and I cannot express enough gratitude to what they have offered me. Thank you.

Hope you enjoy the reading,

Álvaro

Summary

Stroke is a disease that affects millions of people worldwide and which can result in long-lasting motor impairments. The resulting disabilities affect the performance of stroke patients when executing Activities of Daily Life (ADL). Post-treatment of the disease includes rehabilitation exercises, which are goal-oriented repetitive tasks aimed at restoring motor function of the affected body part. The field of rehabilitation robotics presents novel technology for the delivery of these exercises so as to reduce the workload of the clinician and in order to increase the amount of tasks per session.

With that in mind, the eNHANCE device was designed to be used in upper extremity rehabilitation and assistance through reaching tasks. The goal of this thesis is to add functionality to the robotic arm of the eNHANCE device, so that assistance-as-needed is given during training of the upper extremity. In order to do so, two main concepts were addressed: the behavior of the robotic arm and the adjustment of the support level.

The behavior of the robotic arm, on the one hand, concerns the assistance given by the robot, so that it resembles healthy performance in reaching tasks. To do so, a machine learning approach was evaluated to obtain a predicted healthy reaching time which would dictate the behavior of the robot. An experiment investigating different machine learning models and the use of different training dataset – Experiment I- was carried out so as to determine the validity of the machine learning approach in terms of prediction accuracy.

The adjustment of the support level, on the other hand, is related to the motivational functionality of the device. In such a way, assistance-as-needed will increase user engagement and favor motor training. In order to address the adjustment of the support level, a support level controller was postulated. Later, an experimental set-up –Experiment II- observed the behavior of the controller for three different simulated scenarios: when the participants acted normally, fatigued or was lazy. In addition, user perception of the change in support level was documented.

The conclusions from Experiment I led to the decision of choosing a Random Forest as a good candidate model. Furthermore, the features and tasks for the training dataset were specified, with a Base-to-Target task being favored. The final conclusion was that the Machine Learning approach is valid for limits of accuracy of less than 0.25 seconds. The conclusions from Experiment II prove that the proposed support level controller can adjust the support level depending on user contribution in the setting of the eNHANCE device. Furthermore, mean user perception was 50.8% accurate in determining support level change.

The end result of the work presented in this thesis is a control strategy that combines the results from the robot behavior and the adjustment of support level. From the combined action of the machine learning model and the support level controller, assistance-as-needed is thus delivered in an upper extremity rehabilitation setting. To conclude, future lines of work addressing the limitations of this study were proposed. These included the full implementation of the control strategy, its integration with a motivational platform and the evaluation of the control strategy in an experiment involving stroke patients.

List of Contents

List of Figures

List of Acronyms

ADL	Activities of Daily Life
FMA	Fugl–Meyer Assessment
EMG	Electromyography
ML	Machine Learning
BtT	Base-to-Target
TtT	Target-to-Target
ST	Single Tree
RF	Random Forest
ERF	Extremely Randomized Forest
ADA	AdaBoost
RMSE	Root Mean Square Error
MAE	Mean Absolute Error
ROS	Robot Operating System

Chapter I – Introduction

The introduction chapter will lead the way to understanding the research domain of the work presented in this master thesis. In such a way, the aim of this chapter is to clarify to the reader the reasons behind the topic of the thesis as well as the goals (Section 1.1), the problem definition (Section 1.2) and research questions (Section 1.3) to be addressed. Once these introductory bases have been laid out, a report organization (Section 1.4) will be presented as a gateway to connect all subsequent content of the master thesis.

1.1 Motivation & Goal

The advancements in assistive technology make possible its application into an increasing number of health-related conditions. Within the field of assistive technology, robotic support systems offer a useful platform for people with motor and neural diseases, so that they are able to perform activities otherwise difficult for them.

It was within the assistive technology framework that the eNHANCE project came to be. The eNHANCE project is an European funded project with multiple partners which aims at both assisting patients with motor impairment on their daily life routines as well as innovating in the field of assistive technology.

The work presented in this thesis is part of the eNHANCE project, and it is motivated by the development of the eNHANCE device, more specifically, the development of a robotic support system that will help in assistance to Duchenne patients and in rehabilitation of upper extremity motor function to stroke patients. The rehabilitation functionality will combine a motivational platform and the support system so as to promote user participation and contribution for the rehabilitation environment. It is of special importance to underline that the focus throughout this report will fall on the rehabilitation functionality of the eNHANCE device, aimed at stroke patients.

The design of the aforementioned robotic support system entails close interaction with the user. This means that the system will have to adapt itself to the necessities of the user, defined in terms of their performance for reaching tasks. In order to do so, the system will have to make use of the available data to predict user performance so as to control the level of support.

In such a way and in addition to the motivations behind the eNHANCE project, the goal of this thesis is to add functionality to the robotic support system by designing and validating prediction and control strategies which will ensure that appropriate assistance and rehabilitation are given to the user.

1.2 Problem Definition

In order to properly address the goal of the thesis and to accurately state the research question, one must first pinpoint the problem, or problems, to be tackled. For that purpose, it is best to explain the problem in the context of a high-level system architecture, which defines the main building units of the envisioned system as well as the relevant system parameters.

The scope of research refers to the parts of the system that will be investigated during the work in this thesis. Within the scope of research, five distinct units are defined: the user unit, the robotic arm unit, the machine learning model unit, the support level controller unit and the other device modules unit. All of these units –with the exception of the user unit- represent their respective parts within the system, where the other device modules unit references the additional modules of the device (see Section 2.3.1).

Furthermore, certain interactions between the system units are represented by the system parameters. These parameters can be divided in those concerning robot behavior and those concerning support level. The position information and the reaching time prediction are parameters regarding the behavior of the robot, while the user-robot performance, the user contribution, reaching time prediction and support level are those parameters related to the support level. The exact interaction of the system units through the system parameters will become clearer later in the report.

A motivational platform within the system architecture will be separately designed in addition to the units within the scope of research. Even though the work on this thesis does not cover the development of one such platform, the end product of the thesis should take into account possible communication with the platform. Hence, the motivational platform will be kept in mind when making design choices throughout the research.

With the above in mind, Figure 1 shows the conceptual model of the system to be designed. The interaction between the user unit and the robotic arm unit is represented in Figure 1 as a dashed line and represents the combined action of the user and the robotic arm when a reaching task is performed.

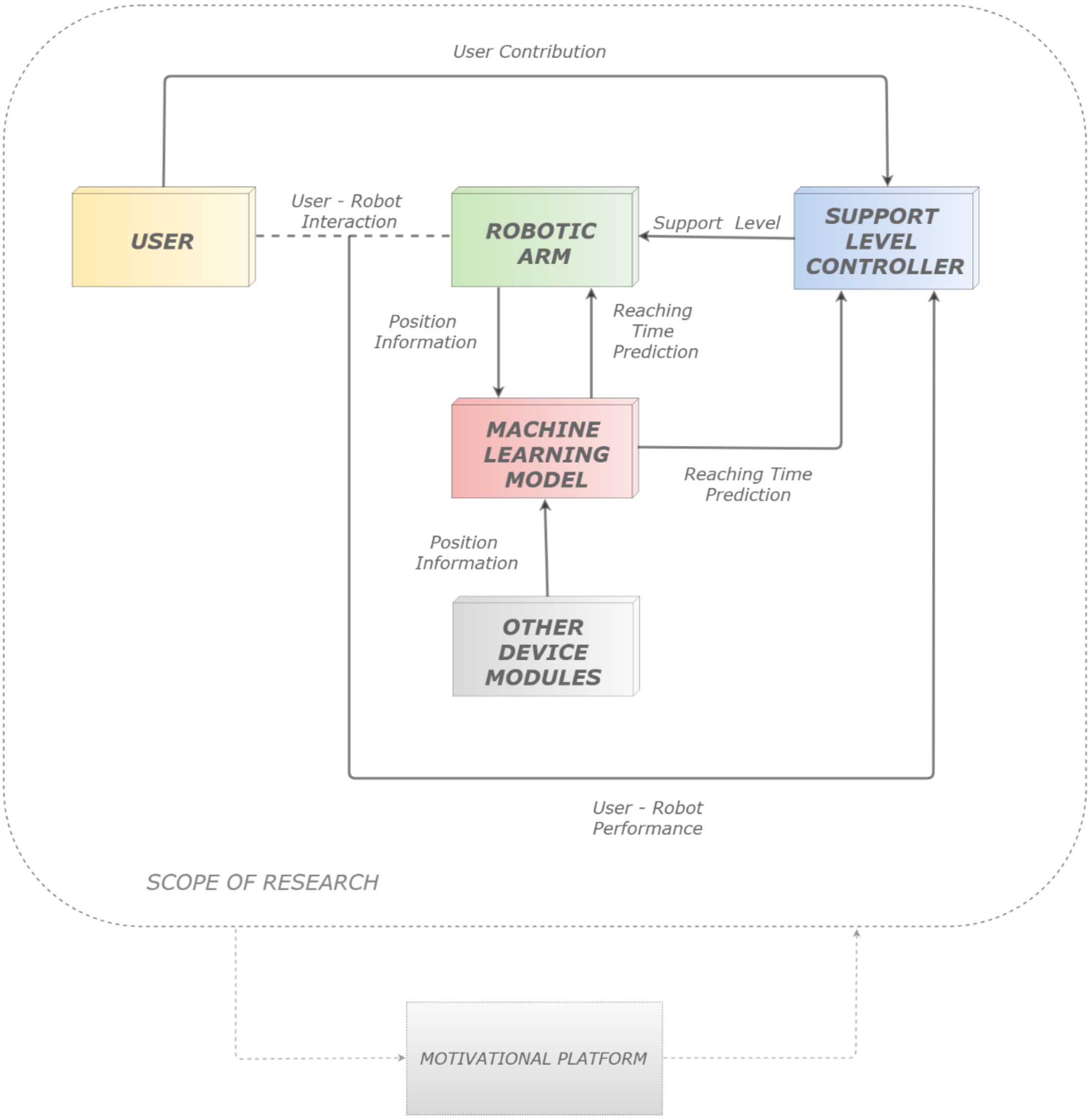


Figure 1 - System Architecture. This model presents the reader with a general idea of the main units involved in the scope of research, as well as the flow of the principal system parameters.

Recalling the previous paragraphs, the main scope of work in this thesis revolves around the behavior of the robotic arm so that it is able to deliver appropriate support levels. Let us further expand on this by stating what is expected from the system, in order to identify potential complications.

Ideally, the robotic arm used by a motor-impaired patient will behave as similar as possible to a healthy individual performing the same tasks. Otherwise, the performance of the user in combination with the robotic arm will not be as high as that of healthy individuals and provided assistance might be insufficient for certain tasks.

Furthermore, the system should promote rehabilitation. If set to full support, the robotic arm will do all the work and there will be no room for user improvement. Moreover, even if not at full support, the robotic arm may still deliver an inadequate support level that will not contribute to the rehabilitation of the user.

Relating back to Figure 1, these two topics – robot behavior and support level- are each bound to specific parts of the overall system architecture. Robot behavior is included in the robotic arm unit, machine learning model unit and other device modules unit, since it determines the performance of the robotic arm. Support level on the other hand, is involved in the interaction between the user and the device, as it is a system parameter whose action will affect the robotic arm unit and consequently, the user-robot interaction.

1.3 Research Question

Based on the above, the main research question can be now specified:

- *What is a possible control strategy such that the device is able to adjust its support during reaching tasks in rehabilitation?*

In order to properly address the research question, it will be subcategorized into two main research sub-questions.

The first sub-question concerns the behavior of the robotic arm. It was concluded that in order to resemble healthy user behavior, the robotic arm should be able to reach a target in space at a similar time as that of the healthy individual. In order to do so, a machine learning approach was proposed as a way for the robotic arm to predict how long it should take the robot to perform the reaching movement. The first sub-question was therefore formulated as:

- *What is a valid Machine Learning approach regarding prediction of healthy user reaching time for upper extremity assistance and control purposes?*

On the other hand, the question of how can the system adjust the support level during exercise, is addressed by the second research sub-question:

- *What is the behavior of a proposed support level controller in terms of delivering different support levels based on user input?*

The approach to address the second sub-question will be to investigate the behavior of a support level controller base on the user-robot interaction and the user contribution. The support level controller behavior references the change in support level set by the controller after several reaching tasks. In such a way, the support level controller unit will contain the necessary commands to ensure the adjustment of the support level.

The connection between problems can be identified in the fact that the behavior of the robotic arm will define the user-robot performance, which will play a role in the adjustment of the support level. In such a way, there will be an interplay between the behavior of the arm -which will have to be sufficient to accomplish the reaching task when the user cannot- and the support level, which will have to promote user participation.

1.4 Report Organization

From now on and for the rest of the thesis, it will be of convenience to state the overall content of the thesis to be divided into two distinct parts with different approaches. The first part (covered in Chapter III) will regard the robot behavior and the prediction of healthy user reaching time, while the second part (covered in Chapter IV) will focus on adjusting the support level for rehabilitation purposes based on system metrics. Each of the aforementioned chapters present the methodology, results and a preliminary discussion regarding their respective topics.

Both parts will be combined during the general discussion (Chapter V) to obtain an answer to the main research question in the form of a control strategy. Lastly, Chapter VI will cover the main findings and conclusions derived from the report.

Chapter II – Background

The background chapter will cover information that is deemed to be relevant for the reader to achieve a better understanding of the topic and framework of the thesis.

It will first start with an insight about stroke, so as to have a clearer conception of this medical condition (Section 2.1). Afterwards, there will be a short overview of the state of the art regarding the application of robotics in assistive and rehabilitation scenarios (Section 2.2). Such scenarios will be the environments on which the eNHANCE device will operate. The set-up of the eNHANCE device will be therefore explained further in the chapter (Section 2.3), in order to show the framework on which the thesis research will be done.

2.1 Stroke – Description, Assessment & Treatment

In order to correctly address and expand on the research question, it is of importance to have a defined idea of the disease suffered by the target patients. This information will not only contribute to the background knowledge, but it will also serve, along the extent of the thesis, as a motivation ground on which certain design and research strategies will be built.

A stroke occurs when a certain region of the brain is deprived of oxygen or damaged due to a cardiovascular accident. The affected brain cells die and the respective bodily function will in turn be compromised [1]. Stroke affects an estimated 15 million people a year around the globe, and it is in fact one of the leading causes of long-term disabilities and death worldwide [2].

Stroke survivors are often affected by major disabilities, ranging from cognitive to motor afflictions. Paralysis and compromised muscle synergies are amongst the most common motor disabilities for stroke patients and heavily affect their performance during Activities of Daily Life (ADL). In order to counteract this handicap, stroke patients often develop compensatory behaviors to accomplish daily tasks. An example of these behaviors being learned non-use, by which the patient stops using the affected body part, given its low performance. This process can be quite detrimental and may exacerbate existing impairments [3].

Several assessment scales have been designed so as to evaluate the status of stroke survivors for further treatment. There are several types of assessments that focus on different issues derived from a stroke. The most relevant to mention for the research presented in this dissertation is the Fugl-Meyer assessment of motor function.

The Fugl-Meyer Assessment (FMA) is a scale which examines the performance of different domains of sensorimotor functionality. It is a widely used test which assigns scores to the stroke patients according to the performance on each domain, where maximum score means full recovery [4].

After initial treatment addressing the most relevant stroke symptoms, post-care is necessary for individuals that have been left with lasting conditions. Based on assessments regarding the patient's condition -such as FMA-, the post-care will be aimed at the recovery of cognitive and motor functions, sometimes varying greatly depending on the patient.

In order to apply post-care, rehabilitation therapies -including physical therapy and rehabilitation devices- offer effective approaches to achieve improved motor function. It has been observed that during rehabilitation, a mixture of beneficial treatments, interventions and therapies undergone over a specific amount of time –usually several months- can recover the functions of stroke patients [5].

One of these rehabilitation treatments is constraint-induced movement therapy. It combines restraining the use of the unaffected extremity, while subjecting the affected extremity to intensive and repetitive task-oriented movements. This technique has been shown to reduce learned nonuse of the affected extremity and to promote functional recovery [6].

During post-stroke rehabilitation treatment, patient motivation is as well generally regarded as an important factor in promoting recovery amongst professionals [7]. Referring back to the system architecture in Figure 1, the motivational platform unit is thus no mere addition, but will become an important part of the rehabilitation process once further research implements it.

Repetitive task-oriented exercises and patient motivation prove therefore useful for addressing upper limb extremity rehabilitation. Although traditionally directed by therapists, robotic devices are able to perform such kind of rehabilitation exercises as well. Consequently, robotic devices have the potential to be used in upper extremity rehabilitation for stroke patients.

2.2 Stroke – Rehabilitation Robotics

This section will give a brief overview of the state of the art on robotics for post-stroke rehabilitation. More specifically, robots regarding upper extremity arm support. This segment is aimed at bridging the gap between stroke rehabilitation and robotics, as well as illustrating the possibilities and potential outcomes of using different technologies –such as the eNHANCE device- in post-stroke rehabilitation.

With the development of new technologies in the field of robotics, robots are increasingly being used for medical applications. In the rehabilitation field, the motivation behind their use is not only to promote and optimize patient recovery, but to additionally reduce the physical workload of the therapist.

Robot-aided rehabilitation presents itself as a useful tool for targeting upper extremity rehabilitation exercises. Within the rehabilitation robots, one can define two categories depending on the function: assistive robots and therapeutic robots [8].

Assistive robots are designed to compensate for the loss of a specific function in a patient. Given that a patient cannot reach the necessary skill level for a specific task, the assistive robot will be in charge of reproducing the lost functionality. Assistive robots can be used for maintaining user independence and in achieving otherwise arduous tasks.

Therapeutic robots on the other hand aim at restoring the functionality of the patient. They achieve this by training the patients in different exercises. Regarding post-stroke rehabilitation, these exercises are more specifically those which are designed to be repetitive and task-specific such as the ones present in the constraint-induced movement therapy. Research has shown that with the appropriate combination of traditional therapy and robotic therapy, upper extremity function is likely to be improved [9, 10].

In a similar fashion, rehabilitation robots can be divided into two specific types depending on their physical configuration: end-effectors and exoskeletons [11].

End-effector robots are connected to the user through a single point at the distal end of the extremity. This interaction point in upper extremity rehabilitation is typically a handle to drive the motion of the limb along a desired path. A key characteristic that sets end-effectors apart from exoskeletons is the lack of a multi-joint system. This makes them more suitable for rehabilitation at later stages, when the patient can input some motor strength to interact with the system.

Exoskeletons on the other hand are devices which contain segments and joints that resemble human anatomy, aligning the exoskeleton axes with the anatomical ones. Given this configuration, they are able to move in coordination with the upper extremities of the user, even within several degrees of freedom. Exoskeletons are suitable for patients at an early stage of rehabilitation, since little motor skills are necessary to accurately control them.

The use of either type of device is up to the application at hand and the available budget, end-effectors being less costly than exoskeletons. Nonetheless, a literature review shows that there can be a positive effect on rehabilitation of upper extremity motor skills from both types of device [12].

As evidenced by the paragraphs above, there are multiple types of robot devices that can be used for different applications within the rehabilitation field. Current research focuses on developing and evaluating said rehabilitation robots so that they are able to interact closely with the user and promote recovery. The interaction between human and device is thus a key component of robotics applied to rehabilitation.

There are several techniques that allow for communication between user and robot. The next paragraphs will very briefly look into some of them and their applications on the state of the art. Special focus will be made on control strategies for robot-assisted therapy, a topic relevant to the scope of this thesis.

One example of such techniques is through direct tactile feedback to the user with the aid of haptic sensors and interfaces. Taking this feedback into account haptics can, for instance, be applied in trajectory control. The haptic forces are thus used as guidance for the user to stay within specific boundaries of the desired trajectory and mechanical end stops ensure that anatomical limits are not exceeded. The haptic feedback in rehabilitating scenarios can be previously modulated so that the user contribution is increased or in such a way that the trajectory is favored [13].

As mentioned in Section 2.1, motivation plays an important role in rehabilitation, especially if taken into account when using robotics. Innovative Virtual Reality environments have been envisioned alongside haptic devices in order to create a motivational platform for stroke patients, with some game-like scenarios. The virtual reality setting allows for a wide range of different rehabilitation tasks which can be beneficial for activities of daily life [14].

Another common technique for assistive control is electromyography. Electromyography (EMG) can be especially useful for getting a robot response from user muscle activity. Research in literature regarding EMG control strategies in robot-assisted therapy include investigation on the feasibility of using EMG signals as triggers for assistance in target-to-target movements on a horizontal plane. In doing so, the residual function of the muscles can be used to specify the assistive torque given to the user by the robot [15]. There are however some limitations to the use of EMG signals, as they are user-dependent and may behave in unexpected ways.

Control strategies for assistance may even combine the aforementioned techniques to obtain performance measures. The concept behind this would be for the robot to assist the user as needed, based on the user’s contribution. In such a way there can be a more dynamic interaction between the robot and the user, while promoting rehabilitation at an appropriate level [16].

To finalize, this section has provided an insight into what type of robots are used for rehabilitation, how robotics can be used for treating stroke patients and which kind of techniques can be used for user-robot interaction.

The reviewed content in literature showed how there are different approaches for control strategies for robots in rehabilitation, which may beneficial for answering the research question. Furthermore, the ample number of studies regarding the use of robotics in post-stroke rehabilitation shows that the framework of the thesis is indeed relevant and in line with current research.

2.3 eNHANCE Device

As evidenced by the literature stated above, there is an increasing demand for investigating and finding innovative approaches to bring robotics to the assistive and rehabilitation fields. The eNHANCE project came to be with the aforementioned in mind.

The eNHANCE project aims at developing a device to assist and train upper extremity motor function in Duchenne and stroke patients [17]. In order to achieve such goal, the project is under the wing of several partners specialized in different modalities, from research universities to manufacturers of assistive technologies.

The realization of the concept within the eNHANCE project is the eNHANCE device. Taking into account the reviewed literature on different types of robots, the device can be classified as a rehabilitation active-assisted therapeutic exoskeleton robot. In other words, it is a device that will promote rehabilitation by providing assistance as needed during specific therapeutic exercises.

2.3.1 Set-up

The set-up of the eNHANCE device is composed of several distinct modules that play a role in the different functionalities of the device. Within the eNHANCE device there are primarily two functions: an assistive function and a rehabilitation function, targeted at Duchenne and stroke patients, respectively.

For the assistive functionality, the aim will be to get users to accomplish otherwise impossible activities of daily life, such as reaching for an object. For the rehabilitation functionality on the other hand, the device will provide assistance as needed to the user, nonetheless promoting rehabilitation by controlling the amount of given support. In order to satisfy the aforementioned functions, the device presents a specific set-up consisting of different modules.

On the one hand, there is an eye-tracking module, which consists of a pair of glasses embedded with eye-tracker technology to detect gaze position relative to the head, a frontal camera to map the vision field and detect objects, as well as a head-tracker with some position markers which signal the head position to a set of cameras. This module is used to detect user intention –along with other intention clues- by estimating the position in the workspace to which the participant wants to direct the task.

A separate module is the robotic arm. The robotic arm will assist in the desired task, after user intention has been determined. Depending on the functionality of the device, the amount of support will be modulated. This support modulation will need to be regulated by assessing the performance of the user. As stated in the introductions section, the behavior of the robot and the control strategy to determine the support level to be delivered are the focus of research present on this thesis.

In addition to the eye-tracker and the robot arm, there are two other modules in the set-up of the device. A wrist module which is in charge of pronation and supination of the hand, and a glove module which will help the user in grasping the target object.

With the action of the modules described above, the device goes from user intention to providing assistance for completion of the task. Future research will investigate the addition of a motivational platform that will offer a training environment, so as to motivate and give feedback to the user as well as obtaining further task performance measures.

At this point, it is of importance to stress that, even though the general set-up of the device will be taken into account when making design decisions during the extent of this report, the focus and module of interest will be the robotic arm.

2.3.2 Robotic Arm

The robotic arm is composed of different sensors, controllers and actuators. The robot has been built previous to the work in this thesis and it does not fall within the scope of the research in this report to examine the different components involved in the robot action and sensing in depth. However, it is relevant for the development of the research to give a brief overview of how the robotic arm works.

Given a specific distance to be covered, the robotic arm generates a velocity profile fulfilling certain maximum velocity and acceleration conditions (see Section 3.1), under certain dynamics. The virtual model on which these dynamics are present is a spring-damper-mass system, with an admittance model for position control. A simple representation of the virtual system can be seen in Figure 2:

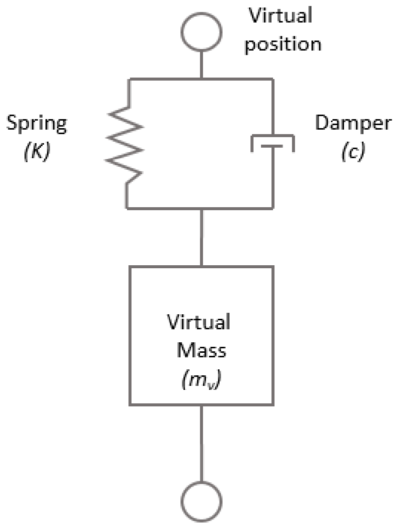


Figure 2 – Robot Dynamics. A simple representation of the model. In it, one can differentiate between the spring with its respective stiffness K, a damper with damping coefficient c, and a virtual mass m_v and position.

The control system will thus drive the movement of the virtual position to the desired position. When the spring stiffness is very high, this will effectively result in the spring-damper system acting as a rod connecting the virtual position and the virtual mass. In reality, this is translated into the robotic arm making the user arm and the support arm accurately travel the required distance.

If the spring stiffness of the system is lowered, one can think of the spring as being slacker, and thus the virtual mass will lag behind the virtual position. At a certain stiffness value this will result in a great distance error between the desired position and the actual position, unless the user exerts some force in the proper direction.

With the above in mind, the support level can be defined as a value that modulates the spring stiffness. The equation relating support level and spring stiffness was set as:

$$K = \frac{SL^3 m_v \left(\frac{4a}{\pi v}\right)^2}{1.001 - SL}$$

Equation (1)

Equation 1 defines the value of the spring in terms of the support level. The nomenclature of the equation is as follows: *SL* is Support Level, m_v is the virtual mass value, *v* is the maximum velocity value and *a* is the maximum acceleration value. All units are in SI.

From the equation, it is clear that at a support level equal to 1, the value of the function will be very high and that when the support level is equal to 0 the stiffness will be zero. In the case of zero a stiffness, a global damping was added to the model so as to avoid instability in movement. Between the 0 and 1 support level, the value of spring stiffness makes the robot more or less compliant to the controller action and the user force input.

Aside from the control described above, there are other strategies within the robot that ensure the correct movement is performed. An example being the control of user and robotic arm configuration by determining the elbow swivel angle. With it, the target is reached following a trajectory that takes into account the arm length of the user.

Besides the different controllers, the robot is similarly equipped with different sensors, such as encoders to determine the position. Especially important during this report will be the interaction force sensor measuring the forces between the user and the robot, as they provide useful information during movement.

An arm support is used to hold the forearm of the user. The force sensor is located in the proximal side of the arm support, and approximately aligned with the elbow. In such a way, the force sensor records the interaction forces between the user and the robot during a reaching task with respect to its location near the elbow.

Figure 3 shows the set-up of the robotic arm and the position of the force sensor, the arm support and the other robotic arm components.

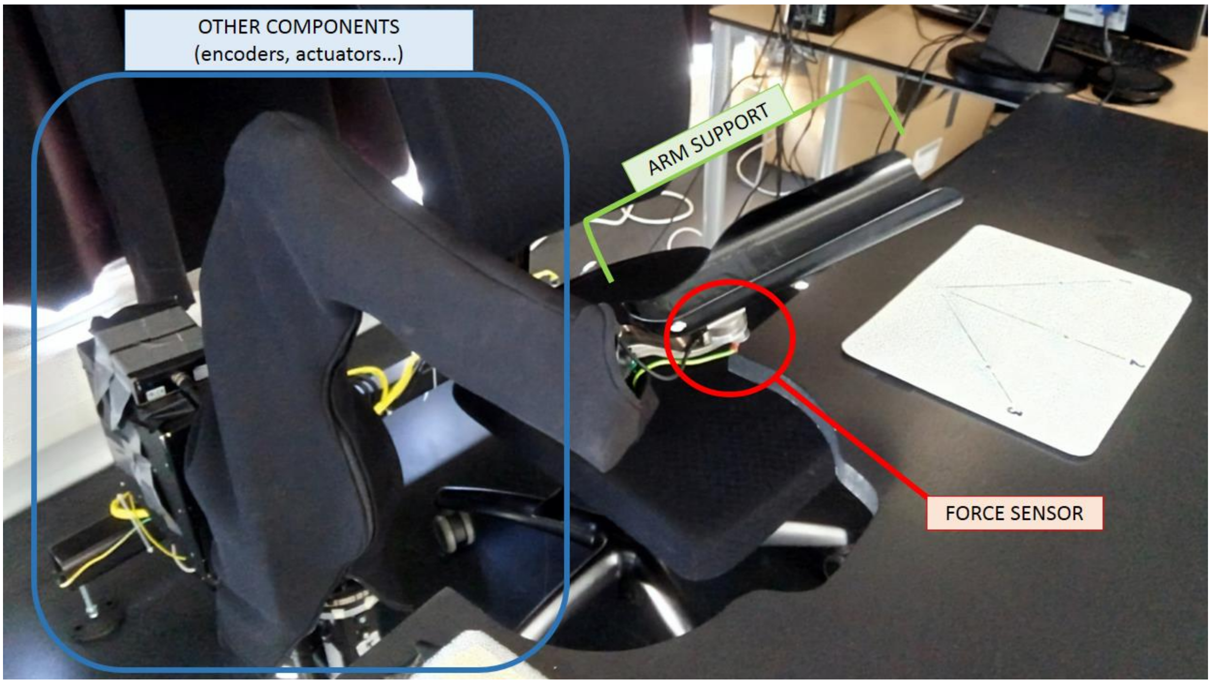


Figure 3 - Robotic arm set-up.

As a final remark, it is worth mentioning that a Robotic Operating System (ROS) is used as a middleware. Through the ROS environment, communication is thus possible between the different systems within the robotic arm.

The overview of the eNHANCE device presented in this section was aimed at providing useful information that will be taken into consideration when making design choices during the report. Given the main research question, the concept of support level presented in this section will be of special importance for delivering appropriate assistance with the robot arm in order to promote rehabilitation in stroke patients.

Chapter III – Time Prediction Validation

This chapter will focus on the movement of the robotic arm. More specifically, on how the behavior of the robotic arm might be specified, based on healthy user reaching movements. To that end, a machine learning approach will be laid out and a validation experiment carried out. There will be a brief introduction to situate the research into context in Section 3.1 as well as an approach to solving the problem. Then, in Section 3.2 the methodology followed during this chapter will be explained. This will lead to presenting the results in section 3.3. Finally, during section 3.4 a preliminary discussion regarding the results will be laid down.

3.1 Introduction

The behavior of the robotic arm is a key property of the system. This is illustrated by the interaction shown in Figure 1, where the robotic arm unit plays a role in robot behavior and in which it receives the support level controller's output as an input. If one were to lower the abstraction of the system to look into the robotic arm unit, it would be composed by the robot dynamics subunit, a velocity profile generator subunit and other components subunit, as shown in Figure 4:

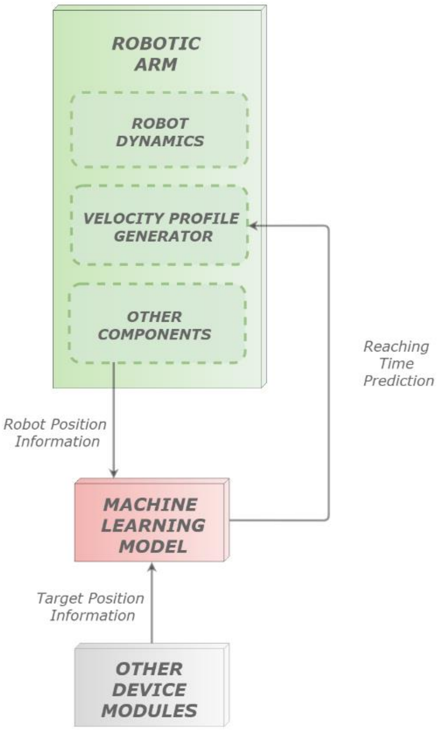


Figure 4 - Robot Behavior: Robotic arm interaction. The level of abstraction of the robotic arm unit is lowered in the Figure. Thus, one can observe three inner units of the robotic arm: the robot dynamics, a velocity profile generator and other components. The interaction between the machine learning model, other device modules and the robotic arm will determine the behavior of the robot.

It is important to remark that only the relevant system parameters and units have been depicted in the figure above, for the sake of clarity. The robot dynamics and the velocity profile generator were mentioned in Section 2.3, describing the robotic arm within the eNHANCE device. It was similarly mentioned that in order to cover a certain distance in a specific amount of time, a reference velocity profile is needed. The velocity profile generator allows the user to input either desired movement time or distance, in return, the generator will output a velocity profile that fulfills the profile, given a maximum acceleration and velocity.

As seen in Figure 4, the desired movement time will be the reaching time prediction coming from the machine learning model. In order to output such prediction, the model will use position information about the robot and the target, coming from other components within the robotic arm and from other device modules respectively (see Section 2.3).

This chapter will evaluate the performance of different trained machine learning models in different types of reaching tasks in terms of prediction accuracy. This will determine whether machine learning can be used as an approach to control robot behavior so that it resembles the behavior of healthy users.

3.2 Research Methodology

The purpose of this section will be to present the relevant methods used to carry out the research in Chapter III. Such methods include: the evaluation metrics used to assess the performance of machine learning algorithms (Section 3.2.1), the machine learning model selection (Section 3.2.2) and an experiment in which the machine learning models were tested during different reaching tasks in Section 3.2.3.

3.2.1 Evaluation Metrics

Evaluation metrics are useful indicators that evaluate the performance of the different kind of regression machine learning models. In this report, there are three main metrics used to assess the models: the Percent Error Accuracy, the Root Mean Squared Error and the Mean Absolute Error.

Percent Error Accuracy

The percent error accuracy (see Equation 2) is a precision between an experimental value and a known value. It is calculated by subtracting the obtained value to the target value, dividing the difference by the reference known value and obtaining the percentage measure:

$$Percent\ Error\ Accuracy = \frac{|\,y - \hat{y}|}{y} * 100$$

Equation (2)

In such a way, it provides information on how different is the experimental value to the target value. A percent error close to zero means that the obtained value is close to the desired value.

Mean Squared Error (MSE) / Root Mean Squared Error (RMSE)

The mean squared error is a measure of error in the prediction. It yields the mean squared value of the difference between prediction and actual value and thus indicates how far away the prediction is from the true value, i.e. the magnitude of the error.

The root mean squared error (RMSE, Equation 3) is the square root of the MSE and takes the units back to the original units of the data. This can be useful for interpretability of the MSE results. The RMSE formula is as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

Equation (3)

Given that the difference is squared, high errors are given more weight in the overall RMSE metric. RMSE is therefore useful to assess models in which high errors are notably worse than smaller errors.

Mean Absolute Error

The Mean Absolute Error (or MAE, see Equation 4) is the average sum of the differences between prediction and actual value, with disregard to whether the estimation is over the actual value or under it. Just as RMSE, it expresses the prediction error in the same units as the output variable. However, it gives the same weight to every prediction error:

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

Equation (4)

Regarding which metric one should rely on between RMSE and MAE depends on the application at hand. RMSE is harder to interpret and has the added drawback that RMSE comparison between different sized data samples is difficult to analyze. On the other hand, the weighted errors in RMSE can be preferable for detecting the error sensitivity of different models [18].

3.2.2 Machine Learning Model Selection

This section will cover the methodology followed to reach a machine learning model for the application at hand. This concerns the machine learning model unit from the system architecture as seen in Figure 5:

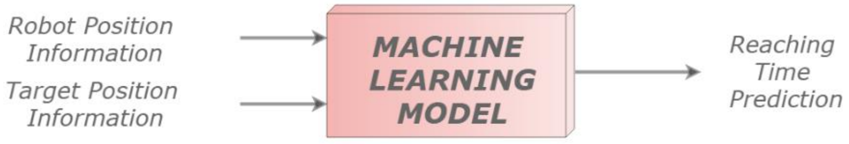


Figure 5 – Machine Learning Model Unit. The selected Machine Learning Model will take robot and target position as inputs and output a predicted reaching time towards the robotic arm.

The reaching time prediction will be used as a system parameter to alter the behavior of the robot after being used as an input to the velocity profile generator. In order to investigate reaching time prediction, a selection of possible machine learning models must therefore be made.

This section will first give a brief overview of the types of models to be used during this study, to a have a clearer view of their differences. Then, the model selection methodology will be presented.

- Machine Learning Algorithms

The type of machine learning techniques to be investigated throughout this research will be supervised learning regression models, more specifically, decision trees.

The complexity of the models was a key factor that affected the decision to only investigate some models over others. Complex models take a lot of computational power, which means expensive equipment that can handle such computations have to be bought. Similarly, a difficult implementation of the model may prove time-consuming for future researchers and there is no need to have an extreme level of accuracy, given the application and the limitations of the robot.

For such reasons, models such as Neural Networks or Support Vector Machines are not explained nor evaluated in the work presented during this thesis. Decision trees were thus selected as potential machine learning algorithms to be used for the applications at hand.

Decision trees are a type of nonparametric models. These kinds of models can come up with variable mapping functions and thus generalize data quite well. In such a way, they are more flexible than parametric methods and have greater predictive power when the underlying structure of the data is unknown. However, they require a large dataset for accurate estimations and might overfit the training data if not tuned properly.

With respect to other nonparametric methods such as neural networks, decision trees are easier to interpret, they can reach a prediction much faster and are more convenient to tune. However, the prediction accuracy of decision trees can be lower than other methods especially when there subjacent non-linearities in the data [19].

There are several different types of decision trees models. Those that have been analyzed in this study are:

a. **Single tree (ST)**: The simplest form of decision trees is a single tree. This kind of decision tree is very easy to interpret and a widely used prediction model. If not tuned properly however, the decision tree models tend to overfit the data, showing a low bias – representing the underlying target function very well- but high variance – where predictions from other datasets are poor-. The bias-variance trade-off is often addressed by pruning the tree, i.e. limiting the depth of the tree.

b. **Random Forest (RF):** Random forests are models that aim at reducing the variance observed in single trees [20]. For that purpose, they are models that average single trees, each trained with different data from the training dataset. This process reduces the variance of the model, increasing the performance of the final model. In such a way, after training several decision trees estimators, the final prediction will be the mean output prediction of the estimators. The Random Forests algorithm is based on the combined use of bagging and random selection of features.

   Bagging consists of drawing random instances from the training dataset to train a single decision tree that will be later averaged. The instances of data are each randomly selected and replaced, in other words, one given observation can be drawn more than one time from the training dataset.

   The random selection of features on the other hand, occurs at every node, in which -from the random selection- one of the features will be chosen based on optimality in the binary splitting.

   Through these techniques, there is a loss of interpretability. However an increase in variance can be achieved -given a sufficiently large training dataset size-, when compared to a tuned single decision tree.

c. **Extremely randomized forest (ERF):** Extremely randomized forests are based in the design process of random forests, but with a few modifications [21].

There are two main distinctions. The first one being that, instead of bagging, extremely randomized forests draw observations for each estimator from the whole training dataset. The second one is the full randomization of the attribute used in a node, without optimizing the selection based on the chosen criterion. Through this modifications, they aim is to further decrease variance, given the additional randomization.

When compared to random forests, ERFs do not always perform better in terms of accuracy. However, they can be computationally faster since there is no computation time spent in selecting the best attribute to split at every node.

d. **AdaBoost (ADA):** AdaBoost is another method to reach a decision tree based model. It is a boosting technique that resembles random forests when single decision trees are used as estimator [22]. The main difference with the random forest is the use of boosting instead of bagging.

During boosting, a main model is used, which is an averaging model updated through several iterations. At first, this model will be trained and the prediction error assessed. Data in the training dataset that are poorly modeled get a higher weight and the training dataset is modified for the next iteration.
The new training dataset is used and a new model obtained, which will be averaged in the main model. This process is repeated several times until a weighted average prediction is reached.

In such a way, AdaBoost focuses on minimizing bias, but it may overfit the data if high variance models are used. Furthermore, it is more computationally expensive than other decision-tree based models [23].

- Model Selection Methodology

The next paragraphs address how to find appropriate candidate models to predict healthy user reaching time. It is important to remark that the presented models are far from definitive or invariable. Instead, they are final products of a methodology that can be followed in later stages of the project to find other good candidates. Therefore, the general methodology will be the main outcome of the model selection process, while using the ML models as demonstrative tools.

Two important definitions must be explained before detailing the model selection methodology: hyperparameters and cross-validation.

The hyperparameters of a model are parameters that are not learned from training, but instead are high-level fixed parameters that affect the structure and complexity of the model. In decision trees the hyperparameters of a single tree are for example the depth of the tree, the number of samples to be considered in a leaf or in a split and other fixed parameters. In an ensemble of decision trees, the same applies in addition to the number of estimators to be used. Although it is not straightforward to find the best parameters for a model, hyperparameter tuning can positively affect the performance of a model if done right.

Cross-validation - or k-fold cross-validation- on the other hand, is a technique in which the available dataset is split into several groups to test the performance of the machine learning model. First, the data is shuffled randomly and split into k groups. For each group, there will be a training dataset and a test dataset so that the model is trained and tested against them. An overall performance metric of the model is finally obtained after using each of the groups as test datasets.

During the model selection process the Spyder Python 2.7 environment was used as the platform for data processing and analysis. This environment is written in Python, which is a widely used programming language in data analysis in machine learning applications. There are therefore many open source packages to deal with machine learning analysis and it is thus of use along the extent of the work presented in this report.

With the above in mind, the outline of the methodology can be defined. First, and as a preliminary step, the training dataset will be looked into. After training, a base model for each machine learning algorithm was obtained. From this base models, feature selection and a hyperparameter search were done so as to investigate whether the model performance was increased. To conclude this section, the selected machine learning settings are specified after the methodology.

0. Training Dataset

The training dataset comes from the data acquisition done in a previous experimental study within the eNHANCE project. Said study investigated the matter of performance measures and their validation for the robotic arm support. These measures were based on the data collected by the sensors used in the system while the user performed base-to-target (BtT) tasks.

During said study, each measurement was made at different target steps, acquiring data in the process. Figure 6 shows the reaching tasks and an overview of the experimental set-up.

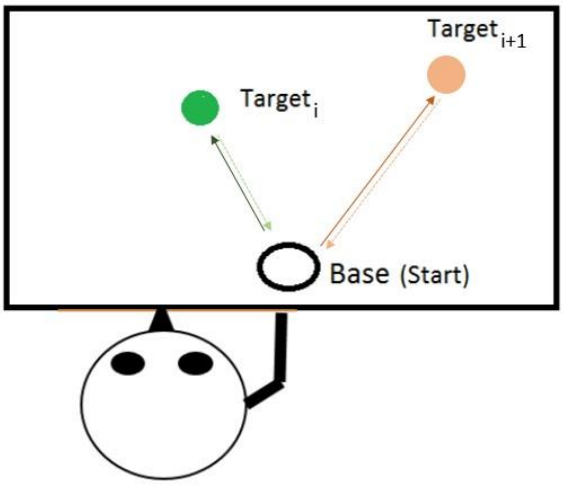


Figure 6 - Set-up during previous research.

With the data collected from the previous study, feature engineering will be the first preliminary step in model selection, which consists of selecting and deleting features based on the relevance to the prediction.

Data was collected on MSJ and reaching time at every Base-to-target movement. It was decided that there is no need in using nor predicting MSJ for the application at hand, as it was not needed for the behavior of the robot nor in later stages as a user performance metric.

Similarly there was a weight parameter (W) that was used to investigate performance during previous research, it referred to added weights place on the arm of the participants. Since the reaching time predictions will concern healthy users performing normally, this parameter was excluded as well.

In such a way, the features of the training dataset for its use in model selection were set to be: the position of the target in x and y coordinates, the distance from the base to the target, the angle between the base position and target position. After some preprocessing of the previous data, the training dataset used in the model selection section was of the form described in Figure 7.

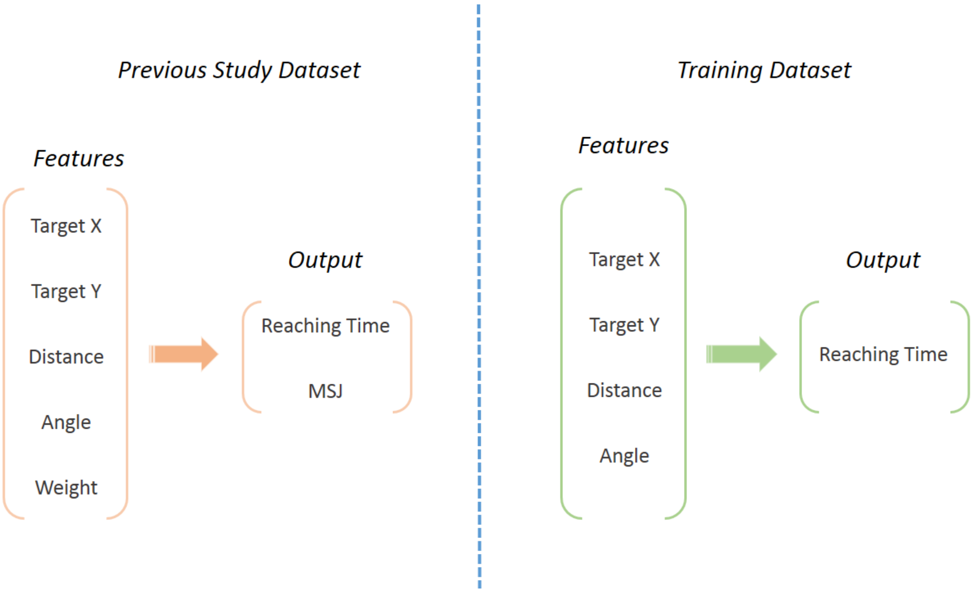


Figure 7 - Features and output of the training dataset. On the left the previous Study Dataset figures, on the right the Training Dataset to be used in model selection after feature engineering.

1. Base model

A preliminary selection of hyperparameters was chosen for each algorithm as a starting point of the model selection analysis. In order to minimize the error and validate the models, several hyperparameters were iterated through, namely the number of k-folds, number of leaves -in the single decision tree- and number of trees –in the ensemble methods-. Due to the motivation to choose a fast trained model (see Section 3.4), the time was another criterion to be minimized. In such a way, 0.25 seconds was chosen as an upper boundary threshold for model selection. Through visualization in Python, appropriate parameters were chosen for each model that minimized both RMSE and time.

2. Feature selection

A model may rely on a set of features for prediction more strongly than on others. Eliminating unnecessary features may result in an improvement in performance, training time and prediction time. Feature engineering as done in the training data paragraph of this section already took care of selecting the most intuitive features, however the underlying significance of a feature for a specific model may not be that straightforward.

Through the Scikit-learn library in python one can get a percentage of the importance of each feature for the model at hand. For all base models the angle feature was the most important even though slightly. The difference of importance between features was small and thus it was decided that all features from Figure 7 will be used in the dataset for model selection.

3. Hyperparameter search in python

An iteration process was manually coded in order to find hyperparameters to create the base models. Scikit-learn however, includes two automatized hyperparameter selection techniques namely a random search and a grid search. In both, a set of possible hyperparameters is given to the function and, as an output, the best possible combination of hyperparameters in terms of a specific performance metric is returned.

In a random search only a random set of these hyperparameters are investigated and although the performance of hyperparameter selection is lowered compared to a full grid search, the running time is drastically lowered.

In the model selection undergone in this study, only a random search was run. Since the performance of the base models was already determined to be good enough (see Section 3.3 and Section 3.4), there was no need to run a full grid search.

This ends the model selection methodology. As stated above, the models analyzed in this section were supervised non-parametric regression models, namely: a single decision tree, a random forest, an extremely randomized forest and an AdaBoost model.

After analysis, the chosen models for each machine learning algorithm and that will be used in the next section thus were:

- For the Decision Tree a minimum sample leaf size of 80 with 12 k-fold cross-validation (CV).
- For the Random Forest, an ensemble of 6 regression trees with a 2 k-fold CV
- For the Extremely Randomized Forest, an ensemble of 7 regression trees with a 2 k-fold CV.
- For the AdaBoost model with 8 estimators with a 2 k-fold CV.

3.2.3 Experiment I – Reaching Time Prediction

This section of the chapter will address the methodology of Experiment I. Said experiment was design in order to assess the validation of the selected machine learning models for their use in reaching time prediction as well as determining whether the acquired training dataset can be used to predict target-to-target (TtT) reaching tasks. The section will start with the motivation, research question and concept of the experiment. Next, the followed methodology including recruitment, measurements, procedure, data acquisition and data processing will be presented to the reader.

There are several reasons that motivate Experiment I. Firstly, given that the available data from previous studies were acquired during base-to-target reaching motions, it is important to consider whether such data is able to predict healthy reaching time tasks for target-to-target motions. Indeed, during activities of daily life one would not expect movements of the arm to be limited to a base to target motion.

The experiment will most importantly provide an assessment for the machine learning model trained with the previous study dataset during target to target reaching tasks. This will be of interest in future data acquisition procedures, as it will help in determining which features are necessary for correctly predicting the reaching task time that will be used to drive the robotic arm action.

Secondly, the experiment will also serve itself as a data acquisition procedure for updating the training dataset. In such a way, during the experiment, data from base to target and target to target tasks will be obtained.

The experiment thus revolved around the following research question:

- Can a Machine Learning model trained with a base-to-target (BtT) dataset of healthy users at maximum performance, be used to accurately predict target-to-target (TtT) reaching task time?

The general concept of the experiment is shown in Figure 8. The BtT trained machine models from the model selection section task as input the experimental data from BtT and TtT reaching tasks. The resulting prediction will then be compared to the actual values recorded during the experiment and after analysis, a prediction accuracy in terms of the performance metrics will be obtained for each BtT trained model for both BtT and TtT tasks.

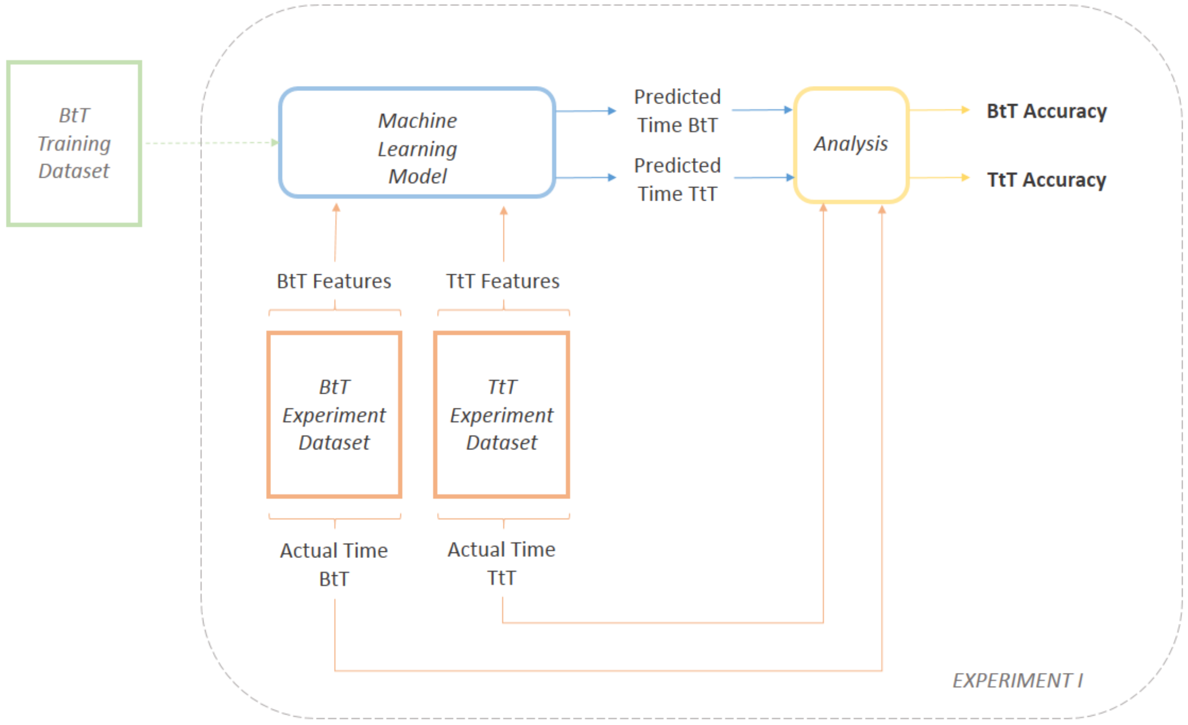


Figure 8 - Concept of Experiment I.

The design of the experiment was laid out with the above concept in mind. The methodology of the experiment was thus built around the experiment concept.

The methodology covers the processes by which data goes through, from procedure to data acquisition and processing. It serves as a guideline for handling and processing data obtained from Experiment I and it was developed on the basis of the previous research studies.

- Recruitment

The experiment was carried out in the Zuidhorst building of the University of Twente, in room ZH284. The target population of this study was adult, healthy individuals. Exclusion criteria included left-handed individuals and those individuals with lowered motor and visual functions. The accessible population covered individuals studying at the University of Twente. A sample of 7 individuals was drawn from the accessible population.

- Measurements

The experiment was performed in an enclosed space at the University of Twente, with the Liyama prolite TF4237MSC-B1AG touchscreen as hardware and the MATLAB-based user interface as software. The participant sat on a chair, and had their movement restricted so that the shoulder joint can move freely but the trunk of the body is kept still.

- Experimental protocol

Previous to the experiment, each individual was given an information brochure and consent form. Once they agreed to participate they were asked to perform the experiment as instructed. To ensure similar conditions, the participants were sat on the same chair and asked to situate their body midline as aligned to the screen midline as possible. An elastic band for body use was placed over the participant's lower chest area and over the back of the chair, to prevent leaning. The participant was asked to solely use their index finger for the experiment and to conduct it at maximum performance.

The experiment was then carried out as follows:

1. Firstly, a calibration task on which the area that the participant could cover was established.
2. Secondly, a Base to Target task on which the participant had to, starting from the base position, reach a randomized target and come back to the base position.
3. And lastly, a Target to Target task on which the participant had to reach for the randomized targets sequentially without going back to the base position.

This scheme was repeated three times for each participant. Between all three parts, a resting time of about 3 to 4 minutes was left for the participant to rest their arm, to account for fatigue effects. The calibration task lasted for about 10 seconds and the other tasks were 120 seconds long. Figure 9 and 10 show a representation of the tasks. Figure 11 shows a user-perspective real-life picture of the reaching tasks.

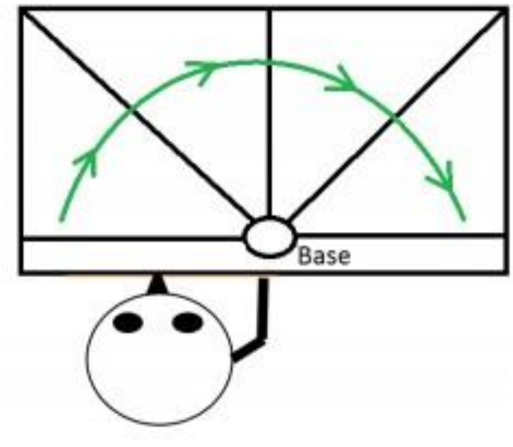


Figure 9 - Maximum range of motion calibration task.

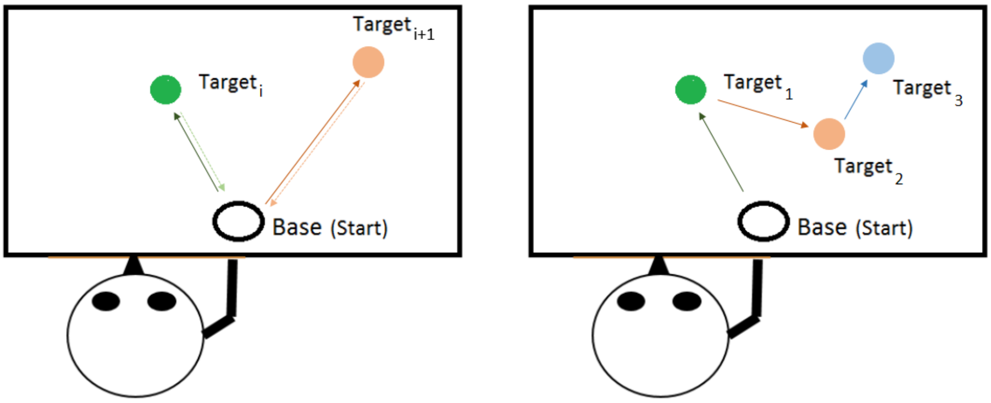


Figure 10 -Set-up of the BtT task (left) and of the TtT task (right).

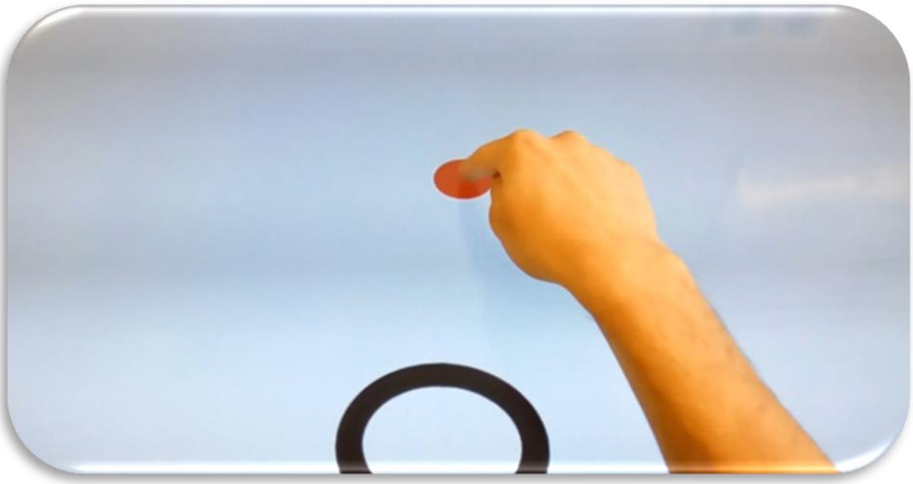


Figure 11 - Real-life user perspective of the set-up for reaching tasks.

- Data Acquisition

The acquisition of data was possible through the use of a Liyama prolite TF4237MSC-B1AG touchscreen as hardware and the MATLAB-based user interface as software. The screen was connected to the computer in such a way that touch-based inputs on the screen were read as mouse actions on the computer.

A MATLAB script was used for the generation of a randomized set of targets and for the acquisition of touch-based mouse positioning and timing. A general script contained calls to the functions in charge of the three different parts of the experiment. After finalization of each task, a ".mat file" was saved, containing the raw data.

- Data Processing

Data processing followed the next steps:

i. The first 60 seconds of data were eliminated to account for learning curve effects. The participant's learning curve reached a plateau within the 60 seconds of performing the task.
ii. Only base to target data was taken into account, since the target to base data is not representative of the task.
iii. Unsuccessful reaching attempts were located and erased.
iv. The elapsed time between the successful attempts was computed.
v. Distance and angle between the starting position (touch-based position) and the target position were computed.
vi. Removal of time outliers *± 2*std*. (std = standard deviation of normal distribution).
vii. Data was saved in an Excel ".csv" file, containing information on elapsed time, starting position x and y, target position x and y, distance and angle.

In order to get a single evaluation of each model, a script was created that read the ".csv" files containing the training data and the preprocessed data. The script also contained the four models from the preliminary study discussed in the previous work section, namely a single decision tree, a random forest, an extremely randomized forest and AdaBoost. The models were set with the hyperparameters presented in Section 3.2.2. The rest of the settings for the models are set as default and can be consulted at the Scikit-learn documentation [24].

Once the script read both the training dataset and the experimental dataset, it fit the training data to the model and, with the appropriate inputs, it predicted the reaching time from the experimental data. Ten total prediction datasets were generated.

For each of the ten experimental vs. prediction datasets, the evaluation metrics were obtained. Afterwards all values were averaged obtain a single value for each evaluation metric, for a specific participant, task and ML model, as seen in the results from Experiment I during Section 3.3.

3.3 Results

This section will present the results concerning Chapter III. There will be two clear parts within this section: the results from the model selection methodology and the results regarding Experiment I.

Model Selection

For the model selection, an analysis on model performance -with RMSE as selection criterion- was done during each step of the model selection process for all machine learning algorithms.

In such a way, the base model of each algorithm –single tree, random forest, extremely randomized forest and AdaBoost- was trained with both the raw dataset (RAW) from the previous model and the dataset after feature engineering (FE). Later, new tuned hyperparameter models after a random search were trained with the dataset after feature selection (HT). Figure 12 shows the results of the analysis:

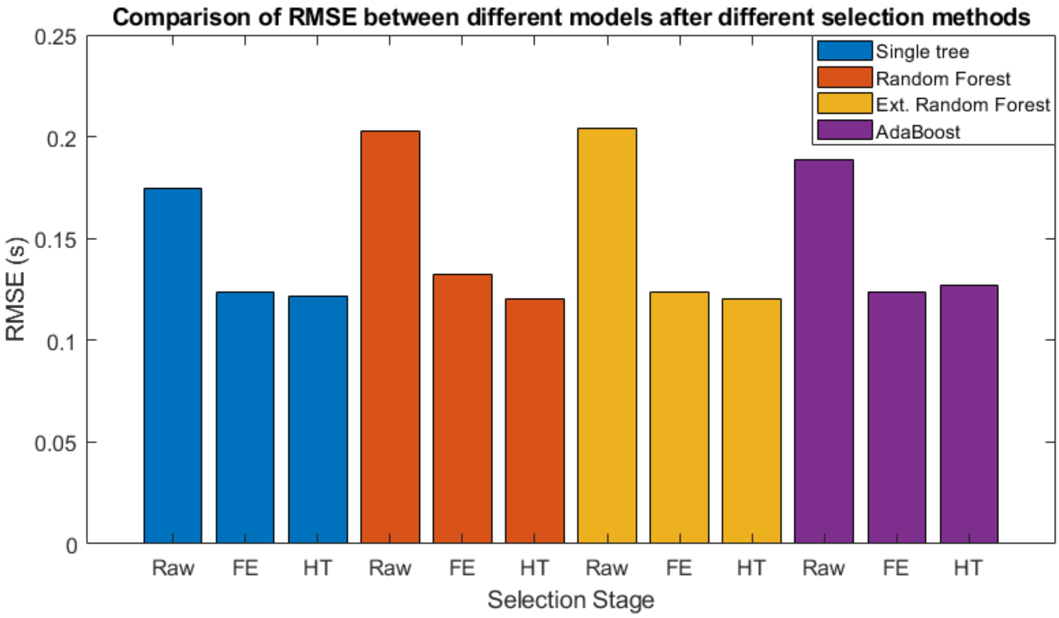


Figure 12 – Model Selection plot. Comparison of the Root Mean Squared error for different machine learning algorithm models at each selection stage. Raw is the dataset from the previous study, FE is after feature engineering, HT are the models after the random hyperparameter tuning.

The results show that a significant improvement in performance –lower RMSE- was obtained after feature selection for all models. However, the performance difference between the feature engineering and the hyperparameter tuning stage is minimal, with even a decrease in performance for AdaBoost after hyperparameter tuning. Even so, all stages show an RMSE which can be within the limits of acceptability for the robot behavior i.e. an RMSE in between 0.1 and 0.2 seconds. It was therefore decided that the base models would be used in Experiment I and that no further hyperparameter tuning or selection process needed to be done.

Experiment I

The next paragraphs cover the results of Experiment I, showing which machine learning algorithm shows the best performance so as to assess whether the BtT trained models can predict TtT reaching time.

As mentioned before, the three evaluation metrics of Section 3.2.1 were used in the assessment of the machine learning models: percent error accuracy, RMSE and MAE. The evaluation metric values in Figure 13, Figure 14 and Figure 15 were obtained after following the methodology for Experiment I as explained in Section 3.2.3. All data has been averaged across the participants for the specific task.

The associated values for each model – namely: Single Tree, Random Forest, Extremely Randomized forest and AdaBoost-, are depicted on different columns for each task. The comparison techniques include CV for obtaining cross-validation evaluation metrics, BtT for the metrics obtained by comparing the prediction to the BtT experimental data and TtT for those metrics obtained through the TtT experimental dataset.

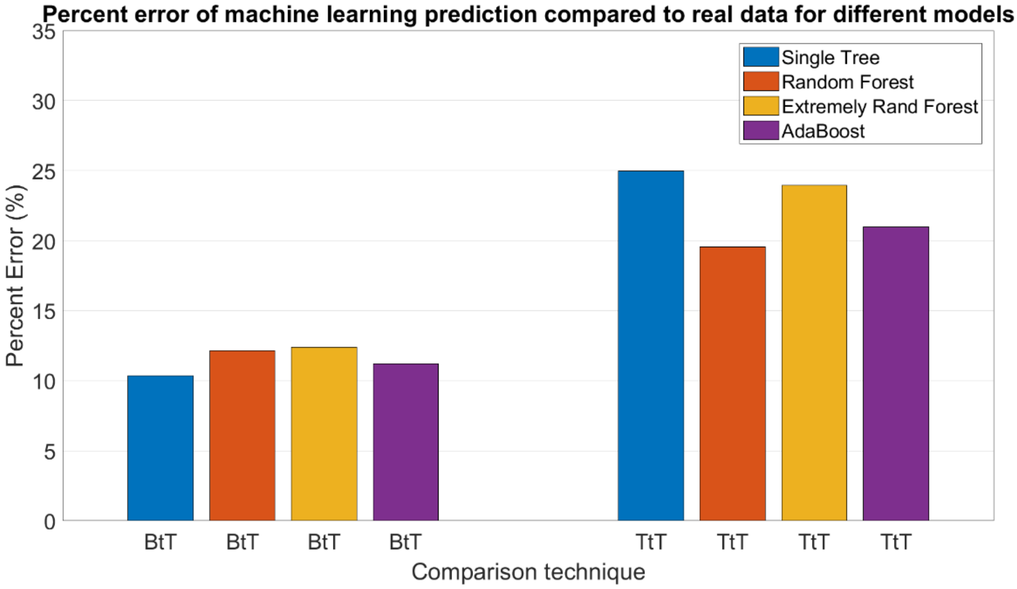


Figure 13 – Percent Error plot. Obtained percentage error in prediction compared to the real experimental data from BtT tasks and TtT tasks across all ML models.

Figure 13 shows the percent error found during either the BtT or TtT comparison for all models. For the BtT comparison, the predicted values are between 10% and 12% off in relation to the experimental value for all models. In contrast, for the TtT comparison, an increase in the difference between experimental and predicted values is noticeable, with values ranging from 19% to 25%.

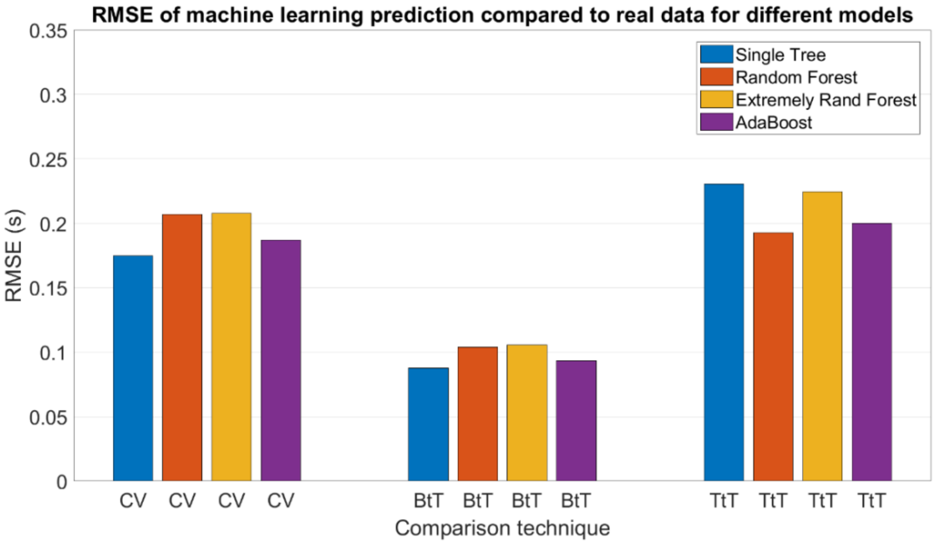


Figure 14 – RMSE plot. Obtained Root Mean Squared Error in prediction compared to the real experimental data from BtT tasks and TtT tasks across all ML models.

Figure 14 shows the results regarding RMSE for all models in CV, BtT and TtT comparison. The errors in CV fluctuate between 0.17 and around 0.21 seconds for all models, whereas in the BtT comparison, these values are reduced and more accurate within a range of 0.09 and 0.11 seconds (see Section 3.4). Similarly, in the TtT task, a maximum error of 0.23 seconds was found for the single tree model and a minimum of 0.19 seconds for the random forest model.

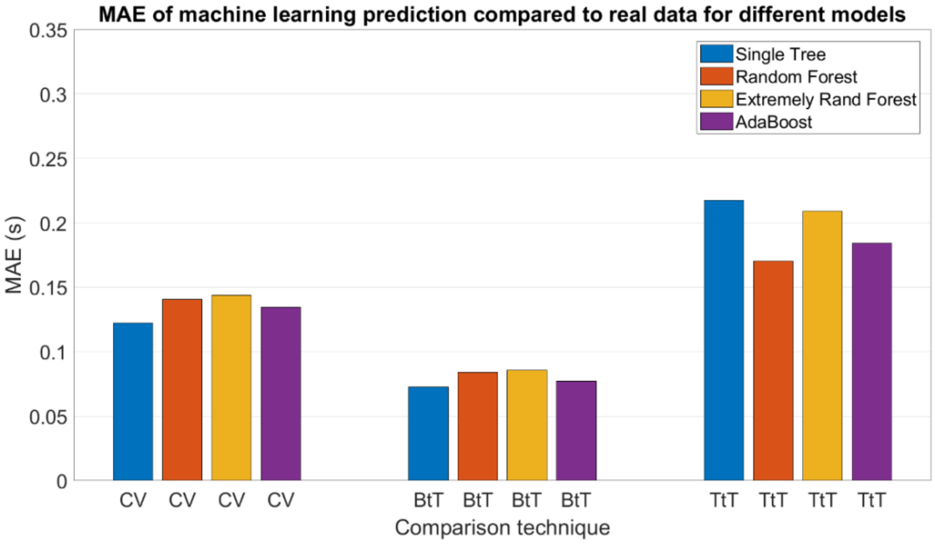


Figure 15 – MAE plot. Obtained Mean Absolute Error in prediction compared to the real experimental data from BtT tasks and TtT tasks across all ML models.

Figure 15 on the other hand shows the MAE in seconds across comparison for all models. It is noticeable that the general trend in the results resembles that of Figure 14 (see Section 3.4). The observed MAE in CV is under 0.15 seconds. In the BtT comparison, the results vary between 0.07 and 0.09 seconds across models. Lastly, the TtT shows the biggest error from all comparison techniques, with an upper limit of 0.22 seconds and a lower limit of 0.17 seconds.

3.4 Preliminary Discussion

Combining both the model selection and Experiment I results, the purpose of this section is to get one step closer to answering the first research sub-question by preliminary discussing the results.

From the model assessment, the single decision tree was the fastest model and the one which performed better in terms of RMSE, allowing for several k-fold cross-validations levels. Out of the ensembles, AdaBoosting showed the lowest RMSE. In practice however, one should be inclined to use ensembles instead of single decision trees, since decision trees tend to overfit the data whereas the ensembles - and more specifically the boosting method -, provide a more robust implementation and can handle outliers better.

Given the results, some clarifications need to be made about the model selection procedure. In a first approach to model selection, the training time of the ML model was agreed to be kept low and similar to human reaction time. In such a way, the model selection was determined by finding an appropriate ML model which would minimize the selection criteria: training time and RMSE as mentioned in the model selection methodology. Time and RMSE were both simultaneously optimized after iteration through the hyper-parameters of each model, and so a definitive structure was chosen for each model.

After careful consideration during the discussion of the experimental results, it was decided that even though the training time should not be excessively high, there will be no time criterion restriction in successive models, as they can be independently updated with data from several patient sessions outside of the use of the device in the rehabilitation setting.

However the selected models described in the model selection methodology and used during Experiment I were those fulfilling both the RMSE and the minimum training time criteria. The reason to keep these models over choosing new ones was threefold: first, the performance after analysis of the models was within a reasonable limit for the application at hand – between 0.1 and 0.2 seconds- and therefore further improvements in model performance would not add extra value to the predictions.

Second, a low training time is usually associated with models that do not require much computational power, in such a way, the model can be updated within a short period of time without expensive equipment.

Third, these models are only used as a way to represent the ML model selection process for its use in the behavior of the robot, as well to validate the use of a model trained with data from one reaching task to predict another type of reaching tasks. They are however subject to change and therefore it is not extremely important to accurately tune them.

This concludes the discussion on the model selection for this chapter, other relevant points will be made during the discussion chapter (Chapter V).

For the preliminary discussion on Experiment I, the first focus point should be on the Base-to-Target (BtT) task results, since it was included in the experiment as a control ground on which to assess the training dataset and the results of the experiment for the Target-to-Target (TtT) task.

The general trend of the RMSE value for the BtT task across the different values withholds for the results. From lower to higher RMSE: ST < ADA < RF < ERF. This fact may come from the multiple cross-validations on the ST and the limited CV of the ensemble model due to the limited training time criterion. The RMSE however still remains a good indicator of accuracy, and it is noticeable that the prediction RMSE is reduced by around half of that of the $\mathrm{RMSE}_{\mathrm{CV}}$. This may arise from the fact that the CV training dataset is a smaller sample size than the whole training dataset used for prediction and as such, its scores a lower performance on the test dataset than the whole training dataset does on the whole prediction dataset.

A similar trend is visible in the MAE for both CV and prediction results and in the percentage accuracy. These trends can be motivated on the same grounds as for the RMSE. The units of RMSE and MAE fall within the accuracy percentage range and validate the use of the training dataset for achieving accuracies between 0.09 and 0.21 seconds.

The observed trend in the BtT task is disrupted when turning to the results of the TtT task. The trend now follows RF < ADA < ERF < ST for the RMSE, MAE and accuracy percentage. This change can be motivated by the fact that STs tend to overfit the data, whereas ensemble methods work better with outliers and in reducing variance being therefore more flexible. Furthermore, the differences in RMSE and MAE values become amplified in the accuracy percentage as an increase of double that of the BtT task can be noticed.

In regards to the experiment research question, TtT reaching time can be predicted from a random forest model trained with BtT training data with an accuracy of around 0.17 seconds, whereas the model predicts BtT data with an accuracy of 0.08 seconds. This was deemed to be an acceptable limit of accuracy, as justified in later sections (see Section 5.1).

Chapter IV – Support Level Controller

This chapter will focus on evaluating the behavior of a designed support level controller in terms of adjusting the support level based on user input. Section 4.1 will start with an introduction to situate the reader in the context of research. After this first approach, section 4.2 will address the research methodology followed throughout the chapter, with an insight into the mechanics of the support level controller as well as presenting the methodology of Experiment II, an experiment designed to evaluate the behavior of the controller. Section 4.3 will then present the results from Experiment II and lastly, Section 4.4 will provide a preliminary discussion on the experimental results.

4.1 Introduction

This chapter aims at answering the second research sub-question on evaluating a proposed support level controller which adjusts the support level based on user input. It will do so by first presenting the proposed support level controller and then evaluating its behavior. This controller will be the model represented by the support level controller unit in the system architecture of Figure 1.

There are several reasons that motivate the addition of a support level controller to the system which adjusts the delivered support level. Most importantly, and as stated in the background chapter, repetitive task-oriented exercises in addition to patient motivation are regarded as effective rehabilitation techniques. By combining the motion of the robotic arm, a support level that promotes user participation and the motivational platform, the eNHANCE system will provide a favorable environment for rehabilitation.

Referring back to the system architecture in Figure 1, and taking into account the deeper level description of the robotic arm unit of Figure 4, a high-level conceptualization of the support level controller interaction can be depicted as shown in Figure 16:

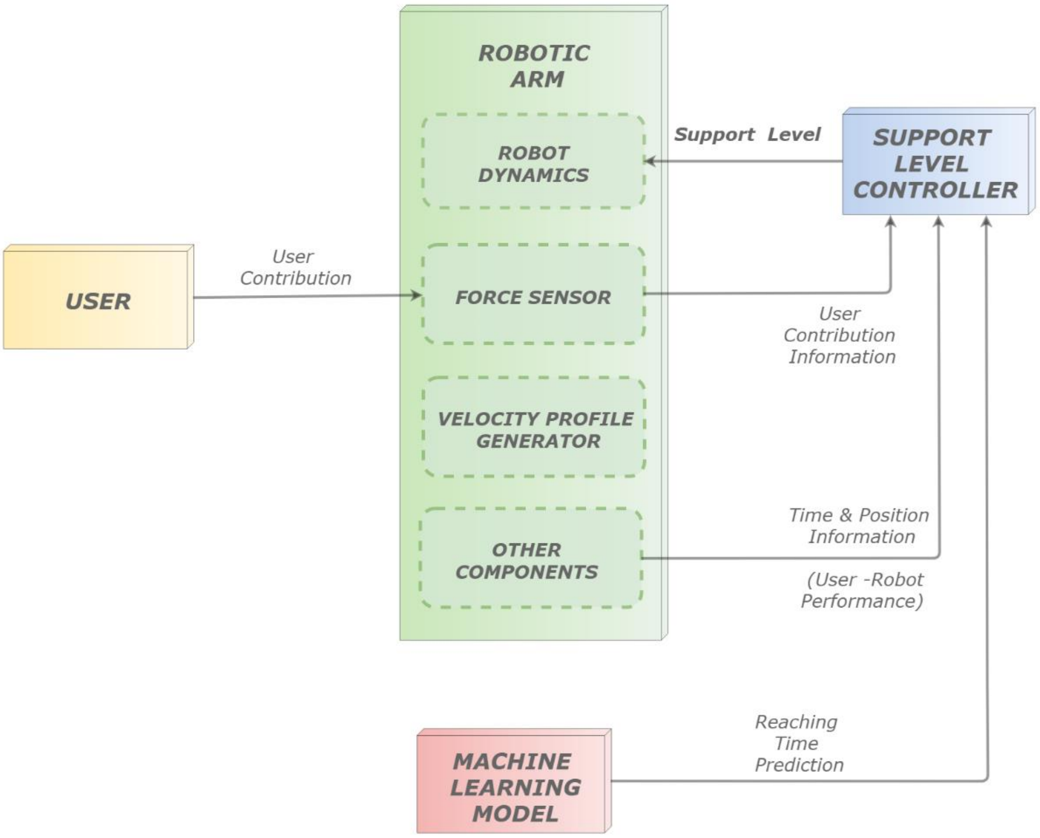


Figure 16 - High-level concept of support level controller action.

There are several remarks to be made about the support level controller action. First, it is important to mention that the interactions and units shown in the figure are only those regarding the support level controller action. The overall system architecture is composed of more units and parameters than those depicted in the figure above.

With that in mind, another remark is to notice that the robotic arm unit contains an additional subunit compared to Figure 4 in Section 3.1. Said subunit is the force sensor, which given its relevance in this chapter, has been separated from the other components subunit.

From the figure, one can distinguish on a higher level three inputs which go into the support level controller: user contribution information, time and position information and reaching time prediction. After the controller's action, the main output is the adjusted support level.

The user contribution is a system parameter which defines the engagement of the user in the user-robot interaction during a reaching task. This parameter is obtained from the force sensor within the robotic arm and later used as an input for the support level controller.

Another system parameter from Figure 1 explained in this section is the user-robot performance. This parameter will come from other sensors within the robotic arm, in the other components unit, which contain the encoders of the robotic arm. The main output information that will be conveyed to the support level controller will be that regarding reaching time and end position of the robotic arm and the user after a reaching task.

A final input to the support level controller is the predicted reaching time, from the machine learning model discussed in Chapter III. This prediction will serve as a reference value for comparison against the time information from the other components unit in the final control strategy (see Chapter V). For that reason, it will not be discussed in depth in this chapter.

The system parameters associated to user contribution and user-robot performance will be called from now on performance metrics. These performance metrics –time, position and user contribution from the force sensor- are measures that indicate how well the task has been performed. In the scope of this research, the performance metrics will indicate how successful a reaching task has been in the setting of the eNHANCE device.

In such a way, the support level controller unit has a specific set of rules that will determine the output support level, based on the inputs depicted in Figure 16. The set support level will ultimately be conveyed to the robot dynamics -as described in Section 2.3.2-, which will in turn make the robotic arm more compliant or less compliant to the user action by altering the spring stiffness.

To conclude to introduction to the support level controller, it is important to remark at this point that the action of the support level controller is carried out in discrete steps. After an initial movement, data on system performance metrics will be collected. These performance metrics are the user contribution and the time and position information. They will be used as inputs into the controller, along with information on previous support level and on predicted reaching time for the task at hand. The support level controller will then provide a set support level that will be used in the next reaching task. In such a way, for a user using the device, there will be several reaching movements with support level controller iterations taking into account previous performance metric data. This process is better represented in Figure 17.

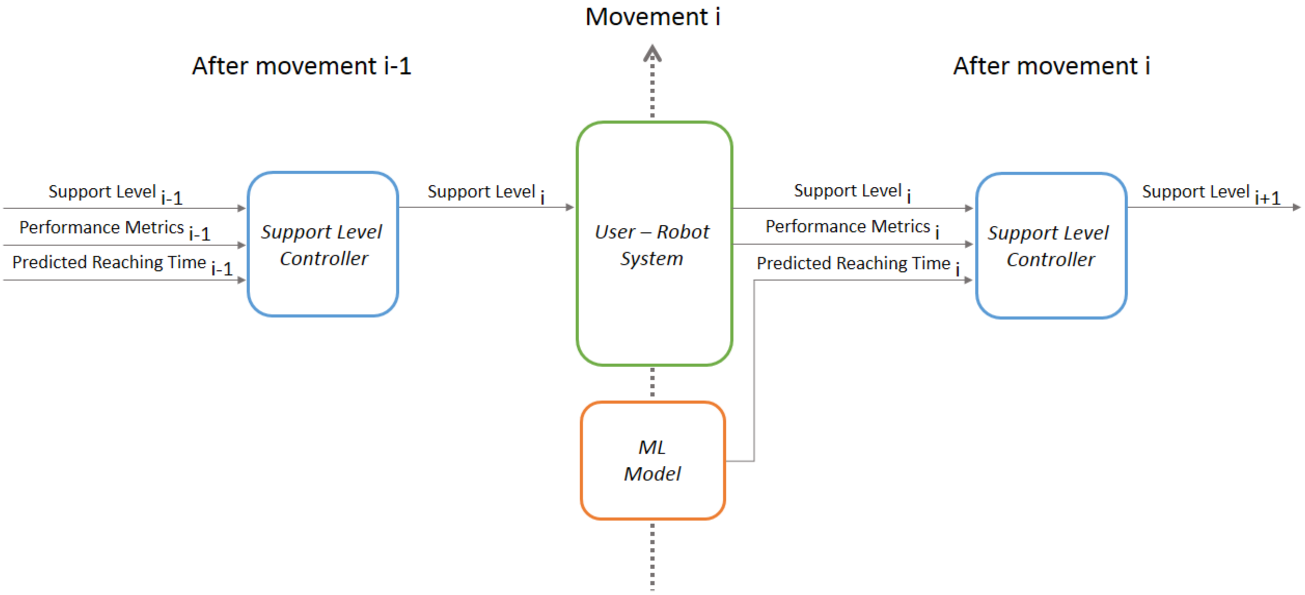


Figure 17 - Iterations of the support level controller action after movements.

4.2 Research Methodology

The purpose of this section will be to present the relevant methods and notions used to carry out the research in this chapter. To do so, there will be two subsections. Section 4.2.1 will provide an insight of the support level controller model, from a high level design concept to a specific model description. Next, Section 4.2.2 will cover the methodology followed in Experiment II, which aims to evaluate the behavior of the support level controller in an experimental setting.

4.2.1 Support Level Controller Model

This section will present to the reader the designed model of the support level controller. The model covers the relevant inputs, settings, and a set of rules that define the workflow of the support level controller. This will be done in a top down approach; from general concepts to specific definitions. First, there will be a focus on the design of the controller on a higher level. Then, a more precise model description will show the decision flowchart of the model used to reach a change in output support level, as well as showing an overall view of the controller.

Model Design

As stated in the introduction section, there are two types of performance metrics. The first one consists of the user-robot performance metrics -time and distance- which quantify the joint performance of the user and the robot in fulfilling the reaching task. The second one is a user performance metric, which will indicate user contribution during the reaching task. Based on these metrics, a design for model operation was created. The next paragraphs will address said design on a high abstraction level.

Let us start with the model design regarding the time performance metric. Three thresholds were proposed for the support level controller model: a predicted healthy user time threshold, an optimal time threshold, and an acceptable time threshold.

In order to assess whether a reaching motion was performed at an appropriate time, there needs to be a reference value. This is when the healthy user prediction reaching coming from a machine learning model time comes into play as an optimal reaching time reference. The prediction of this healthy user reaching time was the goal of Chapter III and will be the predicted healthy user time threshold. The optimal time threshold on the other hand, is a variable threshold which allows for some difference in time from the predicted healthy user time.

Furthermore, in a rehabilitation scenario the participant cannot be expected to always perform optimally. It was therefore reasonable to establish an acceptable time difference in reference to the optimal value, delimited by the acceptable time threshold.

With all the above in mind, the control design of reaching time was devised as depicted in Figure 18.

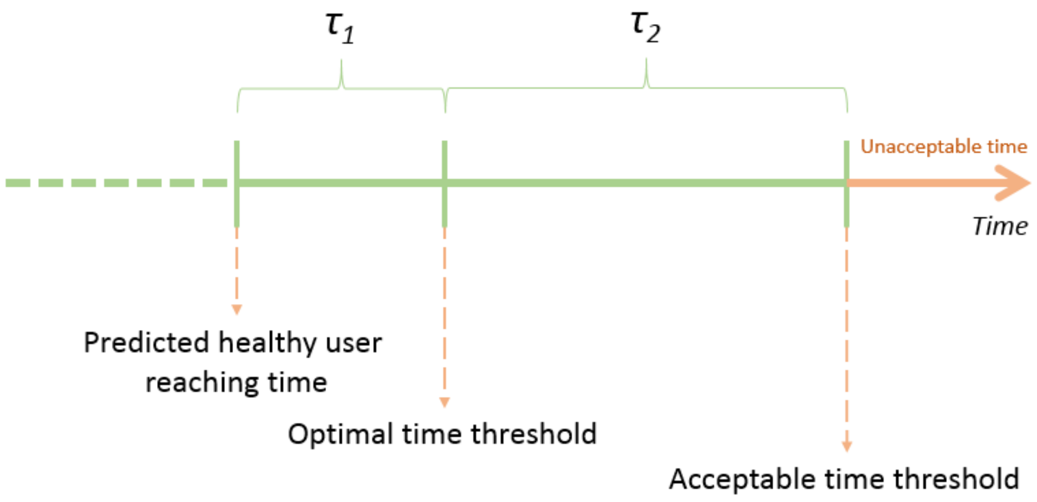

τ1
τ2
Unacceptable time
Time
Predicted healthy user reaching time
Optimal time threshold
Acceptable time threshold

Figure 18 - Time performance metric control design. τ_1 refers to the time interval of optimality whereas τ_2 is the acceptable time interval.

In such a way, there will be two time related settings in the support level controller that will compare the time performance metric to the reference reaching time: τ_1 and τ_2, the optimal time interval and the acceptable time interval, respectively.

It is worth mentioning that the robot itself has a time threshold that determines how long a movement will last before the robot action is stopped and data acquired. This threshold can be set to any value, however it is recommended and reasonable to set it somewhere along the unacceptable time but close to the acceptable time threshold.

Let us now focus on the distance performance metric. Given that there is a known target, once a reaching task has been completed there will be a difference between the end position of the robot and the know target: the distance error. Two areas can be defined for the distance error: an optimal vicinity and an acceptable vicinity. The optimal vicinity for reaching target performance is defined by the inherent healthy user distance error in combination to the robotic arm.

On the other hand and in a similar manner as with the time performance metric, an acceptable range of performance could be favorable to promote rehabilitation and to motivate the user to perform better. This range will be the acceptable vicinity. Figure 19 shows the concept from this line of thought.

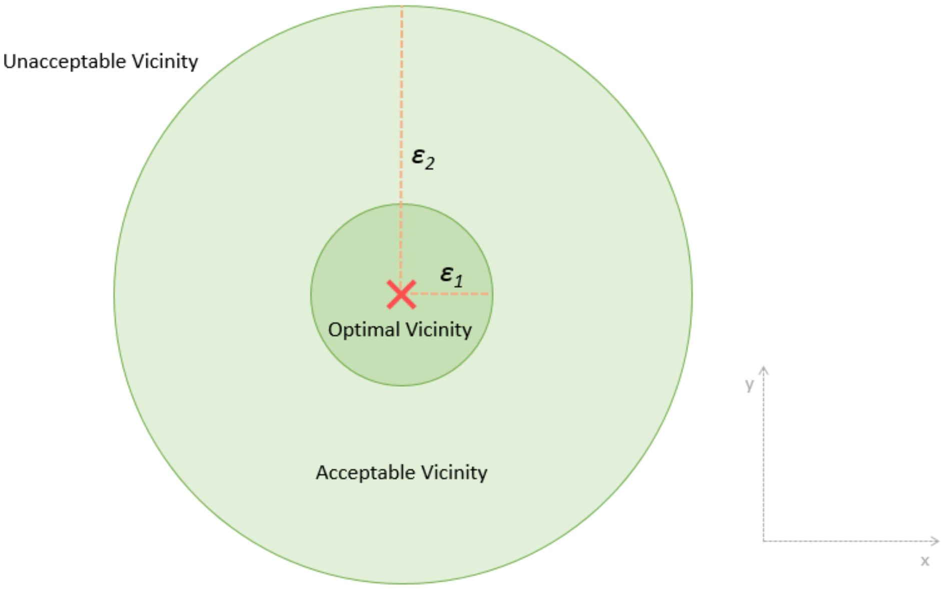


Figure 19 - Distance performance metric control design. The red cross in the middle of the figure represents the position target. ε1 refers to the Cartesian distance defining the optimal vicinity area whereas ε2 defines the acceptable vicinity area.

The user and the robotic arm will therefore end at a position within three different areas: the optimal vicinity, the acceptable vicinity or the unacceptable vicinity. There are thus two defining support level controller parameters which define these areas in terms of cartesian distance: ε_1 for the optimal area and ε_2 for the acceptable area.

Lastly, a strategy was designed for the user contribution performance. During a reaching movement at a specific support level, the force sensor acquires the interaction forces between the robotic arm and the user. These exerted forces will be converted –as described in Section 4.2.2- into the user contribution performance metric, referred from now on as interaction work .

A relationship between interaction work and support level can be extracted from experimental data. For healthy users, each of the interaction work values at a specific support level represent the minimum interaction work needed to reach ε_1 and τ_1, or in other words, to optimally reach the target. This minimum interaction work threshold for a specific support level i, will be referred to as $W_{th,I}$ (see Section 4.2.2 Figure 26).

The performance of the user at a certain support level can therefore be determined by comparing the user interaction work for a given movement against the threshold $W_{th,i}$. As with the other two performance metrics, an acceptable range can be set to account for variability in user performance. This range will be defined by subtracting a certain amount to $W_{th,i}$, resulting in a new acceptable threshold: $W_{margin,i}$.

Similarly to the distance and time metrics, there will thus be three main ranges of performance: enough interaction work, acceptable interaction work, and not enough interaction work. This division will be used in determining whether the user's contribution is enough to reach the target when the robotic arm is delivering a certain support level in comparison to a healthy user. Figure 20 illustrates the aforementioned.

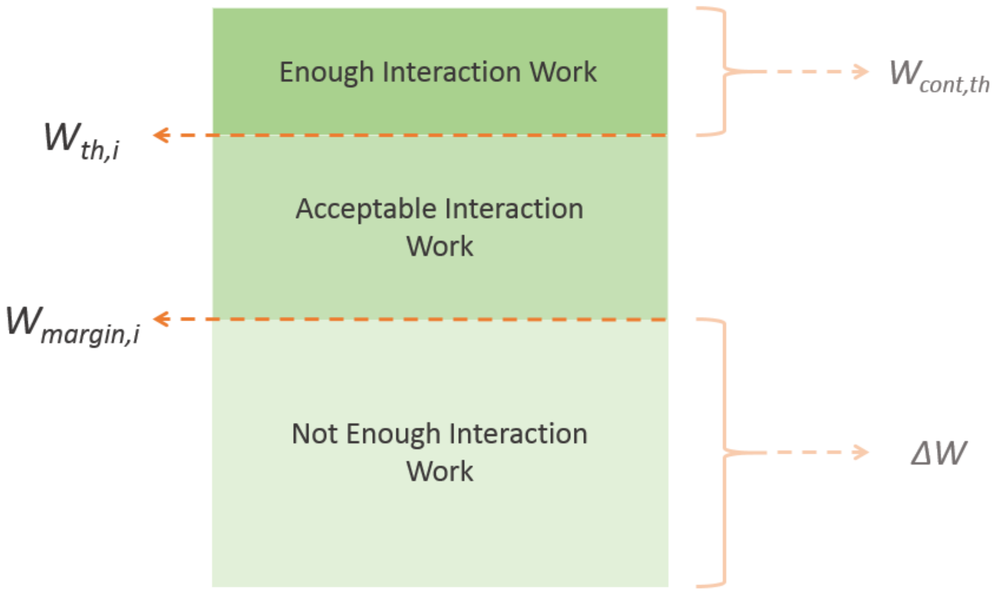


Figure 20 - Interaction work performance metric control design. There are three ranges in which the interaction work of the user may fall into for a reaching movement performed at a specific support level: enough interaction work –defined by $W_{th,i}$, -acceptable interaction work –determined by Wmargin- and not enough interaction work. There are two additional control parameters that will help in further determining the support level to be set: $W_{cont,th}$ and ΔW.

Not only does the support level need to be either increased, lowered or kept the same, depending on the contribution of the user, but the controller has to be able to account for the fact that the reaching task has to be successful to some degree as part of the assistance-as-needed functionality. This is adressed by two additional parameters: $W_{cont,th}$ and ΔW.

$W_{cont,th}$ or work count threshold is an upper threshold for an inner support level controller counter that will determine how many times the user has shown enough interaction work and thus reached acceptable limits. The lower the threshold is the more stress on motivation there will be, in other words, the support level will be decreased more often when the user shows enough interaction work and reaches the acceptable ranges. Similarly, the work count threshold can be modified depending on the capabilities of the user, so that the reaching task is accomplished by keeping the average support level at higher levels (see Appendix B.3).

Another possibility is the case in which the user does not input enough interaction work in relation to the acceptable threshold. There are two possible reasons for this: either the user is not able to input more force to reach acceptable limits or they are able to do so but not motivated. If the support level were to be immediately increased, there is no telling whether it is because of the first or the second reason.

In such a way the ΔW or work difference, is defined as the difference in the interaction work for the recently finished reaching movement and the interaction work of the previous movement. This work difference will be an indicator of whether the user is increasing their interaction work output between movements. When comparing ΔW for two successive movements, the support level controller determines if the user's contribution has increased compared to the previous movement or decreased. If it has increased, the support level will be kept the same as motivation, otherwise, it will be decreased as it means that the user cannot input enough forces to reach the threshold within acceptable limits.

To recapitulate and end this section, for each performance metric –time, distance and interaction work- there are some settings that determine three regions of performance: optimal performance, acceptable performance and unacceptable performance. Furthermore, in the case of interaction work, two additional parameters ensure that the motivational side of the controller –which is related to lower support- does not impede the fulfillment of the reaching task.

Model Description

This section will provide a final, more specific view of the support level controller model. First, a flowchart depicting the decision making for adjusting the support level will be presented. Secondly, a list will be shown with the description of the relevant parameters, inputs and outputs that play a role in the action of the support level controller. These can be separated in three categories: performance metric measurements, controller settings and model outputs.

For an iteration of the controller, the model will get certain performance metric measurements. These measurements are those which are variable between reaching movements and may be related to the performance metrics or to the controller parameters. The previous support level is one of such parameters but it is separated from the others to emphasize the change in support level. Besides these variable measurements, the controller settings are defined by the controller parameters, fixed throughout movement iterations and used as decision thresholds.

The code for the support level controller model was written and validated in MATLAB and later converted to Python for integration with the device and for evaluation during Experiment II (see Section 4.2.2). For the sake of clarity, a flowchart showing the decision-making process reaching a support level change is depicted in Figure 21, in which the performance metric measurements and the controller parameters are used as inputs.

In addition to the decisions, the criteria for each step are depicted below their respective decision text boxes. The decision criteria contain the parameters discussed in the model design paragraphs, along with hypothetical performance metrics: ε_i, τ_i and W_i.

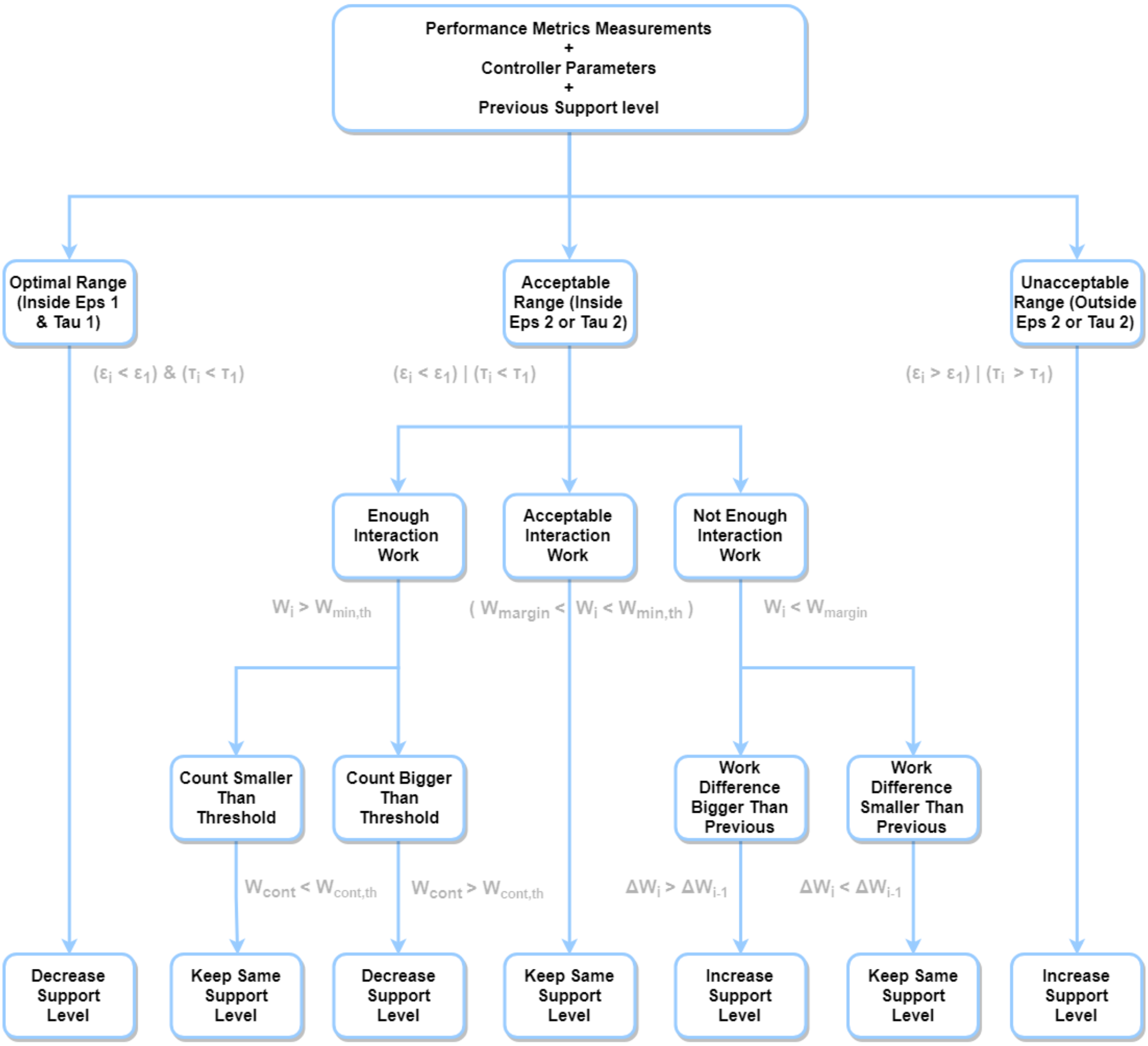


Figure 21 - Flowchart of the control strategy model. Given a set of performance metric measurements, support level controller parameters and a specific support level, the chart represents the decisions made so that a new support level is set.

A list with the notation for the support level controller parameter settings, the performance metric measurements and the model outputs are shown next:

- <u>Support level controller Parameter Settings</u>
  - ε_1: Optimal position vicinity distance.
  - ε_2: Acceptable position vicinity distance.
  - τ_1: Optimal time interval.
  - τ_2: Acceptable time interval.
  - $W_{cont,th}$: Enough counter threshold.
  - $W_{min,th}$: Interaction work – Support Level relationship.
  - W_{margin}: Margin of acceptable work.

- <u>Performance Metric Measurements</u>

    - Pos_{tar}: Position of the target to be reached relative to robot frame.
    - Pos_{prev}: Previous end position.
    - T_p: Predicted time to reach the target.
    - T_{prev}: Previous end time.
    - W_{prev}: Previous work input.
    - W_{cont}: Enough work counter.
    - SL_{prev}: Previous support level.
    - ΔW: Work difference between two consecutive movements.

It is worth noting that the definition of previous for the measurements used as model inputs is in reference to the fact that only once the movement has been finished does the support level controller act with the acquired data as seen in Figure 17. After the model has reached a decision following the flowchart, the outputs of the model will be:

- <u>Model Outputs</u>

    - Support level set: Support level to be used in the next movement.
    - ΔW: Updated work difference.
    - W_{cont}: Updated enough work counter.

Notice that these model outputs will be used as inputs in the next iteration of the support level controller action. With the information from the model design and description paragraphs, a final overview of the support level controller model can be depicted as in Figure 22:

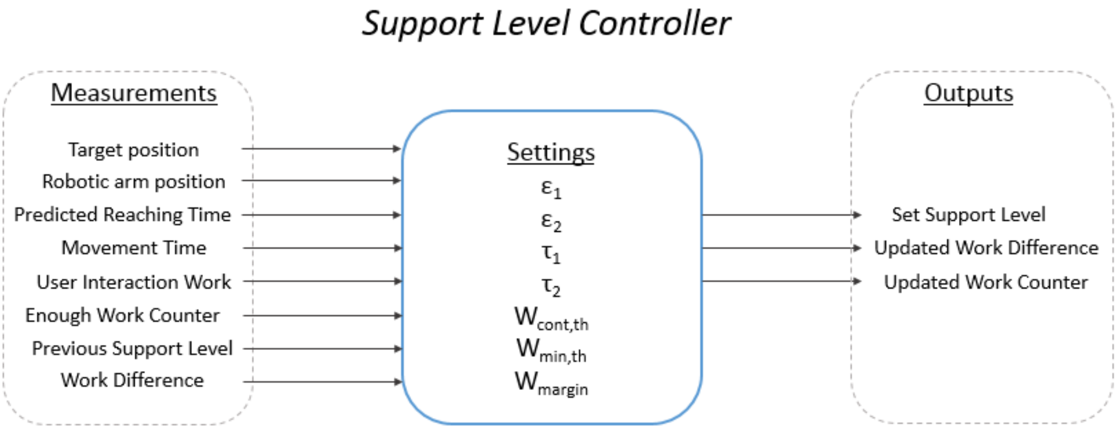


Figure 22 – Support Level Controller Overview. Overview of the inputs, settings and outputs that play a role in the Support Level Controller Model.

4.2.2 Experiment II – Support Level Controller Behavior

This section presents the methodology of an experiment designed to analyze the behavior of the support level controller presented in the previous sections. This methodology includes several paragraphs covering an introduction to the experiment, the recruitment, the measurements, the experimental protocol and the preprocessing of the obtained data.

- Introduction

The purpose of Experiment II is to analyze the experimental behavior of the designed support level controller when integrated into the robotic arm. The postulated support level control strategy model from Section 4.2.1 was converted to a Python script previous to the experiment and later implemented in the ROS communication system, so that a communication could be established between the robotic arm and the support level controller.

During the experiment, the performance metrics –namely distance, time and interaction work- were acquired during reaching tasks. In order to evaluate the behavior of the controller, it was tested against different scenarios to observe the change in support level and determine the extent to which the current support level controller can deal with different tasks.

Taking the above into consideration, the focus of the research was therefore aimed at answering the following question:

- What is the observed change in support level –i.e. the behavior- of the support level controller when used in combination with the robotic arm during reaching tasks in different scenarios?

The approach to answering this research question was to train participants in the use of the robotic arm and later instruct them on three different scenarios: a normal scenario, a fatigue scenario and a lazy scenario. Through the appropriate instructions, the participants simulated these scenarios whilst using the support level controller. The behavior of the controller in combination with the robotic arm was thus obtained for each scenario, with an additional participant survey on support level change perception.

- Recruitment

The target population of this study was adult, healthy individuals. Exclusion criteria included left-handed individuals and those individuals with lowered motor functions. The accessible population covered individuals studying or working at the University of Twente. A sample of 5 individuals was drawn from the accessible population.

- Measurements

The experiment was performed in an enclosed space at the University of Twente. The chair with the built-in arm support was used to acquire the data. A table with a target marked on it with tape, was placed in front of the seated user at an appropriate height.

The device was connected to a computer, where a display showed the relevant information regarding the outputs and inputs of the system, as well as the chosen settings. Data from the display was logged and saved into a file for later analysis. An overview of the experimental setup can be seen in Figure 23.

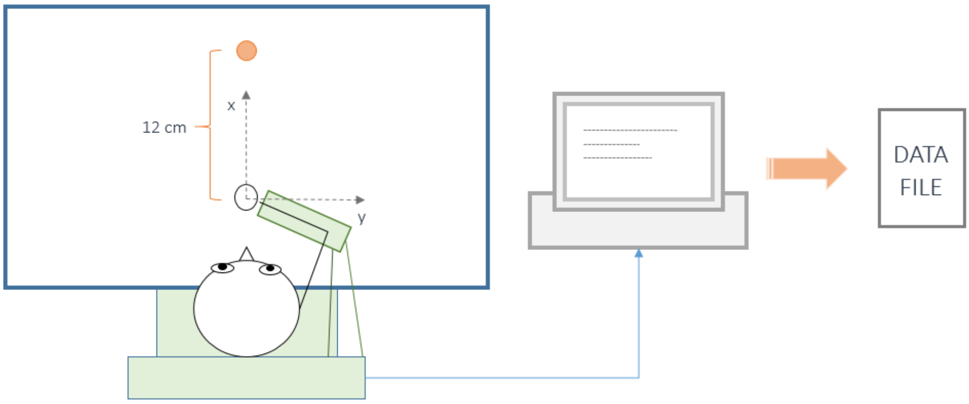


Figure 23 Experimental set-up of Experiment II. The target was marked on a table in front of the user, 12 cm away from the base position of the robotic arm. Data was then displayed on the computer and later logged into a file.

The position of the target remained constant throughout the experiment. It was set at 12 centimeters from the base position of the robot and aligned with the x-axis of the robot's reference frame. For the sake of clarity, a picture of the real-life experiment set-up is displayed next in Figure 24.

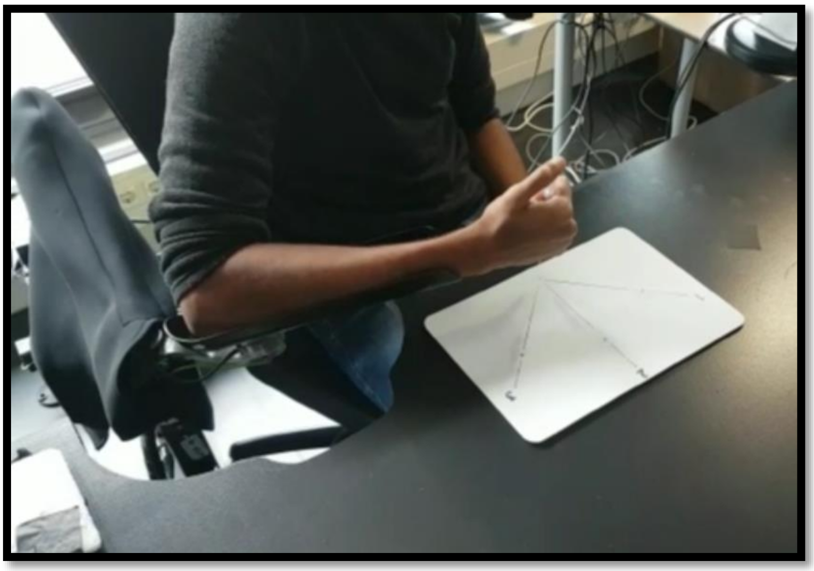


Figure 24 Real-Life Experimental set-up of Experiment II.

Regarding the support level, the possible support levels to be used during the experiment were, on a scale from 0 (zero support) to 1 (full support):

Support Level = [0, 0.25, 0.35, 0.45, 0.55, 0.65, 1]

The motivation behind having this specific set of 7 support levels comes from the perceived difference experienced by users in preliminary experiments (see Appendix B).

At the time, the following range of support levels was used: [0, 0.13, 0.25, 0.38, 0.5, 0.63, 0.75, 0.88, 1]. It was determined that for a range between 0.25 and 0.63, the support level felt more linear, whereas for 0 to 0.25 and 0.63 to 100, the difference between support levels was almost non-noticeable.

Furthermore, the selection of support levels is additionally supported by the support level vs. spring stiffness plot as shown in the next figure and based on Equation 1.

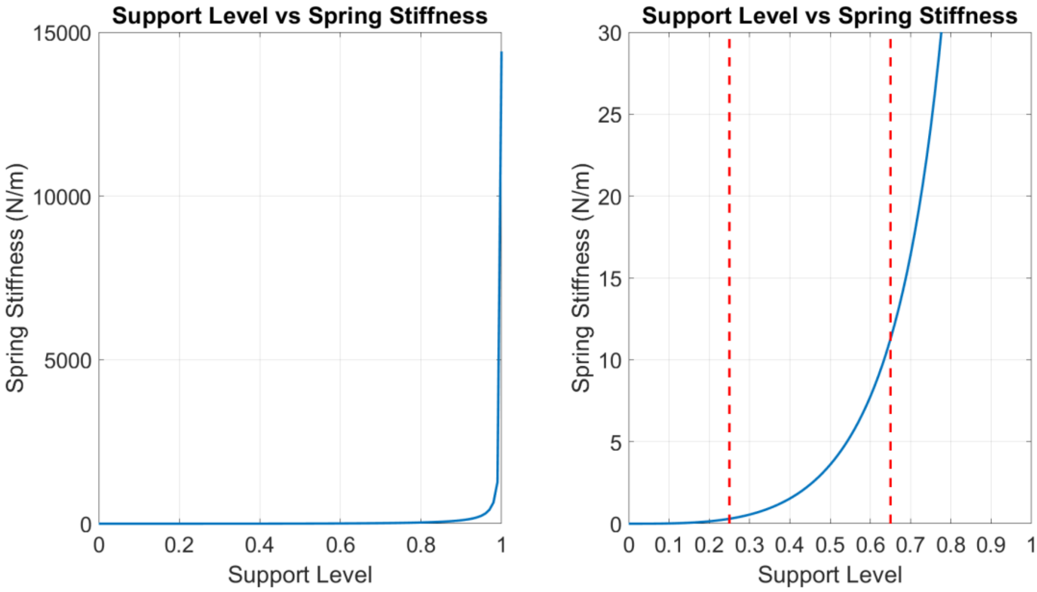


Figure 25 - Support Level vs. Spring Stiffness plot. Equation 1 has been plotted to show the values the spring stiffness takes for a specific support level. The plot on the left shows the relationship's boundaries whereas the plot on the right is a zoomed version in which the region with the most change in stiffness is seen (0.25 to 0.65).

Recalling Equation 1 for the sake of clarity:

$$K = \frac{SL^3 m_v \left(\frac{4a}{\pi v}\right)^2}{1.001 - SL}$$

Equation (5)

As quoted from Section 2.3.2: Equation 1 defines the value of the spring in terms of the support level. The nomenclature of the equation is as follows: *SL* is Support Level, m_v is the virtual mass value, *v* is the maximum velocity value and *a* is the maximum acceleration value. All units are in SI.

From Figure 25 and based on Equation 5, it can be observed that the biggest change in stiffness is found for support levels in between the 0.2 to 0.8 range. This agrees with the aforementioned user perception of support level.

With the above in mind, the 0.25 – 0.65 range was expanded to be further investigated, while keeping 0 and 1 as no support and full support respectively.

- Experimental protocol

Before the start of the experiment, the individuals that agreed to participate in the experiment were asked to sign the informed consent form and revise the information brochure. After consent was given and a participant number assigned, the individual was asked to sit on a chair. They were briefed about the experiment procedure once again by the researcher to make the process run smoothly.

In order to keep the experimental conditions as invariable as possible, the participant was instructed to adjust the chair at a comfortable height, with their bellybutton aligned with the target axis and the back adequately leaned against the chair.

The length of the upper and lower arm of the participants was measured before the procedure began. As explained in Section 2.3, this is to ensure a proper control for the robotic arm of the elbow angle and to make the movement more comfortable for the user.

After this, the experiment began. It was split up in three phases. The first two did not require the support level controller and were aimed as preparation for the use of the device and the support level controller. The last phase was designed to answer the research question by testing the support level controller in three different scenarios.

1. **<u>Training phase</u>**

This phase addresses the training period required for the user to get familiarized with the device. Given the complex interaction and behavior of the robot in coordination with the user, there is a learning curve when carrying out specific tasks and the user needs to learn how to deal with the behavior of the robot.

In order to train the user, 3 movements were carried out at each of the support levels mentioned above, in a randomized order. This randomization was included in the ROS environment. The instructions given to the participant during this phase were the following:

- The goal is to reach the target.
- Feel the contribution of the robot and supply the necessary force.
- Try to make the whole movement as smooth as possible, but accurate.
- Do not push or pull too hard, let the robot guide you.

2. Work –Support level acquisition phase

Once the participant has been trained in the use of the robot, the next step was to determine the contribution of the user in terms of input work at each support level.

This would allow the support level controller to set a boundary of acceptable work input and will, after a certain amount of iterations, provide a support level in accordance with the performance of the user. An example of Work vs. Support level relationship graph is shown in Figure 26.

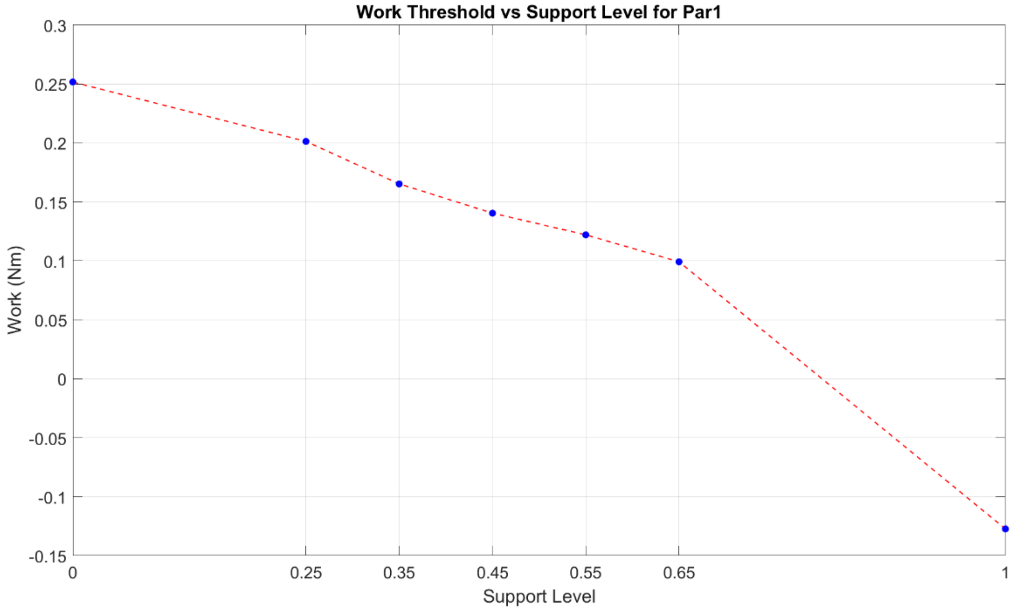


Figure 26 - Example of Work (W) vs. Support Level (SL) relationship. The boundaries of acceptable work input are considered by the support level controller in addition to other constraints to find an appropriate output SL given a specific work input.

Four movements were carried out at each support level in order from 1 to 0. The researcher had to compute the average of the input work for a certain support level and to input the chosen boundaries into the support level controller. The instructions given to the participant will be the following:

- The goal is to reach the target.
- Feel the contribution of the robot and supply the necessary force.
- Try to make the whole movement as smooth as possible but accurate.
- Do not push or pull too hard, let the robot guide you.

3. Scenario phase

This was the phase in which relevant data for answering the research question was gathered. It consisted of 3 different scenarios where the participant had to reach for the target given a specific set of instructions. The support level controller was active during this phase and it will output a certain support level based on the user performance metrics – interaction work, time and distance accuracy-, and on intrinsic control parameters.

Based on the control strategy model in Section 4.2.1, the support level controller inputs can be categorized into preset inputs -same for all participants-, user-specific inputs and experimental inputs -updated during experiment-. The preset inputs and user-specific inputs were set as follows:

- Preset inputs
    - ε_1: Optimal position vicinity. Set as 1 cm.
    - ε_2: Acceptable position vicinity. Set as 4 cm.
    - τ_1: Optimal time interval. Set as 1.25 s.
    - τ_1: Acceptable time interval. Set as 1.5 s.
    - $W_{cont,th}$: Convergence factor. Set to 2.
    - Pos_{tar}: Position relative to robot reference frame of target to be reached. Set at [0.69, 0.35] (x and y coordinates, respectively).

- User-specific inputs
    - $W_{min,th}$: User W vs. SL relationship. From 2nd phase.
    - W_{margin}: Set to 0.02. (see Appendix B.3)

For each scenario, around 20 movements were logged. This number of iterations was thought to give the support level controller enough time to converge around a certain output support level. After each movement, the user was asked to estimate the perceived support level difference in terms of same, higher or lower. This perception was be written down by the researcher after every movement in the perceived support level document.

There was a set of instructions that was given to the participant across all scenarios, namely:

- Feel the contribution of the robot and supply the necessary force.
- Try to make the whole movement as smooth as possible.
- Do not push or pull too hard, let the robot guide you.

For each scenario there was an additional set of instructions to be followed. The three scenarios were:

a. Normal scenario (10 min)

In this scenario the participant had to behave normally, trying to reach the target at their regular healthy performance. The expected support level controller behavior is a convergence of the mean support level towards zero support level. This scenario will check if the support level controller's response is comparable to the expected response. After each movement, the perceived support level difference was collected.

The specific instructions given to the participant are:

- The goal is to reach the target performing normally and accurately.

b. Fatigue scenario

In this scenario the user was instructed to output an approximate constant work output of that of half their maximum output (zero-level support), to resemble fatigue. This will give information on the behavior of the support level controller when an unknown limit support level is to be reached. Feedback from the researcher to the user may be possible in order to fluctuate around a specific work output. After each movement, the perceived support level difference was collected.

The specific instructions given to the participant were:

- The goal is to constantly perform at half your estimated maximum performance.
- Find a balance between the force you use and the perceived support level.
- There will be a moment where your performance is not sufficient to reach the target, which is to be expected.
- You should still try to reach the target once at your half performance, but it does not matter if you fail every once in a while.

c. Lazy scenario

During the lazy scenario the user had to underperform on purpose in terms of reaching the target. In terms of work output, the user is still capable of reaching the target but relays too much on the robot action. The expected support level controller behavior was to lower the support level and give feedback to increase user participation.

The specific instructions given to the participant were:

- The goal is to "trick" the robot into working more than you do.
- Try to find a balance between minimum effort you have to do and reaching the target.
- You should however do as the robot tells you in the feedback, if it motivates you, you have to input more force.

- Data acquisition

The procedure above was repeated for each of the five participants. After sitting in the chair, the arm length was input in a GUI screen on the computer. The robotic arm was then homed to the base position and the training phase began.

Two data files were acquired for each phase of the experiment. One of them was a text file with information on work input, distance error, time error, set support level and a feedback message from the support level controller. The other file included raw data on forces, trajectory and time, as well as other additional information. These files were acquired for every phase and scenario of the experimental procedure.

Some notable events that affected the measurements were noticed during the acquisition of data.

One of them was the limit flag event. A limit flag was raised when the motor was out of boundaries and therefore the robotic arm could not continue the motion. This usually happened close to the homing position and thus the distance and time errors during these movements can be considered artifacts. After a limit flag is raised, homing was required to resume the task. After a limit flag, the program assumes that another movement is needed so a repetition of the movement is done. In such a way, there are always at least 20 reaching tasks in the data without a limit flag.

Another remarkable phenomenon was the fact that the virtual position of the robot laid further than the physical edge of the robotic arm, somewhere along the hand of the user. The training phase helped the participants to take this fact into account while getting used to the behavior of the robot. However, given the duration of the experiment and the different support levels, the user perceived target position was sometimes biased.

To account for this, sometimes in between phases of the experiment a full support level was set so that the user could see where the virtual position and target position of the robot laid. This would ensure that the distance error was kept within the usual performance of the user-robot interaction across phases.

- Data Processing

Minimal data processing was involved in obtaining analyzable results from the log files. This was due to the fact that during the implementation of the support level controller script in the robot environment, the parameters of interest were already calculated and logged during the execution of the program. In such a way the distance error and time error were automatically calculated.

Interaction work calculation was as well automatized, however and for the sake of clarity, it is of importance to define how the interaction work was calculated.

For a given movement, both the trajectory -x and y-axis displacement- and the forces were recorded at any given sample time. Work can be defined by the integral a variable force along a path as seen in Equation 6.

$$W_x = \int_a^b F(x)dx$$

Equation (6)

Since the data are vectors of the sampled force and displacement difference, the implementation in MATLAB code (see Equation 7) to obtain the corresponding work can be in this case the dot product of both vectors:

$$W_x = dot(F_x, \Delta x)$$

Equation (7)

Once this calculation was done, the total interaction work (in Equation 8) was calculated as the sum of the work present in both x and y components:

$$W_{int,tot} = W_x + W_y$$

Equation (8)

With this calculation, a single performance metric that indicated the contribution of the user to the movement could be obtained. A negative total interaction work can be present under the circumstance that the robot is displacing the user's arm towards the target but the user is applying force –intentionally or unintentionally- in the opposite direction, whereas positive work will happen when the force applied by the user is in the direction of the movement.

It is important to remark that the forces that the sensor records are local interaction forces along the x and y-axis of the robot, as depicted in Figure 23. Local meaning that the forces are recorded at the point of contact between the user's arm and the robotic arm where the sensor is located, at the proximal end of the arm support (see Figure 3). For this experiment, the local forces were transformed into the forces that would be experienced at the endpoint of the robot.

Another important remark is to notice the nomenclature of interaction work used in the report instead of work. Given that not all of the trajectory comes from the read interaction forces, calling the calculation work should be taken with a grain of salt. Instead, a proposed nomenclature for the calculation was interaction work, sometimes referred to as work throughout the report.

Moving on to other data processing methods, the limit flag movements were kept since otherwise the change in support level in the support level controller's behavior is not representative of the actual support level progression. For this reason, the number of movements in the figures may be bigger than 20, which is the set limit for successful movements.

The magnitude in distance error of the limit flag artifact was however reduced to 4 cm in the training phase figures for a better representation of the phase, as can be noted in the next section.

A similar event happened for the time metric. The robot estimates the time it should take for the arm to reach the target based on the velocity profile. If that limit is reached, a value of two seconds is returned in the data.

Besides the aforementioned, the perceived support level had to be manually transcribed into a text file for processing in MATLAB. This file consisted of two columns, one with the actual support level and the other with a logical assignment, which was 1 for successful support level perceptions and 0 otherwise. With such a data file, the calculation of the accuracy on perceived support level changes across participants (Equation 9) and for each support level (Equation 10) was as follows:

$$Accuracy\ of\ participant\ X\ =\frac{Number\ of\ correctly\ perceived\ changes}{Total\ number\ of\ support\ level\ changes}*100$$

Equation (9)

$$Accuracy\ SL\ X\ =\frac{\begin{matrix}Number\ of\ correctly\ perceived\ changes\ towards\\ SL\ X\ across\ all\ participants\end{matrix}}{\begin{matrix}Total\ number\ of\ changes\ towards\\ SL\ X\ across\ all\ participants\end{matrix}}*100$$

Equation (10)

4.3 Results

The results on support level controller behavior and perceived support level are presented in this section after the data was correspondingly acquired and processed. For the sake of briefness only selected figures are presented in this section; those that are representative and show points of interest. The rest of the results can be seen in Appendix B.4.

Firstly, let us consider the training phase for analysis. Figures 27 and 28 show the distance and time errors for participant 1 and participant 3, respectively. The reason behind showing the results from these specific participants is that their training phase showed the clearest training based on the distance and time errors, for reasonable support levels – not always in the high range-.

In such a way, both figures contain information on the distance error and time error that each participant presented in the training phase during the 20 movements.

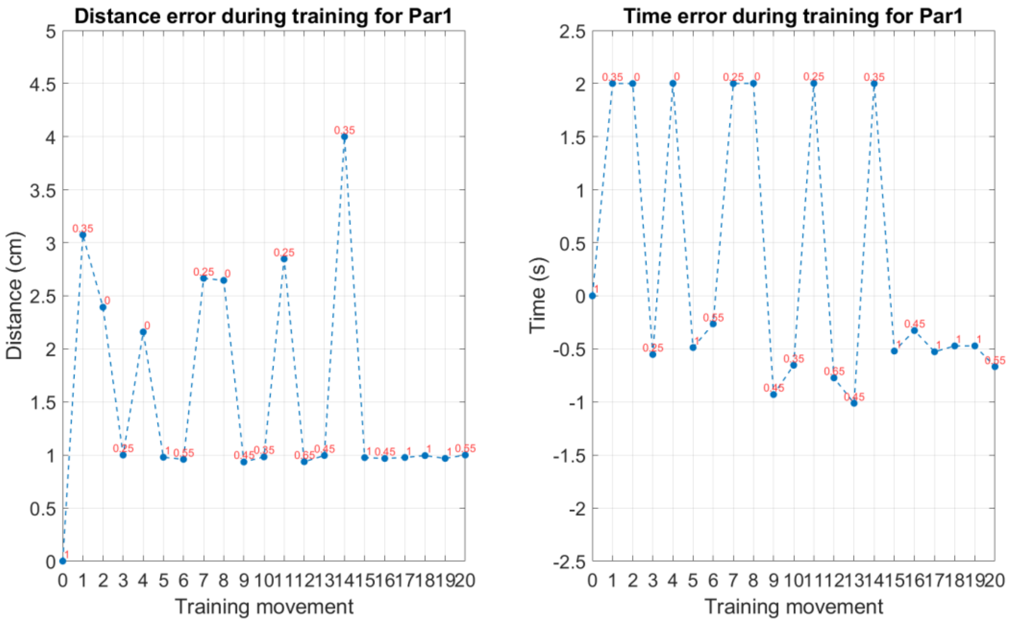


Figure 27 – Distance and Time error plots for Participant 1. Distance error plot (on the left) and Time error plot (right) for participant 1 during the training phase. The support level is specified in red for each training movement.

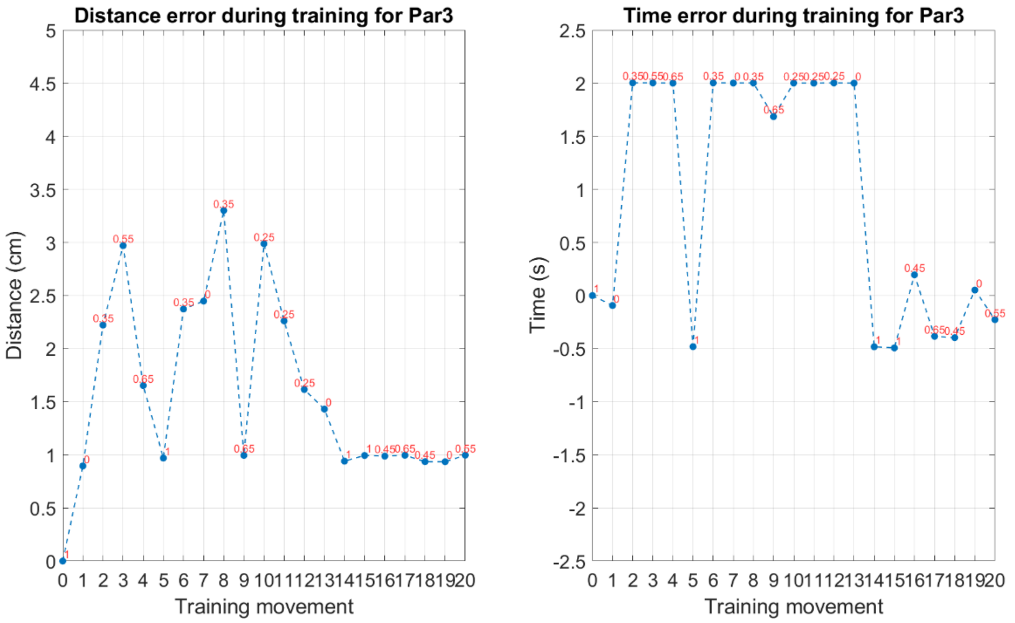


Figure 28 – Distance and Time error plots for Participant 3. Distance error plot (on the left) and Time error plot (right) for participant 3 during the training phase.

It can be seen that for the first 14 to 15 movements, the error oscillates between one and three centimeters for each participant. Furthermore, the robot showed a limit flag for participant 1 at movement 14, where the distance error is 4 centimeters. After the first 14 to 15 movements, the distance error of the participants both stabilized at around 1 centimeter from the actual target position. In participant 1 this result is ambiguous since the support levels for those movements are high. However, this was not the case for participant 3, as seen in the corresponding figure.

A similar trend appears when looking at the time error plots. As previously mentioned, a value of 2 seconds means that the participant did not reach the target in the expected time. It can be observed that for the 14 to 15 first movements, the time limit was reached multiple times for both participants. After those events however, the time error was more stable and around -0.5 seconds the estimated time. This negative value indicated that the user and the robotic arm reached the optimal vicinity earlier than the expected reaching time from the robotic arm.

The threshold phase will be analyzed next. Given that there are 5 different profiles, all of them have been plotted in the same figure for briefness and better comparison. In addition, a first order regression model with the data from all participants was used to get a possible threshold to be used in future research, so as to not have to carry out the threshold phase for all participants. Figure 29 shows the results of the mean work calculated at each support level for each participant, as well as the regression line:

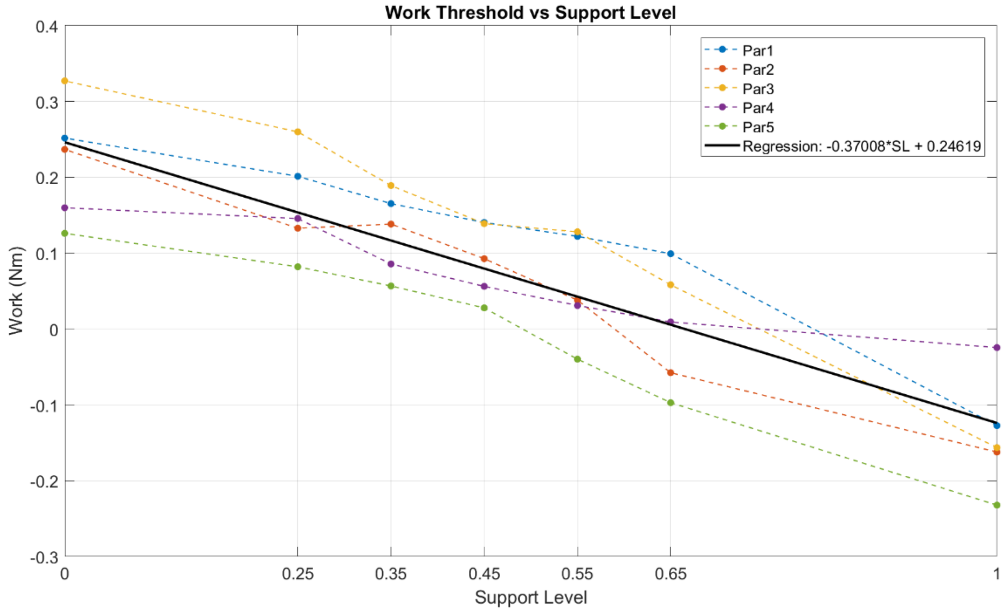


Figure 29 - Work threshold plotted against the different support levels for each participant. The dots represent the mean interaction work calculated from each participant at each support level. In addition, a regression line has been plotted which was generated from a first-order model with all participant data.

It is discernible from the figure that there is a general trend for all participants: the lower the support, the more work input it was used to reach the target. However, there are eventual cases in which the average work for a support level is bigger than the work for the adjacent lower support level as seen in participant 2 at support levels 0.25 and 0.35.

Although a general trend is followed, the main difference across participants is the amount of work exerted on the robotic arm. The biggest difference can be observed between participant 3 and 5. How relevant this difference is when designing the support level controller remains open to discussion.

As a last remark, it can be seen that the generated regression line follows the same trend as the participant data and appears to be a good estimate of the mean work values from participants at each support level.

Following the training and threshold phases, an analysis on the scenario phase will be laid out next. Two figures – Figure 30 and Figure 31 from participants 2 and 4, respectively- will be shown as demonstrative tools to represent the support level controller's behavior to the different scenarios. The results from participants 2 and 4 are shown next given that they displayed the biggest difference in support level controller behavior and in force profiles. Therefore, a more extensive discussion can be drawn from the results of these two participants.

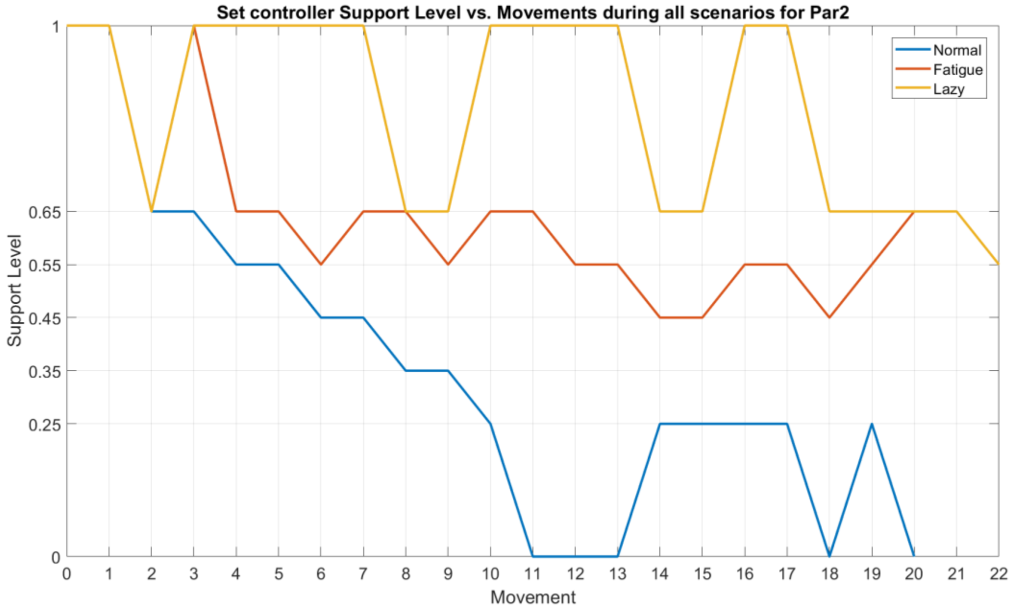


Figure 30 - Support level controller's behavior during all scenarios for participant 2. The set support level vs. the movement number is plotted for each of the three scenarios: Normal (Blue), Fatigue (Red) and Lazy (Yellow).

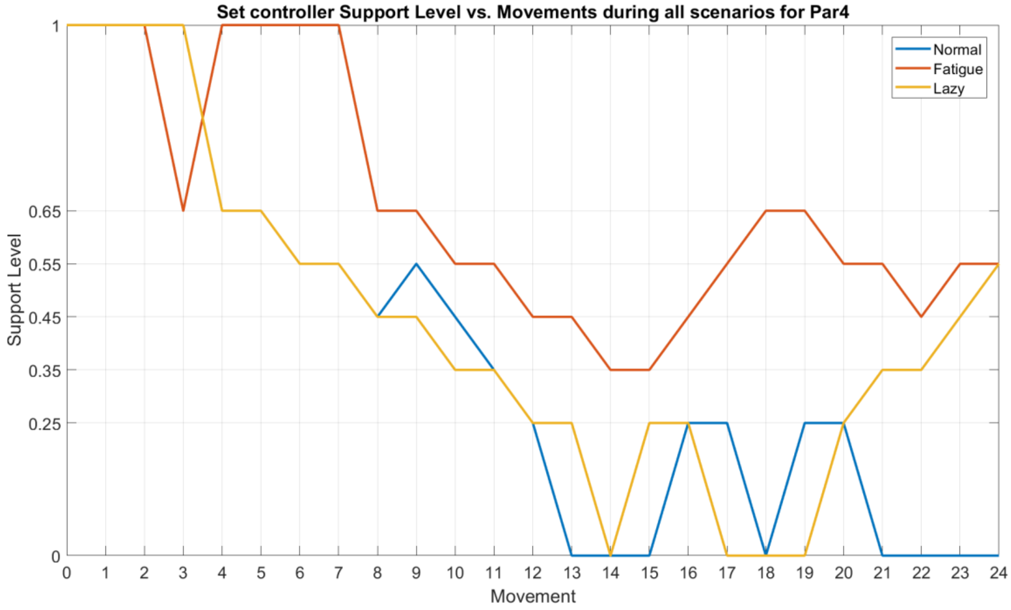


Figure 31 - Support level controller's behavior during all scenarios for participant 4. The set support level vs. the movement number is plotted for each of the three scenarios: Normal (Blue), Fatigue (Red) and Lazy (Yellow).

The expected outcome during the normal scenario was convergence towards support level zero. It can be observed in both figures that this was indeed the case, participant 2 reaching zero support level at the 11th movement and participant 4 at the 13th. In some occasions during the following movements, the support level was set to 0.25 and fluctuated between those two values.

During the fatigued scenario, there was an initial decrease in support level for both participants. After a certain number of movements, the support level stopped decreasing and oscillated around the middle range of support levels for both participants.

The lazy scenario shows the most differences between participants. In participant 2, the support level did not decrease further than 0.65 until the second to last movement, whereas in participant 4, the support level reached 0 at movement 14 only to increase to the mid-range of support after movement 19. This difference will be discussed later in the report (see Section 4.4 and Section 5.2.1).

Another point of research regarding the fatigue and lazy scenario differences is the force profile of the participants over time. For the sake of comparison, Figure 32 shows the force profiles of both participants 2 and 4, in accordance with the results shown in the previous figures:

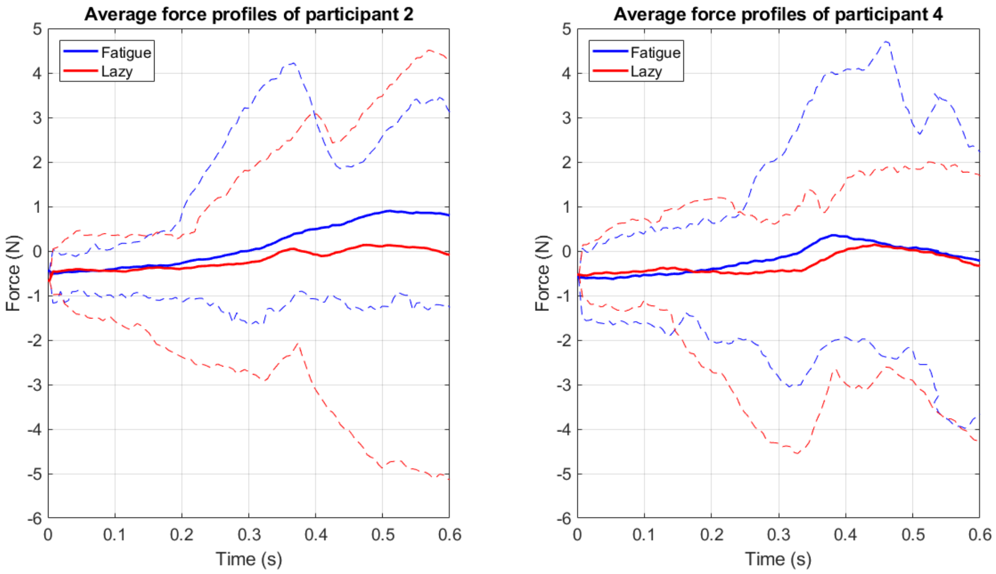


Figure 32 – Force profiles or Participants 2 and 4. Figure with the plotted force profiles of participant 2 (left) and participant 4 (right). The average force profile is across all movements and support levels. Both averaged force profiles from the fatigue and lazy scenarios are displayed along with their maximum and minimum values, represented by the dotted lines.

The mean of the forces exerted by participants 2 and 4 during the scenario phase is represented for both the fatigue and scenario phases in Figure 32. Each scenario additionally shows the maximum and minimum force values for a given time across movements.

With that in mind, the main point of interest revolves around the differences in scenarios for each participant. In participant 4 it can be seen that the lazy force profile increases in magnitude later than the fatigue profile. This effect is less obvious in participant 2 but still noticeable. Further discussion on this will be done in the next section of this chapter (Section 4.4).

Let us finally end the results section presenting the perception in support level. The following figures will show the results of such support level perception, where the calculation of both types of accuracy was performed as explained in the data processing section.

Figure 33 thus gives information about the total accuracy of each participant in determining the change in support level.

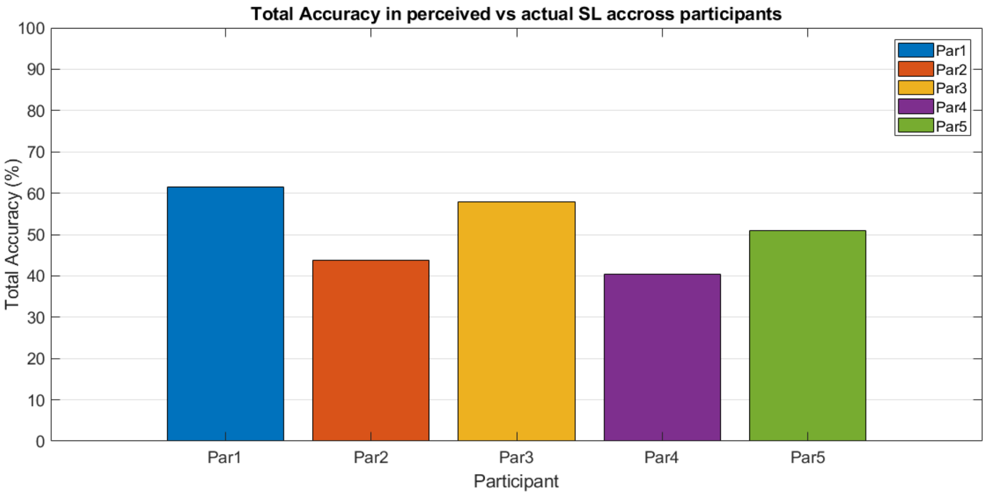


Figure 33 - Accuracy for each participant in determining changes in support level.

The total accuracy in determining the change in support level varies across participants, ranging from around 40% for participant 4 to 60% for participant 1. The mean accuracy is thus of around 50%, meaning an estimated one out of two perceived changes was correctly perceived.

Perhaps of more interest are the results shown in Figure 34, where the accuracy for change towards a specific support level is represented. This includes data from all users on whether the support level was lower, the same or higher than the previous one. In such a way for, say support level 0.45, there are three possible changes: coming from 0.35, 0.45 or 0.55. In such a way, the single accuracy measure in 0.45 contains information on correct guesses from these three changes.

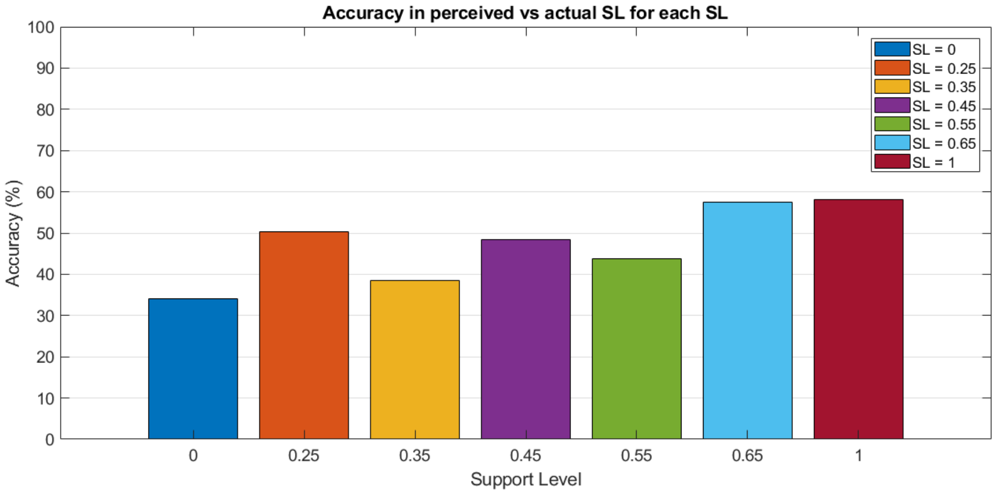
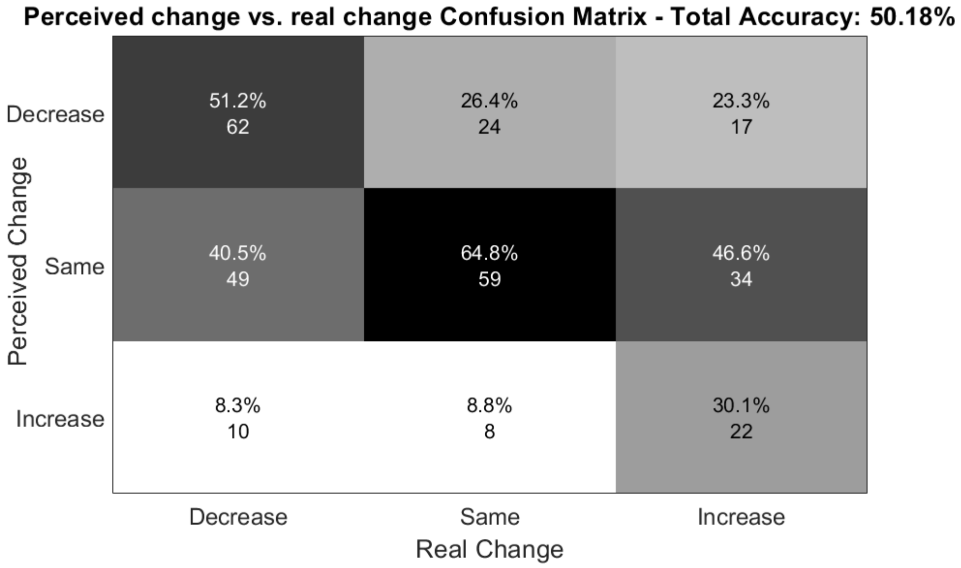


Figure 34 - Accuracy in determining the change towards a specific support level with data from all participants.

The figure shows that the highest accuracy was in those changes leading to support level 1 with a rate of almost 60% correct guesses, whereas the lowest accuracy was encountered when changing to support level 0 with 35%. The second highest was support level 0.65 with a little less accuracy than support level 1. For the other support levels the accuracy ranged from around 40% to 50%.

As a last result, a confusion matrix with the perceived and actual changes was obtained, as seen in Figure 35:



	Decrease	Same	Increase
Decrease	51.2% 62	26.4% 24	23.3% 17
Same	40.5% 49	64.8% 59	46.6% 34
Increase	8.3% 10	8.8% 8	30.1% 22

Figure 35 – Confusion Matrix derived from data from all participants.

The confusion matrix shows that the perceived changes have a total accuracy of 50.18%, which is in accordance with the previous results in the perception of support level. Furthermore, and in order of higher accuracy, the same support levels were better perceived with an accuracy of 64.8%, followed by the decreases which show an accuracy of 51.2%. The worst change in perception was found for increases in the support level, with an accuracy of 30.1%.

4.4 Preliminary Discussion

This section will discuss the results from Section 4.3, as a first interpretative approach. In order to do so, the results from the three phases –training, threshold acquisition and scenario- will be addressed in that order. Moreover, the scenario phase analysis will include an interpretation of the results from the normal scenario, the fatigue scenario and the lazy scenario.

Regarding the training phase, a stabilization of both the distance error and time error can be observed at around 15 movements. This phenomenon is not only present in the data shown during Section 4.3, but across other participant data (see Appendix B.4). It could be therefore reasonable to say that for the average user to get used to the device, a minimum of 15 movements is necessary. Furthermore, it appears that the natural user-robot distance error is of around 1 centimeter from the target position. This could be a useful specification for the optimal area ε_1 of the support level controller.

In contrast, the time error for successful and accurate movements –omitting limit flags- is usually negative and around -0.5 seconds, meaning that the user action makes the robotic arm travel faster than what the robot expects is a reasonable time to reach the target. This predicted time from the robot will, in future implementations of the controller, be replaced by the predicted time from the machine learning algorithm as stated in Chapter III.

Let us now discuss the results from the threshold phase. The work threshold versus support level relationship seems to follow the trend of more interaction work at lower support level. In such a way, the regression line presented in the results is a good candidate for use in further research so as to avoid repeating the threshold phase for every user. This is thought to hold true at least for healthy participants as they can exert much more force than what is necessary for the reaching task. The extent to which the work threshold regression is fit for use in the support level control applied to stroke users is a future line of research work.

The last discussion regarding the behavior of the support level controller will be concerned with the scenario phase. The normal scenario results showed the expected support level change across participants, reaching support level zero before the 20th movement. It should be noted that this might only be the case in healthy participants, given their work threshold relationship. In case of a stroke patient, the support level controller with a healthy work threshold could converge towards the mid-range of support level values, given that they might not be able to exert as much force as healthy users to accurately reach the target.

In the fatigue scenario, the participants were instructed to use half of the estimated performance they used the normal scenario for reaching the target at zero support level. It is of importance to say at this point that the task instructions were found lacking specificity for the majority of the users, so a reconsideration of the instructions may have to be drawn up in future work. Most participants seemed however to carry out the task better than they perceived, with the support level controller oscillating around a mid-range support level.

Another point of interest was the question of whether the support level controller could differentiate between a fatigue scenario and a lazy scenario. The results in Section 4.3 are inconclusive on this regard, given that the motivational feedback was not appropriately delivered.

This is due to the fact that motivational messages were displayed on the screen based on the action of the support level, and they were read out loud by the researcher to the participant mid-movement. The time to react to the motivational message was in some cases insufficient –as might be the case for participant 2-, whereas it seemed sufficient in others –for example, in participant 4-.

The results in the force profiles could relate to the results in the support level controller behavior for participants 2 and 4. The magnitude of the force in the lazy scenario for participant 2 is much lower than that of the fatigue scenario, which could motivate why the behavior of the support level controller stayed within high support levels. Participant 4 on the other hand showed a better response to motivational messages and a better understanding of the lazy scenario. The force profile from the lazy scenario for participant 4 shows a clear difference in the time at which the forces were exerted, with an increase in magnitude at a later time when compared to the forces during the fatigued scenario. The force profile should therefore be considered as a potential source of information to distinguish between whether a user is being lazy or is fatigued.

Besides the behavior of the support level controller, the perception of the participants regarding a change in support level provided an additional investigation point. As shown in the results section, participants got on average one out of two changes in support level right. During the experiments, all participants reported difficulties in assessing whether the support level was higher, lower or the same as the previous support level, with a more confident attitude when going from full support level 1 to support level 0.65 or vice versa. This might motivate the results, in which support levels 1 and 0.65 were the most and second most accurate support levels, respectively. Furthermore, zero support level was the hardest one to estimate for participants. This may arise given that at zero support level, other kinds of robot dynamics are in play as mentioned in Section 2.3.2. The extent to which this is relevant is open for future discussion.

The results from the confusion matrix show that the users were more accurate in estimating that the support level does not change. In addition, there is a distinct difference between perception in decreasing the support level and increasing the support level. This may arise from the fact that from full support to 0.65 support there is a big change in spring stiffness and thus it is more noticeable. It could also be that decreases in support level are more intuitive to the user. Once they have exerted a specific previous force, the fact that they have to exert more force for lower support levels is more distinct than if the support level is raised, given the action of the robotic arm.

The support levels used in Experiment II seem to be a good division of the support level range. Given that there are only five effective support levels in which the robot and the arm interact, fewer support levels may prove insufficient for rehabilitation purposes. The amount of levels to which the 0.25 to 0.65 range can be divided up is open for further research, although the results from the perception of change in support level raise the question on whether it may better to have a more divided range so that the support level change remains as unnoticed as possible for the participant during rehabilitation.

Chapter V — Discussion

The discussion chapter will focus on answering the research question by presenting the results of the report, situating them on a broader context and evaluating them. Given the structure of the report, separated in Chapters III and IV, the discussion chapter will be divided in a similar fashion for the sake of clarity. In such a way, Section 5.1 of this chapter will cover the first sub-question and the results from Chapter III, while Section 5.2 will answer the second sub-question by stating the relevant results from Chapter IV.

During each of the aforementioned sections, the corresponding research sub-question will be answered and supported by showing the relevance of the results. Furthermore, conflict and limitations of the results of each chapter will be similarly discussed, along with suggestions on potential future work on each section.

Section 5.3 in this chapter will address the main discussion, where the discussions on the two previous sections will come together so as to answer the main research question. This will show the validity of the research presented in this dissertation by providing a final solution to the problem as well as further limitations and potential future work.

5.1 Time prediction validation – Discussion

This section of the discussion chapter will cover the findings in Chapter III regarding reaching time prediction as well as giving an answer to the first research sub-question. In order to do so, there will be an analysis of the results of the model selection and of Experiment I, in that order. Next, the results will be framed in a bigger context to state their relevance. To finalize the section, some limitations will be discussed as well as possible lines of work.

Beforehand, let us go back to the sub-question related to Chapter III. Recalling the introduction chapter, the research sub-question was formulated as follows:

- *What is a valid Machine Learning approach regarding prediction of healthy user reaching time for upper extremity assistance and control purposes?*

The motivation behind the first research sub-question was to assess whether the predicted reaching time generated by a machine learning model would output results that are acceptable for use regarding robot behavior. The next paragraphs will therefore establish the validity of the used machine learning approach for such a purpose, by evaluating the results and defining their limits of accuracy.

5.1.1 Principal Findings and Context of Research

Firstly, let us address the machine learning model results described in Section 3.3. These results show two relevant points: the importance of selecting appropriate features and the limits of prediction performance.

The RMSE results from the comparison in Figure 12 are representative of these two points. After selecting relevant features for training, there was a significant reduction in RMSE for each algorithm when compared to the raw dataset. This illustrates the importance of having an appropriate set of model features.

Furthermore, the results show that the cross-validated performance of the models appears to be maxed out at around 0.12 seconds no matter the tuning of the hyperparameters. This apparent limit in performance for the training dataset represents the ability of the models to describe the mapping of the features to the predicted output and could be motivated in the limited dataset size or in not having enough features to describe the relationship between input and output. A limit of 0.12 seconds is however believed good for the application at hand. Therefore, no investigation in the addition of other features is recommended, whereas the issue of training dataset size will be resolved by the addition of new training data in future experimental sessions.

It is worth noting that the set of features is relevant regardless of the training dataset, whereas the limit in performance of 0.12 seconds is specific to the research on the investigated models. The notion of accuracy performance limit as well as the dataset size effect can however be applicable to future machine learning models.

Once the results from the machine learning model have been analyzed, let us turn to the results from Experiment I. The following can be extracted by looking at Figures 13, 14 and 15.

First of all, the comparison of the three evaluation metrics –percent error, RMSE and MAE- all show similar results in terms of the trend in differences between model performances across the different comparison techniques –CV (where applicable), BtT and TtT-. Given the scope of research and the description of the evaluation metrics in Section 3.2.1, the comparison of the three metrics is redundant, as they all indicate similar differences in performance. RMSE will be recommended as the only evaluation metric to be used in future studies, as it is more widely used and gives a better indication of high errors in predictions, which are undesirable for robot behavior.

Second, Random Forest will be the recommended algorithm to be used in future research. This is motivated by several reasons detailed next. One of them is that the algorithm shows a fast training time, which is useful for obtaining models in-between uses of the device.

Another reason is that, out of all the algorithms, RF proved to be one of the least accurate for predicting BtT reaching task time -albeit within a short margin-, while it obtained the best accuracy for TtT when trained with BtT data. This supports the idea that Random Forest is more flexible in determining the underlying mapping from feature data to prediction, a desirable property when predicting TtT tasks with BtT data.

When compared to AdaBoost –which showed acceptable results in both BtT and TtT predictions-, a Random Forest is easier to implement and interpret. In combination to the point made about single decision trees in the paragraphs above, there are enough supporting arguments to reason that the use of a Random Forests as the Machine Learning Model to be used in the eNHANCE device setting is a good option.

Third, the results show that the predicted reaching time RMSE is between 0.19 and 0.23 seconds for TtT tasks and between 0.09 and 0.11 seconds for BtT tasks, when using a training BtT dataset. This suggests that a training dataset from specific tasks and participants can be used for predicting reaching time for different tasks and participants within a prediction accuracy limit. As stated above, this limit may vary depending on training dataset size and feature selection, but less than 0.25 seconds is considered as an acceptable accuracy limit error for the application at hand.

Extra analysis on the generalization of the prediction and the effect of changing the training dataset can be seen in Appendix A.1. Even with a limited dataset size, the accuracy error stayed within less than or close to 0.25 seconds for the RMSE metric. Furthermore, and given that RMSE penalizes high errors in prediction, it can be seen that a BtT training dataset for TtT reaching time prediction performs slightly better –with an average RMSE of 0.18 seconds across participants- than a TtT training dataset for BtT reaching time prediction across participants –with an average RMSE of 0.23 seconds-. This difference might not be that relevant considered its magnitude, and it supports the following proposal: data acquisition in future research for later training of a Random Forest model can be of either a BtT or TtT task, but if one task is to be preferred, it should be BtT reaching movements.

In such a way, Experiment I provides a methodology to acquire data for training a machine learning model. Such a model is recommended to be a Random Forest in future research, and evaluated by using the Root Mean Squared Error. Ideally, the training dataset will be a combination of BtT and TtT tasks features, with more emphasis on BtT task data if need be. Furthermore, the limits of accuracy for the model should be enough for robot behavior if kept under 0.25 seconds. In the long run, it is expected that with a bigger training dataset size, the performance of the model will increase i.e. a better accuracy obtained. Machine learning is therefore suited to map the underlying function between movement features and predicted reaching time, within a specific limit of accuracy.

The first sub-question is therefore answered by having the previous paragraphs in mind: The machine learning approach presented in the research methodology of this dissertation is valid within a limit of accuracy of less than 0.25 for assistance and control during upper limb rehabilitation. The validity of this limit of accuracy is further supported by the variability in the reaching time observed in participants during the experiments in Chapter IV and Appendix B.1.

Once the first research sub-question has been answered, it is wise to situate the relevance of the results into a broader context. In this report, the machine learning approach was used with the aim of providing a way for the robot to be able to behave like a healthy user, or at least, as similarly as possible as a healthy user.

The relevance of the result therefore revolves around robot behavior: why it matters and why should it resemble human behavior. As stated in Chapter II during the background section, there are several advantages of using robots instead of conventional therapies in rehabilitation. One of such advantages is the fact that rehabilitation is promoted by repetitive task-oriented exercises and a robot is able to deliver such exercises in a much more efficient way than a therapist. A literature review on robotics applied for stroke rehabilitation discusses how motor learning is possible in stroke patients as well as how a rehabilitation robot therapy can be more efficient compared to traditional therapy, given that the robots can deliver more training exercises in the same time period [25].

Since the robot is in charge of assisting in the movement, robot behavior is important. A movement which is too fast may endanger patients, especially those with reduced mobility, and a movement which is too slow will not promote motor learning.

Healthy user reaching time is therefore a good candidate for being a target reference time, especially for full assistance. This is further motivated by the fact that during ADL, healthy user reaching times are desirable performances for assistive devices, as they result in activities being carried out by the user and the robot at a level of that of a healthy individual.

5.1.2 Limitations and Future Work

Having answered the first research sub-question and putting the results in a bigger context, an evaluation of the results to pinpoint limitations and future lines of work will be presented.

The most pressing limitation of the results is the fact that the data acquisition in Experiment I and the training dataset derived from the previous studies were obtained without the use of the robotic arm. Healthy participants were instructed to carry out the experiments at their maximum performance, resulting in an average reaching time between 0.6 and 0.8 seconds. The reaching movements were done without any physical constraints around the arms of the participants, in contrast to the case when the robotic arm is used.

The average reaching time is therefore not representative of the action of the user in combination with the robot. In addition, reaching times of the healthy users may prove too demanding for the robot to be achieved, given that the robot has a limited capacity in terms of generating fast velocity profiles with high accelerations. The reason behind reaching time data acquisition without the robot was a limited robot availability during the experimental period and the extent to which the training dataset derived from Experiment I can be used in the control strategy is open to investigation.

In future implementations of the machine learning approach, training data will have to be acquired from healthy users reaching for a target when the robot does not provide support – i.e. when the support level is 0-. This will ensure that the motion of the participant will take into account the robot interface and that the limits of robot action are not exceeded in the resulting reaching time prediction. With that in mind, the reaching time prediction can be used as part of the control strategy presented in Section 5.3.

The above derives in another limitation of the results: the available dataset. The data acquired during previous research and in Experiment I will not be of use in future implementations of the control strategy with the robotic arm. Therefore, a new training dataset will have to be acquired from several future uses of the robotic arm. The data acquisition could be done by the internal sensors of the robotic arm and the eye-tracking module, however, the use of the touchscreen as in previous research could provide a more independent acquisition of the data –for further validity-, as well as being faster and easier to carry out.

During this section, the results of Chapter III have been used to answer the first research sub-question in the context of robot behavior, as well as indicating limitations in the work and future recommendation. From the results, the main takeaway is that the machine learning model unit from the system architecture will consist of a random forest model, trained with features regarding healthy user reaching tasks performed in the setting of the robotic arm. This will result in a reaching time prediction that will be used both in assistance and as part of the control strategy proposed in Section 5.3.

5.2 Support Level Controller – Discussion

This section will cover the discussion on the results obtained from Chapter IV. The purpose of this section will be to answer the second research sub-question by evaluating said results. To do so, there will first be an analysis of the results, stating the most relevant findings. This analysis will consist of a brief overview of the findings during a preliminary study of the support level controller model, followed by a more extensive breakdown of the results obtained in Experiment II. Next, the results will be put into context by contrasting the findings with relevant literature. After that, the section will end by mentioning the limitations and possible future work on the support level controller.

It should be noted that, given that a preliminary discussion was already written for the experiment (Section 4.4), the analysis of the results in the present section will be presented in a much more straight-forward fashion.

To start with the content of this section, let us recall the second research sub-question:

- *What is the behavior of a proposed support level controller in terms of delivering different support levels based on user input?*

The research question refers back to the problem of adjusting support level depending on user input. The support level controller unit was added to the system architecture in Figure 1 as a way to address this problem. The support level controller model was later postulated in Section 4.2.1 so as to offer a description of the support level controller unit.

The behavior of such support level controller model will determine whether the support level can be modulated depending on the contribution of the user, to which extent it can do so, as well as pinpointing potential model improvements and considerations. The second research sub-question will thus be answered by stating the results derived from Chapter IV.

5.2.1 Principal Findings and Context of Research

The next paragraphs start the analysis of the results by addressing the relevant findings obtained during the design of the support level controller model. These results are further described in Appendix B.3 and provide insight into the effect of the enough work count threshold $W_{cont,th}$ and the margin of acceptable work $W_{margin.}$ In this preliminary study, the effect of each of these parameters was investigated, so as to shed a light on their effect in the behavior of the support level controller. Table 1 summarizes the results from such study:

$W_{cont,th}$	W_{margin}	**Mean Support Level**
2	0	0.4538
1	0	0.3841
1	0.02	0.3605
2	0.02	0.3917

Table 1. Summary of the results from Appendix B.3. The values of $W_{cont,th}$ *and* W_{margin} *were modified to observe their effect on the mean support level observed after 40 controller iterations.*

The mean support level is the average support level obtained for 40 iterations of the controller with a participant dataset. Said dataset contained information on the participant performing at a 0.38 support level for different reaching trials.

The results show that a bigger margin results in a lower mean support level i.e. there is more focus on user effort and motivation. On the other hand, the higher the enough work count threshold, the higher the mean support level and thus the system will be more focused on achieving the reaching task.

In a rehabilitation scenario of the eNHANCE device, the idea will be to motivate the user and thus a good initial setting would be an enough work count threshold of 1 and some degree of margin based on user contribution –to be determined by the researcher-. However, the parameters may be adjusted if more assistance is needed for achieving the reaching task. It should be noted that this study was carried out previous to the implementation of the controller in the device, and therefore more experimental research can be carried out to fully determine the effect of this two parameters on the support level controller behavior.

After this first specific analysis on controller related parameters, let us now turn the focus on the main outcomes of Chapter IV: the results from Experiment II. The results will be discussed in the same order as the research was conducted: training phase, threshold acquisition phase, scenario phase and participant perception.

The most relevant point to be made about the results in the training phase is the time it took for the participants to get used to cooperating with the device. As evidenced by the results in Section 4.3 and Appendix B.4, almost all participants except Participant 5 reached optimal accuracy – around 1 centimeter- within the first 15 movements regardless of randomized support level. These results suggest that a minimal of 15 movements done at random support levels could be used as a lower limit for future training with the robotic arm.

Next, let us focus on the results from the threshold phase. As stated in Section 4.3, there is a general trend of higher interaction work at lower support levels for all participants. The use of the regression line derived in results will have to be validated in future experiments. It is hypothesized that the regression line should pose no problem for healthy users but future work should examine the behavior of the controller when a stroke patient is using the robotic arm with such a threshold interaction work relationship. The current design of the controller is based on the idea that stroke patients output less work than healthy users. This may not hold true given that stroke patients present compensatory strategies which could affect their force profile.

There is however existing literature supporting the fact that work in stroke patients can be used as a measure for assessing rehabilitation. In such a study, the work of stroke patients was initially low and significantly improved after training [26]. The question on stroke user work comparison to healthy users is still unanswered, but the study does provide an insight into work differences in stroke patients before and after training.

The results of the scenario phase will be stated next. These results will give useful insight into the behavior of the postulated support level controller model.

For a normal scenario, the support level converged to zero, as expected for healthy participants. In line with the research in literature indicated in the previous paragraph, a stroke patient may contribute less to the reaching movement in terms of applied forces. In such a case, the results of the stroke patient instructed to perform normally are hypothesized to resemble the behavior observed in the fatigue scenario for healthy users.

Another research point during the scenario phase was to establish whether the acquired data could help in determining if the user is underperforming – being lazy- or if the user is fatigued. The results from the force profiles showed that a possible determination point for distinguishing both cases could be found in the time at which the user starts contributing to the movement. In a lazy scenario, the participant will take longer to act, as they first wait to see if the robot is able to reach the target with no additional input.

Furthermore, motivational feedback presents itself as a good method to encourage lazy participants to perform better. The results from the scenario phase regarding motivation during the lazy scenario are however inconclusive and further work should be done in this regard. It is hypothesized that under proper motivational feedback, the difference between the fatigue and lazy scenarios in terms of set support levels will be more pronounced.

Finally, the results regarding user perception will be discussed. The mean accuracy across participants in estimating the change in support level is of around 50%. Within this mean accuracy, same –i.e. no change- support levels were more often correctly reported, followed by a decrease in support level and finally an increase in support level, the latter being the less accurate perception.

These results raise the question of whether the division of support levels set during Experiment II is appropriate for its use in a rehabilitation scenario. There were 7 different support levels in the experiment which resulted in the previously stated perception accuracy. If a lower perception accuracy is desired, more divisions within the support levels can be set, so that the change is smoother. However, the extent to which the perception should be correct is open for discussion.

On the one hand, it may be beneficial for the progression of the reaching exercises if the user does not notice the changes in support level, so that there is no user bias in the next movement. In special, an increase in support level noticed by the user could be detrimental, since the user could identify that they have to engage less during the movement. In such a way, by reducing the perception in change of support level, the user can better focus on the rehabilitation task, instead of spending cognitive resources on the support level perception.

On the other hand, it is perhaps beneficial that there is some degree of accurate perception in support level, especially in a decrease of support level. Literature shows that error-enhancing techniques result in better training and rehabilitation scores than corrective rehabilitation methods [27]. Even though the eNHANCE device is not an error-enhancing device per se, but acts on an assistance-as-needed principle, the study showed that there is indeed adaptation in stroke patients when the error is increased. For the application at hand, a hypothesis can be raised on whether if a patient notices a decrease in support level, they will foresee a higher error and respond by raising their engagement. If such is the case, the correct perception of a decreased support level could be beneficial for the rehabilitation process.

In such a way and taking both of the above points, it is hypothesized that a 50% accuracy could be a good middle ground on which to define the division of support levels, so that the trade-off between focusing on the reaching task and increased engagement is balanced.

It is worth mentioning that the perception in changes in support level is limited by the fact that the support level changes in a one-step fashion -either the next higher, next lower or same support level-. For small changes in support level –and thus in spring stiffness- it is hard for the user to estimate the change. As suggested above, this fact might be beneficial for rehabilitation purposes, as it avoids bias from the participant on the next expected support level.

The main outcome from the results regarding the behavior of the controller has been stated in the previous paragraphs. With all the above in mind, the second research sub-question can be answered: the behavior of the postulated support level controller in terms of adjusting support level suggest that it is indeed possible for the controller to adjust support level in terms of user and robot performance as evidenced by the changes in support levels during Experiment II.

The next step in this discussion section will be to situate the results in a general context. The field of relevance of the support level controller originates in the role of training in rehabilitation for stroke patients. More specifically and as evidenced by the literature review in Chapter II: motivational exercises, which play an important role in therapy.

In the setting of rehabilitation, assistance-as-needed robots can provide a good platform for applications that require a balance between assistance in a task and motivation during exercise. Through the implementation of controllers in the rehabilitation robots, the provided assistance can be tuned depending on several user and robot related metrics. The idea for a support level controller unit in a control strategy to provide appropriate assistance in the setting of the eNHANCE device stems from this context.

Literature regarding the design of controllers for upper limb rehabilitation is novel, with a few studies showing promising controller implementations. A literature review discussing different types of controllers for rehabilitation in neurologic injury provides a clearer view of the aforementioned [28]. The control strategies vary from predefined control assistance, in which the thresholds or limits of the controller are set throughout the task to be performed, to adaptive controllers, which change their parameters while the user is performing the task.

The support level controller model postulated in this report falls into the category of predefined assistance-as-needed controllers, as it works on the basis of predetermined thresholds.

5.2.2 Limitations and future work

The limitations of the research done in Chapter IV are of importance to define future improvements. The main limitation of the research is the fact that all results have been derived from healthy user data. More specifically, the interaction work has to be contrasted with stroke patient data, so as to fully state that it can be used as a user performance metric.

Another key limitation regarding interaction work is the fact that the experiment was carried out for a single target. The resulting regression line describing the interaction work versus support level relationship may not hold for targets in other positions of the workspace, which could negatively affect the behavior of the support level controller in determining the performance of the user. During a preliminary study in Appendix B.2, a small insight regarding the use of the controller for different targets was investigated. This study showed potentially positive results, but further research should be done.

In addition, the motivational feedback to engage lazy users has to be further implemented, with either the addition of the motivational platform depicted in the system architecture, or a better user interface for future experiments with the robotic arm. Given the poor implementation of feedback during experiment II, no conclusive results could be found for the behavior of the controller and for determining whether it could differentiate between lazy users and fatigued users.

Future lines of work to address the limitations regarding the controller of the behavior are proposed in the next paragraphs.

Ideally, the support level controller will have to be tested in an experiment including stroke patients. This will provide useful insight into the force profile of stroke users and consequently on their contribution in terms of interaction work. Through the experiment, the minimum interaction work of stroke patients versus support level relationship can be established.

In case that experiments with stroke patients are too time-demanding for the development of the project, another option is to use healthy users with added impairments. These impairments will be aimed at mimicking the behavior of stroke patients in healthy users. A possible approach could be to use restraining braces or weights, however, this could be counterproductive given the set-up of the robotic arm.

A more feasible technique would be to intentionally make the user fatigued by instructing them to lift some weights or do some exercise before using the robotic arm. Through several sessions, data on normal, fatigued and lazy performances from different participants could be acquired for a better insight into the behavior of the controller.

Furthermore, the set-up during these sessions could include a much more dynamic set of reaching movements, so that there are more targets in space. In such a way, useful information on the difference in performance for different positions in the workspace –if any or significant- can be found.

A current line of research within the project is the addition of a motivational platform, if ready for integration, it could be used in the procedure proposed above. Otherwise, a simple feedback from the robot to the user can be set up, either auditory or visual. This will validate the decision flowchart of the model as described in Section 4.2.1, by relating feedback, controller behavior and user performance.

This section has offered an insight into the relevance and limitations of the results obtained during Chapter IV. In order to properly conclude the section, let us summarize the key takeaways.

On the one hand, the support level controller model contains two parameters that can be changed by the researcher to put more emphasis on assistance – higher $W_{cont,th}$ and therefore higher mean support level – or motivation - bigger W_{margin} which will decrease the support level-.

On the other hand, in an experimental setting a minimum recommended number of training movements is 15. In addition, the interaction work versus support level regression line obtained during Experiment II can be used for future experiments. Regarding controller behavior, the support level was indeed adjusted in terms of user performance. However, further research should be done on the differences between a lazy participant and a fatigued participant by looking into a better motivational feedback and investigating the difference in force profiles.

Lastly, the division of support levels in promoting rehabilitation should be focused on the 0.25 to 0.65 range. The number of divisions is open for research and discussion, based on user perception of changes in support level.

5.3 Main Discussion – Control Strategy

This section will combine the discussions of Section 5.1 and 5.2 in order to answer the main research question. Given the analysis, limitations and future work derived from the results in the previous sections, this section will synthesize them to first provide an overview of the proposed control strategy. This control strategy will be then postulated as a platform on which future lines of work can be built upon.

Recalling the introduction chapter, the main research question in this dissertation was stated as follows:

- *What is a possible control strategy such that the device is able to adjust its support during reaching tasks in rehabilitation?*

The main research question was motivated on the problem statements regarding the behavior of the robot and the adjustment of support level. Robot behavior was covered in Chapter III and discussed in Section 5.1 and it investigated how the robot should operate during the reaching tasks in rehabilitation. Adjustment of support level on the other hand was addressed by the postulation and investigation of a support level controller, covered in Chapter IV and Section 5.2.

5.3.1 Control Strategy

To answer the research question, a possible control strategy will resemble the high-level system architecture from Figure 1 and will be specified by the results obtained from Chapter III and Chapter IV. In such a way, the proposed control strategy which answers the main research question is as depicted in Figure 36:

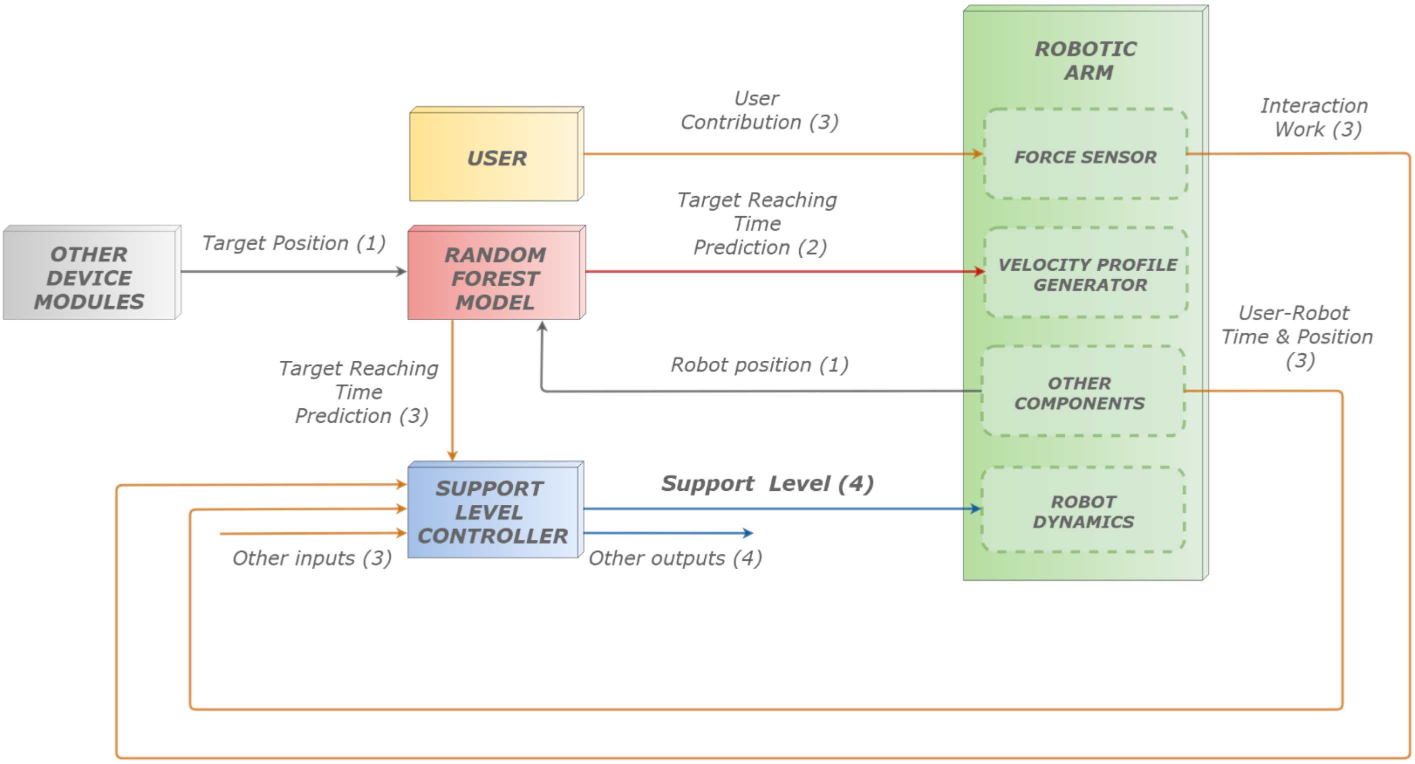


Figure 36 - Control Strategy. The units and subunits depicted in the figure as well as the relevant parameters are related to the system architecture in Figure 1 through the content of the report as well as Figures 3 and 14. The numbers by the parameters indicate the step in which they can be observed.

Let us expand on the control strategy defined in Figure 36. The units represented in the control strategy have been defined and investigated all throughout the report: the other device modules unit, the machine learning model unit –which is now referred to as random forest model-, the user unit, the robotic arm with its corresponding subunits and the support level controller unit. All of them will play a role in the control strategy and their interactions are defined by the system parameters as specified in Figure 36.

It can be noticed that beside the definition of the system parameters in the interaction between units, there is a number. This number will represent the time step in which each parameter can be observed. The steps are defined in a sequence of time events during the reaching exercises. In such a way, the controller action in terms of time steps can be defined as:

1. After the other modules of the eNHANCE device detect the intention of the user, the target position and the robot position will be used as input features in the random forest model.

2. The random forest model will output a reaching time prediction and will convey this information to the velocity profile generator within the robotic arm. In such a way, the behavior of the robot is established for the desired movement.

3. In accordance with the robot dynamics of the robotic arm and the behavior determined by the velocity profile generator, the movement will be performed by the user and the robot. After the movement is performed, the user performance metric –interaction work- and the user-robot performance metrics – robot position and time- will be used as inputs for the support level controller. In addition, the predicted reaching time of the movement – to be used as a time reference- as well as the other inputs – as defined in Section 4.2.1- will similarly be used as inputs for the support level controller.

4. After the support level controller has reached a decision following the flowchart in 4.2.3, some outputs of the system will be produced to be used in the next movement iteration. Most importantly, a support level will have been set and used to modulate the spring stiffness of the robot dynamics for the next movement.

This control strategy is designed with the different components of the eNHANCE device in mind (as described in Section 2.3), as well as with the results from Chapter III –the random forest model- and from Chapter IV –the support level controller model-. In such a way, the control strategy will adjust the level of support in a reaching task rehabilitation scenario.

5.3.2 Limitations and Future Work

Once the main research question has been answered, let us discuss the limitations of this control strategy. The limitations connect to the discussions in Section 5.1 and Section 5.2.

Regarding the machine learning model, the main limitation is that as of the writing of this dissertation, the available training dataset is small in size. This means that the control strategy may not perform properly –in terms of task fulfillment- if implemented without first gathering more training data. Furthermore, the limits of accuracy of the training data from the results in Chapter III –less than 0.25 seconds- are only hypothesized to be similar to those that will be obtained in the control strategy, as the training datasets originate from different tasks –exercises without and with the robotic arm-.

In terms of the support level controller there are several limitations found in the control strategy. First, the question of how the support level controller will behave for stroke patients, given that they may show a different interaction work profile is open for further research. Some ideas on how to address this fact and about experiments to be carried out were given in Section 5.2. In addition, the support level controller model could be further improved, so that the control strategy is more efficient in promoting rehabilitation and motivation.

In order to do so, the integration of a communication between the support level controller model and the motivational platform from the system architecture in Figure 1 will have to be implemented. However and at this current state of research there is no such communication between the support level controller and the motivational platform.

Furthermore, the design of the support level controller model does not have to be limited to the rules defined in Section 4.2.1, other controller implementations could be more efficient – such as an adaptive controller, as discussed in Section 5.2-.

The last limitation to remark is that the control strategy has yet to be implemented in the framework of the eNHANCE project. The random forest model and the communication with the other modules of the device will have to be integrated into the ROS environment first, so as to use the control strategy.

Possible lines of work for the control strategy will be discussed next. Regarding the random forest model, a line of work would be to implement the control strategy and to test the limits of accuracy of the predictions compared to the times found in the reaching tasks performed by the robot and the user. If the limits are similar to those found in the results from Chapter III, no additional modifications or additions may be done on the training dataset. Furthermore, by performing such an experiment, more training data can be simultaneously acquired. This data will contain information on reaching time for different targets in space when the robotic arm is used by a healthy user with no support and could be used to improve the limits of accuracy. The reaching tasks from the experiment could cover a mix of BtT tasks and TtT tasks so as have more data on both types of reaching exercises.

For the support level controller, possible experiments were already proposed in Section 5.2. By carrying out such experiments, more information on controller behavior can be obtained and thus the controller can be further tuned. An additional line of work would be to expand the controller with parameters coming from the motivational platform, so that the decision to change support level is further based in user motivation and exercise engagement. Furthermore the addition of a GUI for the support level controller in which the controller settings can be changed may prove useful for future researchers.

Let us now discuss possible work regarding the overall control strategy. The research throughout this report has been revolving around short-term solutions, limitations and improvements regarding the device. However, and as stated in the background section, the rehabilitation therapies in stroke patients are usually applied over long periods of time.

In such a way, a long-term control strategy could be beneficial for keeping track of the improvement of stroke patients, as well as for delivering appropriate support levels depending on the stage of rehabilitation.

For such a purpose, let us first picture the rehabilitation process of one individual. The user will take part in several therapy sessions using the eNHANCE device throughout what could be months. In the beginning, the stroke patient will display impaired mobility and therefore the control strategy will tend to set support levels that are in the high range. Ideally, by the end of the therapy, the stroke patient will have improved their performance in terms of reaching the targets, and therefore the control strategy will tend to set lower support levels.

In order to keep track of this change, it will be of use to compute a mean support level for a specific patient session. In such a way, the progression of support level and session number could give an indication of whether the stroke patient has been rehabilitated. In addition, the next session can be programmed to start with a specific support level, given the participant's progression in mean support level.

Let us now think about generalizing this to several stroke patients. If a correlation between the mean support level progression and Fugl-Meyer Assessment scores is found, a database relating both parameters could be established. If a new stroke patient is going to use the eNHANCE device, they could be assessed in terms of their FMA score and through the database, assigned the appropriate therapy in terms of support level.

Having answered the main research question, the research in this report is finalized. In such a way and within the context of reaching task in upper limb rehabilitation for stroke patients, the main outcome from the dissertation has been the proposal of a control strategy, which uses a machine learning approach to dictate robot behavior while being able to adjust support level in terms of user contribution.

Chapter VI — Conclusion

The conclusion chapter will summarize the contents and findings of presented in this master thesis. First, a brief summary of the research carried out will be written down, while revisiting the relevant results and answers to the research question. Then, a brief recapitulation of the limitations of the work presented in this thesis will be done which will lead to revisiting future lines of work. Finally, the contribution of the research done in this report will be presented, along with a final statement.

The main research question in this research revolved around designing a control strategy for its use in the eNHANCE device, in order to provide assistance-as-needed to the user in reaching tasks for upper limb rehabilitation. To approach the research question, two main fields of research were investigated: the validation of a machine learning approach for its use in determining robot behavior during reaching tasks, and the design and evaluation of a support level controller to adjust the support delivered by the robotic arm.

The validation of the machine learning approach was based on evaluating four different machine learning algorithms – a single tree, a randomized forest, an extremely randomized forest and an AdaBoost algorithm-, and was addressed during Chapter III. The evaluation was carried out in two steps, first a methodology was drafted to obtain a proper model for each of the algorithms, and second the models were used to predict two different types of reaching tasks: Base-to-Target and Target-to-Base. In doing so, the validity of a previous training dataset would be investigated as well.

The results of the model selection determined the input features to be used in the prediction of reaching time: robot position, target position, the distance between both and their relative angle. The methodology to obtain the machine learning models was an additional result from the model selection research.

The results on reaching time prediction for different reaching tasks determined the limits of accuracy obtained for the models trained with the previous dataset of Base-to-Target reaching time data. The results for Base-to-Target prediction showed a Root Mean Square error between 0.09 seconds and 0.11 seconds. The results for the Target-to-Target predictions had a Root Mean Square Error between 0.19 seconds and 0.23 seconds. From the results, it was decided that a random forest model would be used as part of the control strategy, as it showed to be the model that showed the better trade-off between the accuracy of predictions.

With the above in mind, a limit of accuracy of 0.25 was deemed to be enough for the applications at hand, in such a way the validity of using machine learning approach to determine robot behavior within this accuracy limit was corroborated.

The other scope of research addressed the adjustment of the support level and was covered in Chapter IV. First, a support level controller model was postulated, which defined the relevant metrics that the controller will use to reach a decision regarding a change in support level. These metrics were: distance error, time error and interaction work. The latter defines user engagement in a given reaching movement, by giving an indication of the forces exerted by the user on the robotic arm.

The behavior of the controller in an experimental setting was observed, by conducting an experiment with three scenarios: a normal scenario, a fatigue scenario and a lazy scenario. The behavior showed that the controller is indeed able to adjust the support level in terms of the performance of the user in cooperation with the robotic arm. Furthermore, the perception of the participants in changes of support level was investigated. The mean perception accuracy was 50.8%. With no change in support level being the most accurate perception, followed by a decrease in support level and an increase in support level, in order of correct accuracy.

With the above in mind, let us restate the main research question one final time:

- *What is a possible control strategy such that the device is able to adjust its support during reaching tasks in rehabilitation?*

The results from both chapters were taken into account to answer the main research question by proposing the control strategy in Chapter V. The random forest model and the support level controller therefore play key roles in said control strategy, whose purpose will be to provide assistance-as-needed during rehabilitation for stroke patients by adjusting the support level.

The limitations of the control strategy are consequently related to the random forest model and the support level controller. There is not enough training data at the time of writing this report to evaluate the predicted time from the random forest model within the control strategy, in such a way, the limits of accuracy for the control strategy are yet unknown.

Furthermore, the support level controller shows inconclusive results on differentiating fatigued and lazy performances. It follows that the support level controller model might need to be updated in the context of the control strategy to account for this differentiation, otherwise the adjusted support level might not promote user motivation.

In order to address these limitations, future lines of work have been proposed. First, the appropriate limits of accuracy for the control strategy will have to be determined. In the processes, training data for the random forest model will be acquired. This experiment would consist of a combination of both Base-to-Target and Target-to-Target reaching tasks, which will have to be performed with the robotic at zero support. If needed, a model can be then obtained after fitting the training dataset, following the methodology proposed in Chapter III. In such a way, the limits of accuracy of the control strategy for robot behavior during assistance can be redefined.

After carrying out the work proposed in the previous paragraph the behavior of the support level controller can be further investigated. A possible experiment could be to design different sessions for a specific participant. In one of these sessions, the participant would lift weights previous to the use of the robotic arm so as to simulate fatigue. Furthermore, the motivational platform could be integrated if available, otherwise a more precise motivational feedback from the controller to the participant can be designed with auditory or visual stimuli.

As a final note on future work, once the control strategy is integrated, a functionality that computes the mean support level for a single participant session should be added. This is motivated by the discussion in Section 5.3.2, regarding how this mean support level could be used as a reference for observing the progression in rehabilitation for a specific user and to determine which support level should be given to new users.

To conclude and as evidenced by the background and the discussion chapters, the use of robotics in rehabilitation is a novel and promising field within therapy for stroke patients. The eNHANCE device is therefore a promising platform to detect user intention and provide assistance-as-needed to heavily affected stroke patients.

The research presented in this report is therefore relevant for the field of robotics in rehabilitation, as it is framed within the scope of the eNHANCE device. In particular, the results of the work presented in this thesis –in the form of a control strategy- provide a solid ground to deliver proper assistance during reaching tasks repetitions while motivating the user by adjusting the support level of the robotic arm.

Appendices

APPENDIX A — Machine Learning Approach & Robot Behavior

Appendix A.1 – Generalization and TtT-trained Model Comparison

As an extra point of further analysis, a comparison between a TtT trained model and a BtT one was investigate as well as a generalization of the results for the participant data. Given the results from Experiment I, the analysis will be done with a random forest as a base model.

The procedure was as follows. Taking the experiment data, a certain participant data was left out as a test dataset while all the other participant's data was used. The number of data points was reduced in the BtT training dataset to match the number of samples of TtT via a randomized deletion of data.

Two models for both TtT and BtT training data were selected on the basis of a grid search and RMSE minimization. All features were taken into account. Afterwards, both models were used for prediction of BtT and TtT reach time with the input features of the withheld participant data.

The trained models were:

Random Forest BtT trained

n_estimators = 500

max_depth = 37

min_samples_leaf = 2

min_samples_split = 1

Random Forest TtT trained

n_estimators = 500

max_depth = 13

min_samples_leaf = 15

min_samples_split = 2

The goal of this section will be to determine whether a TtT trained model could be more suited for BtT and TtT reaching time prediction that a BtT trained mode. In addition, given the fact that in order to get a training dataset and a comparison dataset one participant's data was left out, some generalization observations could be made from the results.

Figures 37, Figure 38 and Figure 39 show the percent error, the RMSE and MAE for each held participant data. In such a way and for each left-out participant data, the BtT trained model and the TtT trained one where used to predict the left-out participant reaching time data with both the BtT and TtT features of the participant.

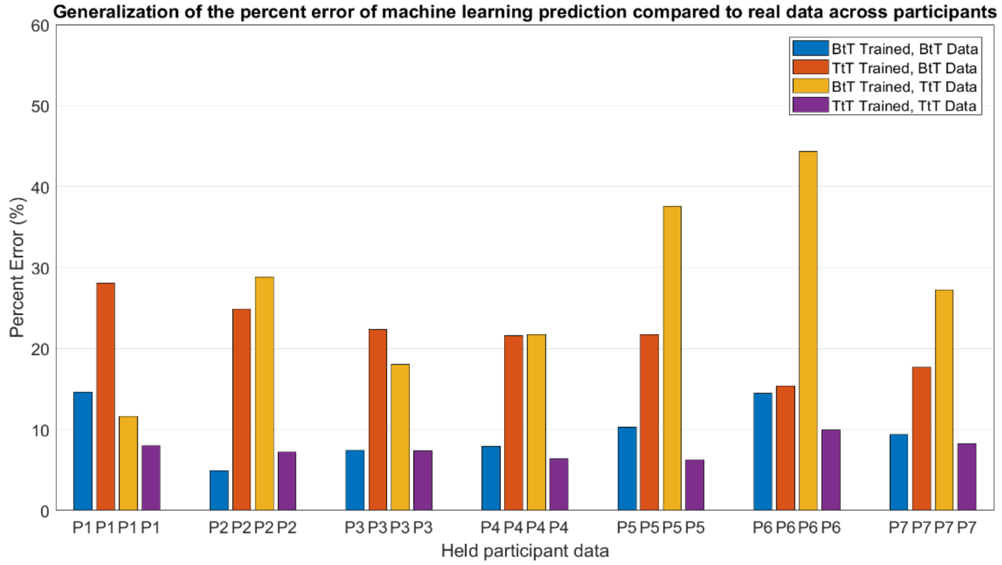


Figure 37 Appendix – Percent Error Generalization. Percent Error of prediction for models trained with either BtT or TtT Data and compared to either BtT or TtT reaching time data from a held-out participant dataset.

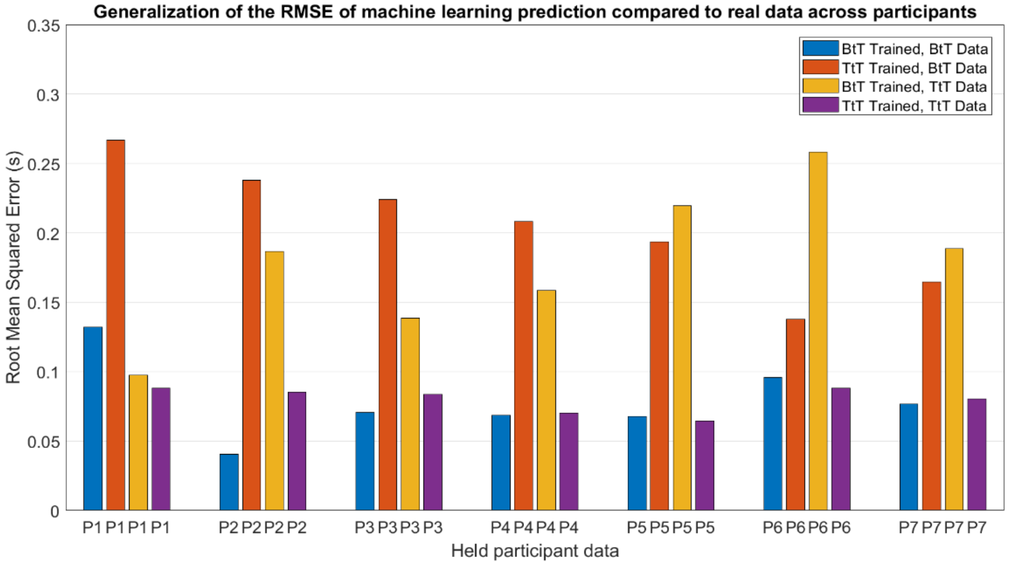


Figure 38 Appendix – RMSE generalization. RMSE of prediction for models trained with either BtT or TtT Data and compared to either BtT or TtT reaching time data from a held-out participant dataset.

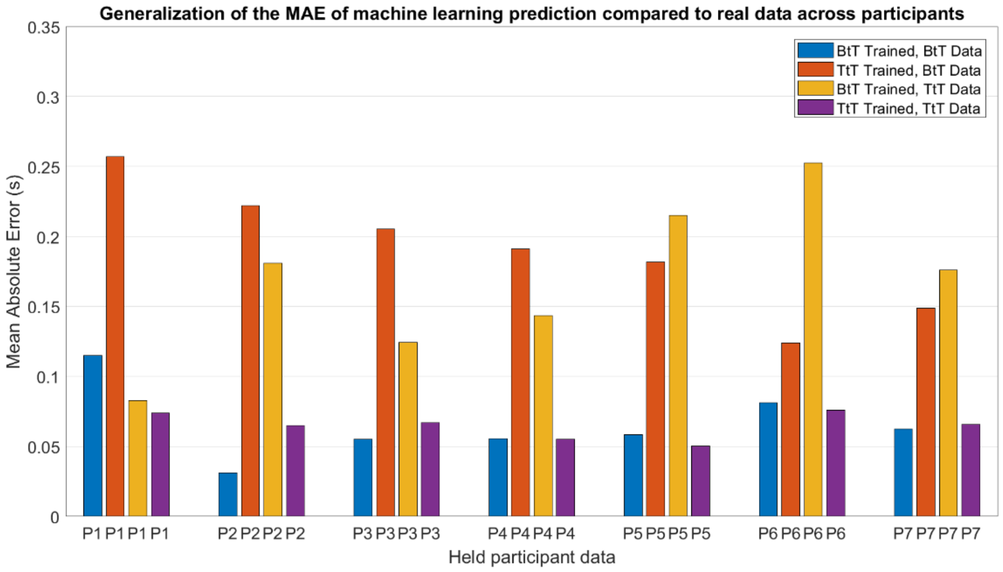


Figure 39 Appendix –MAE generalization. MAE of prediction for models trained with either BtT or TtT Data and compared to either BtT or TtT reaching time data from a held-out participant dataset.

The following can be said for the results. Even with a limited dataset size, the accuracy error stayed within less than or close to 0.25 seconds for the RMSE metric. Furthermore, and given that RMSE penalizes high errors in prediction, it can be seen that a BtT training dataset for TtT reaching time prediction performs slightly better –with an average RMSE of 0.18 seconds across participants- than a TtT training dataset for BtT reaching time prediction across participants –with an average RMSE of 0.23 seconds-. This difference might not be that relevant considered its magnitude, and it supports the following proposal: data acquisition in future research for later training of a Random Forest model can be of either a BtT or TtT task, but if one task is to be preferred, it should be BtT reaching movements.

APPENDIX B — Support Level Controller

Appendix B.1 – Time and Distance Analysis

The analysis of the data in a preliminary experiment will look into the possibility of using the proposed performance metrics as control values for the adjustment of the support level. Intuitively, the error in distance and error in time are likely to be good performance metrics, which are a combination of the user arm and the robotic arm performance.

Figure 40 shows the error in distance relative to the target position and error in time relative to the time predicted by the robot.

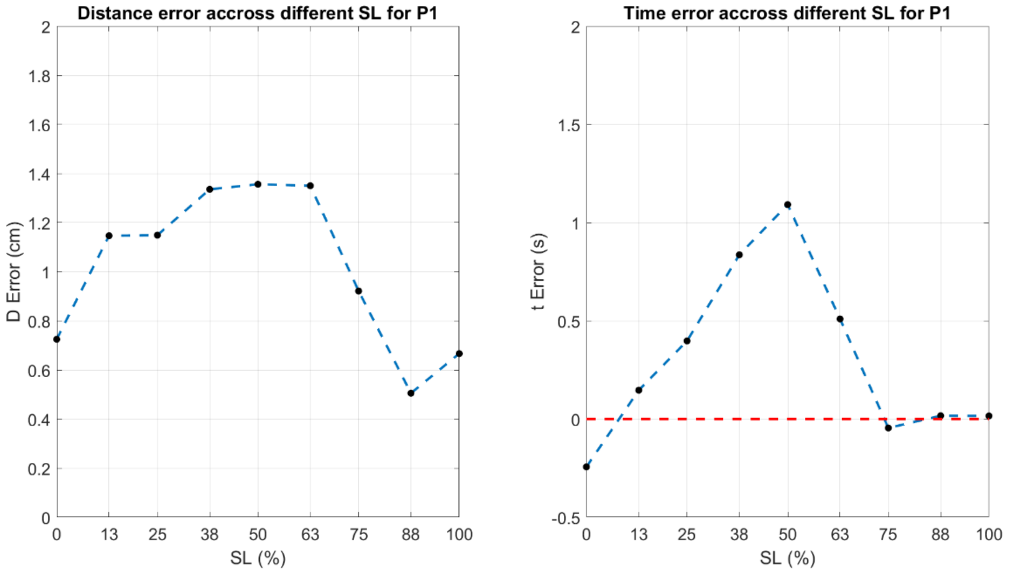


Figure 40 Appendix - Time and Distance error study. Average distance error and time error across different support levels for participant 1 reaching for all targets.

At high support levels, namely the 75-100 % range, the robot is driving the motion of the user's arm and robotic arm. In other words, the distance errors associated to that range are the accuracy of the robot in reaching the target. It can be observed that for the 38-63 % range the accuracy decreases. This is the range in which both the robot and the user work together towards reaching the target. The accuracy increase is likely the result of this interaction, as the user has to find a way to adapt the behavior of the robot. This idea is further supported at the 0-25 % support level range, where the accuracy increases again. In special at support level 0 where the user is mostly in control of the motion, the accuracy is high.

In the time error plot one can see a similar phenomenon taking place, where at higher support levels the error is almost zero and for the middle range it increases only to decrease in the lower support level section. Most noticeable is the negative value of time error for the zero support level, which means that the user's action led the robotic arm to travel faster than what it expected.

Appendix B.2 – Use of Controller for Different Targets

Another investigation point is concerned with the use of the support level controller for different target positions. In order to analyze this point, Figure 41 shows the behavior of the support level controller with data from participant 2 but from different targets. The starting support level was set at 0.63, since the 1 to 0.75 range was perceived as full support by the users.

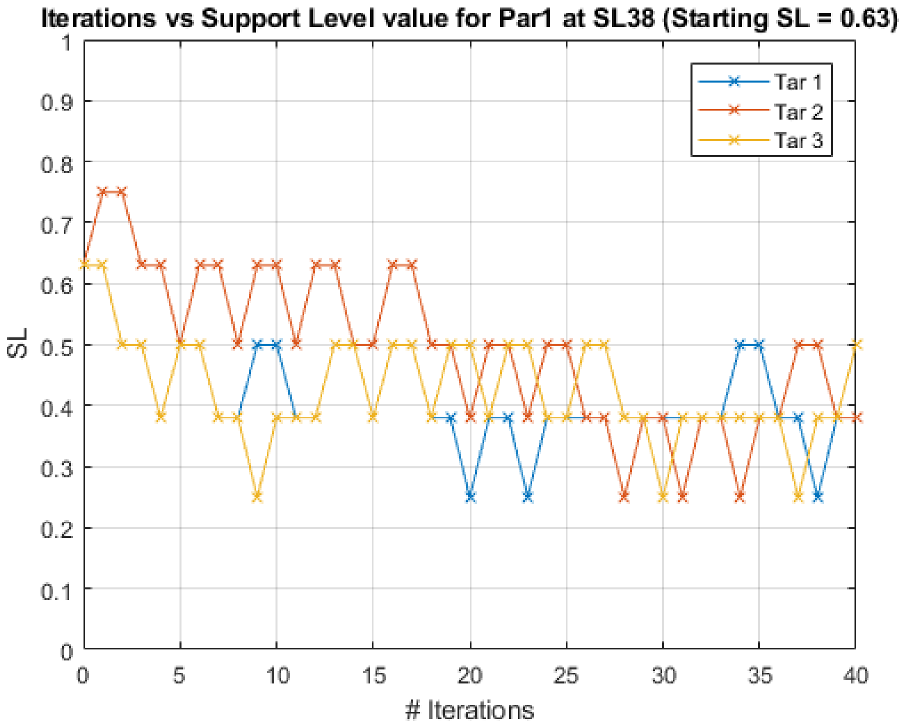


Figure 41 Appendix - SL controller Behavior: variability between targets. Plot showing the behavior of the support level controller starting from 1 Support Level and fed with participant 2 data towards target 2 at a 0.38 support level performance.

The figure shows that for each target, the set support level converges to the performance level of the user. This suggests that a single interaction work - support level relationship could be used across targets. The difference in support level progression may come from the randomized selection of movements towards the target, however, further research on the use of a single interaction work - support level relationship for different targets in the workspace should be done.

Appendix B.3 – Enough Work Count Threshold & Work Margin

Let us investigate the two additional support level controller parameters: $W_{cont,th}$ and W_{margin} and their effect on the behavior of the support level controller.

As said in Section 4.2.1, $W_{cont,th}$ is an upper threshold for an inner support level controller counter that will determine how many times the user has shown enough interaction work and reached acceptable limits. W_{margin} on the other hand delimits the acceptable range of interaction work under the specified threshold. The next four Figures 42, 43, 44 and 45, will show the effect of varying these two parameters both by themselves and jointly.

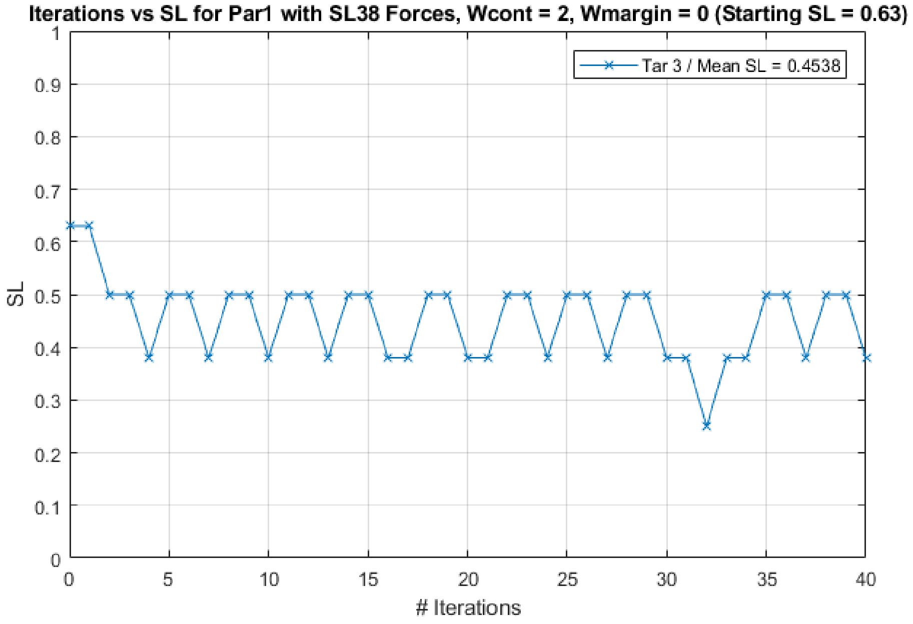


Figure 42 Appendix - Study Wcont & Wmargin 1. Plot showing the behavior of the support level controller starting from 0.63 Support Level and fed with participant 1 data towards target 3 at a 0.38 support level performance. The Wcont,th is set to two while Wmargin is set to 0. The mean support level after iterations is shown on the upper right corner of the plot.

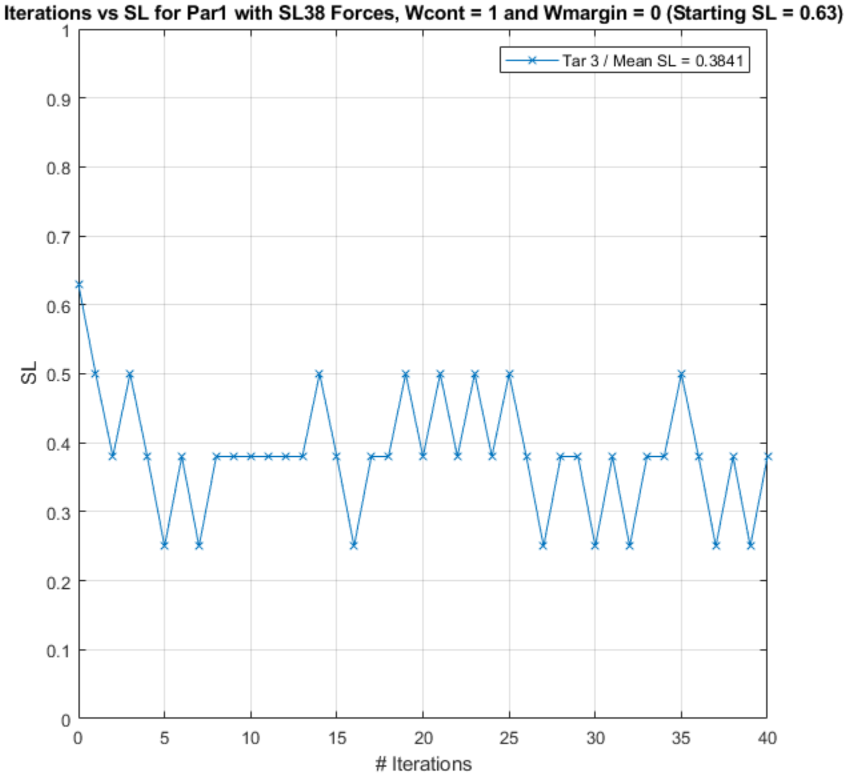


Figure 43 Appendix - Study Wcont & Wmargin 2. Plot showing the behavior of the support level controller starting from 0.63 Support Level and fed with participant 1 data towards target 3 at a 0.38 support level performance. The Wcont,th is set to 1 while Wmargin is set to 0. The mean support level after iterations is shown on the upper right corner of the plot

From the two figures above, one can see that $W_{cont,th}$ determines how long the it takes the support level controller to change between support levels, given that the user is in an acceptable range of interaction work. The mean support level of Figure 42 with $W_{cont,th}$ = 2 is higher than that of Figure 43 and so a higher emphasis on assistance is given. It should be noted that amount the support level fluctuations when $W_{cont,th}$ = 1 is really high, which might not be desirable from a use perspective.

Figure 43 can be then compared with Figure 44 down below to observe the effect of W_{margin}. At a bigger margin, the fluctuation of the support level is greatly reduced, and the mean support level is reduced. This will result in the support level controller acting in a motivational fashion. The amount of margin that the support level controller should allow is open to research.

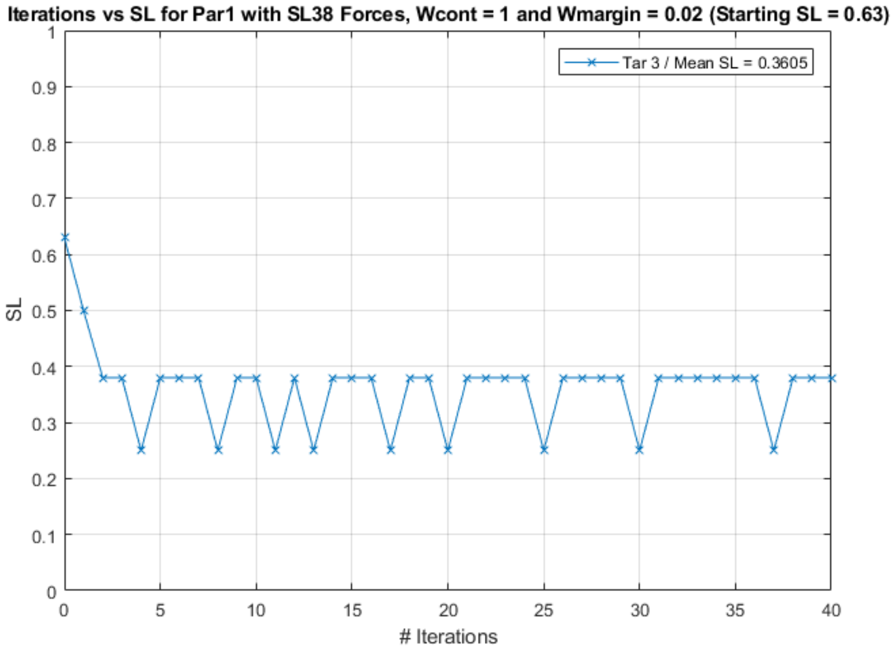


Figure 44 Appendix - Study Wcont & Wmargin 3. Plot showing the behavior of the support level controller starting from 0.63 Support Level and fed with participant 1 data towards target 3 at a 0.38 support level performance. The Wcont,th is set to 1 while Wmargin is set to 0.02. The mean support level after iterations is shown on the upper right corner of the plot.

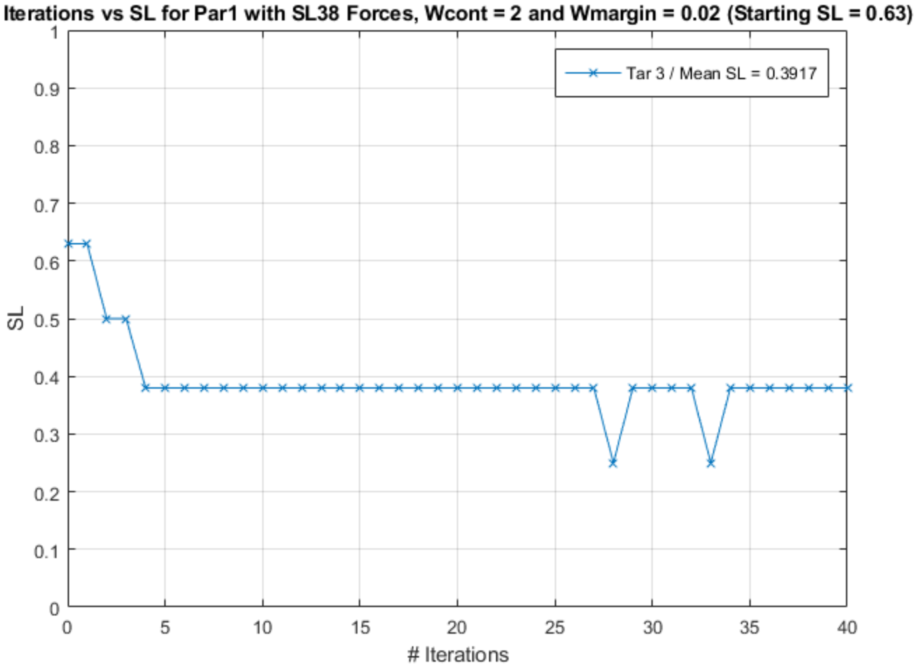


Figure 45 Appendix - Study Wcont & Wmargin 4. Plot showing the behavior of the support level controller starting from 0.63 Support Level and fed with participant 1 data towards target 3 at a 0.38 support level performance. The Wcont,th is set to 2 while Wmargin is set to 0.02. The mean support level after iterations is shown on the upper right corner of the plot.

In the last plot shown in Figure 45, the joint action of both $W_{cont,th}$ and W_{margin} can be observed. The fluctuation in support level is minimal and the mean support level converges to the expected support level within a small difference. The desirability of this support level controller behavior should be investigated in future work.

As a general conclusion from this investigation, it can be said that the user defined support level controller parameters $W_{cont,th}$ and W_{margin} are able to shift the support level controller's behavior to either a motivational or an assistance focus.

In such a way a higher margin results on a lower mean support level, that is, there will be more focus on user effort and motivation. On the other hand, the higher counter threshold, the higher the mean support level and so the system will be more focused achieving the reaching task.

Appendix B.4 – Other Figures – Experiment II

This section will present the entirety of the results from the training phase and scenario phase for all participant that too part in Experiment II.

<u>Training phase</u>

The distance error and time error for all participants during the 20 training movements are shown in the figures of the training phase. By each data point, the support level at which the movement was performed is indicated with a number in red. Participants 2 and 4 were especially good at getting used to the device, with a low error at early movements, while participants 1 and 3 took a bit longer, stabilizing around 15 movements. Participant 5 showed great time errors and thus they would need more training movements.

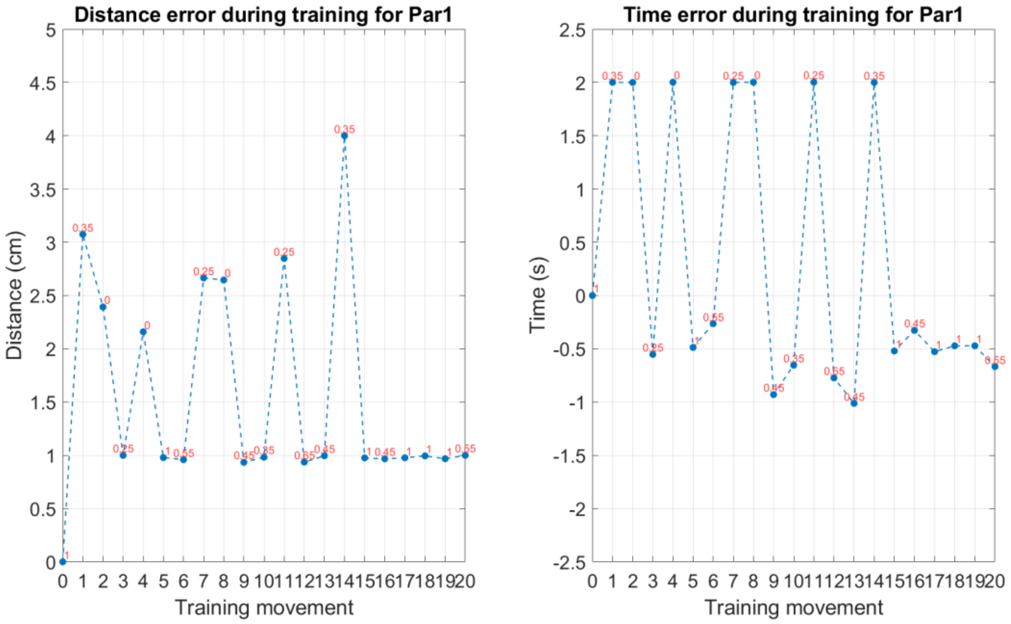


Figure 46 Appendix - Distance & Time error for Participant 1.

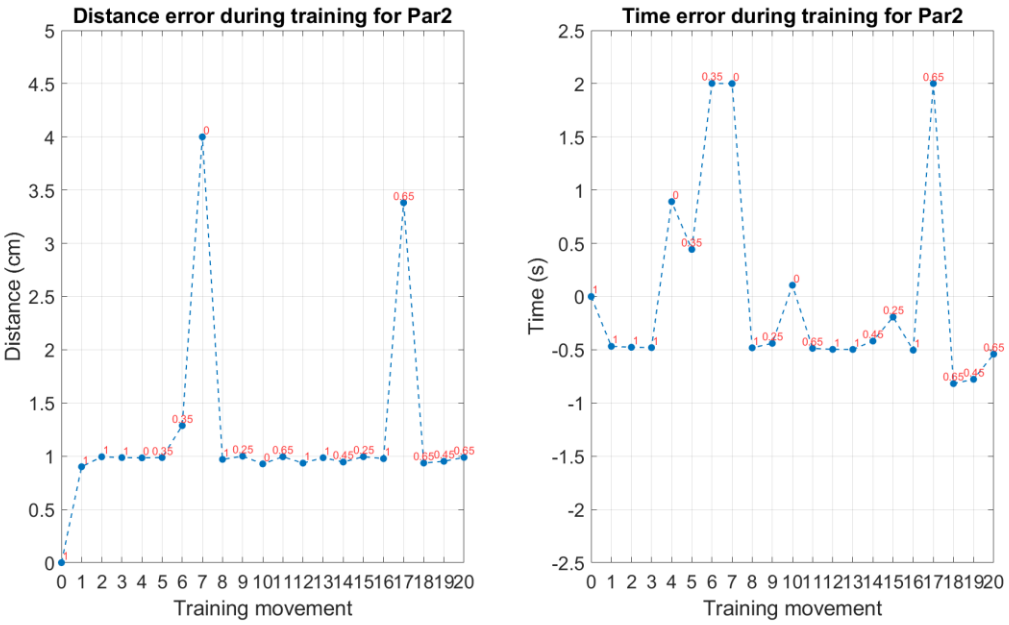


Figure 47 Appendix - Distance & Time error for participant 2.

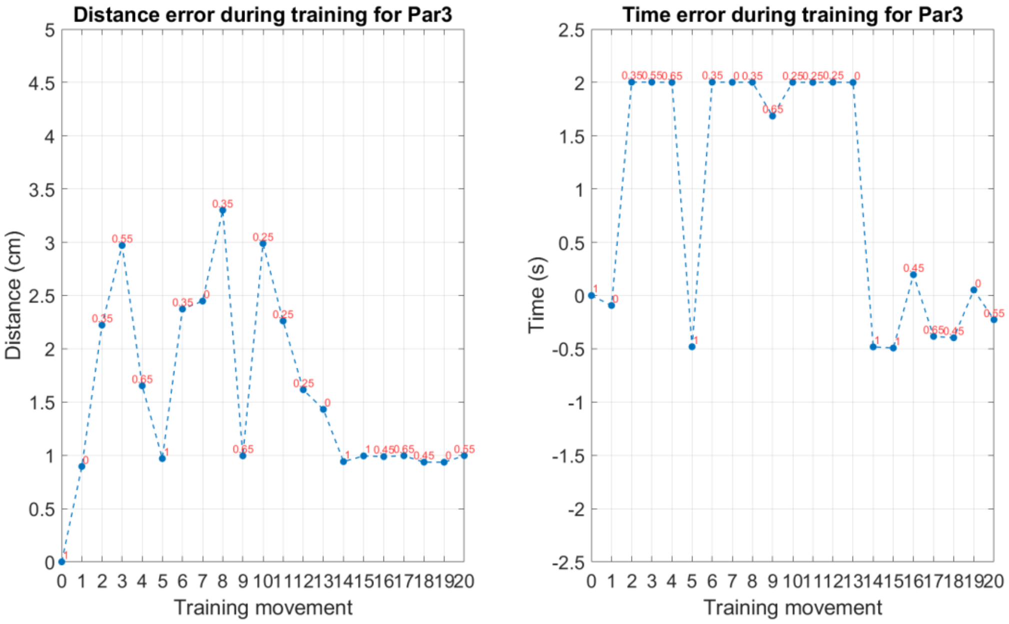


Figure 48 Appendix - Distance & Time error for participant 3.

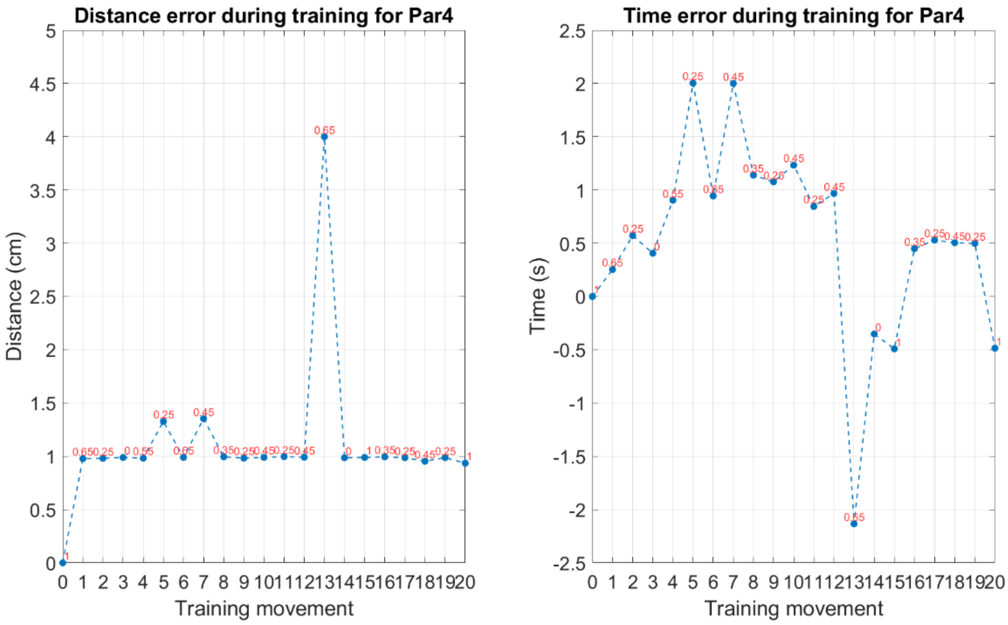


Figure 49 Appendix - Distance & Time error for participant 4.

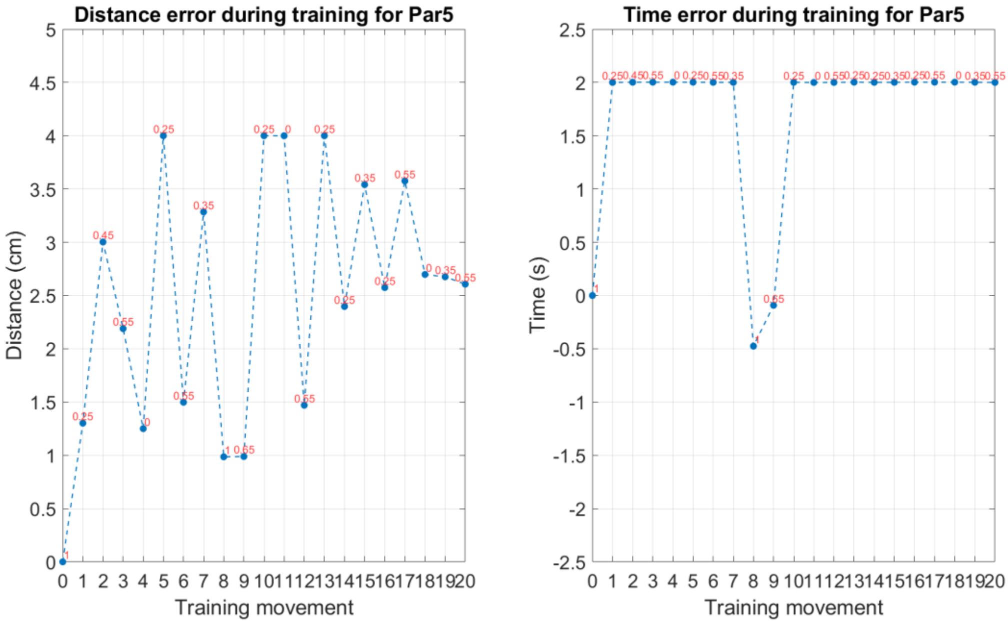


Figure 50 Appendix - Distance & Time error for participant 5.

Scenario Phase

The following figures display the behavior of the support level controller in terms of set support level progression during each scenario for each participant. For the normal scenario all support levels converged to zero around the 20th movement, which is an expected controller behavior. For the fatigue scenario all participants showed a convergence of support level, except for participant 5. Finally, the lazy scenario showed the most variability between participants, with participants 1 and 2 staying at a high range of support levels, whereas the support level was decreased in other participants. This last effect is thought to come from a poor lazy scenario feedback implementation.

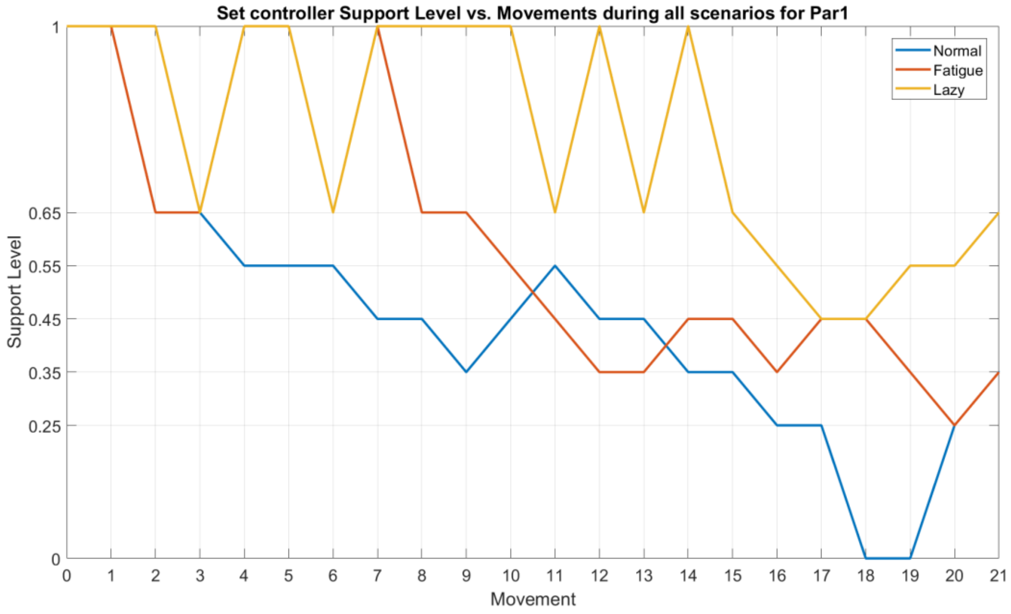


Figure 51 Appendix - Scenario phase results for participant 1.

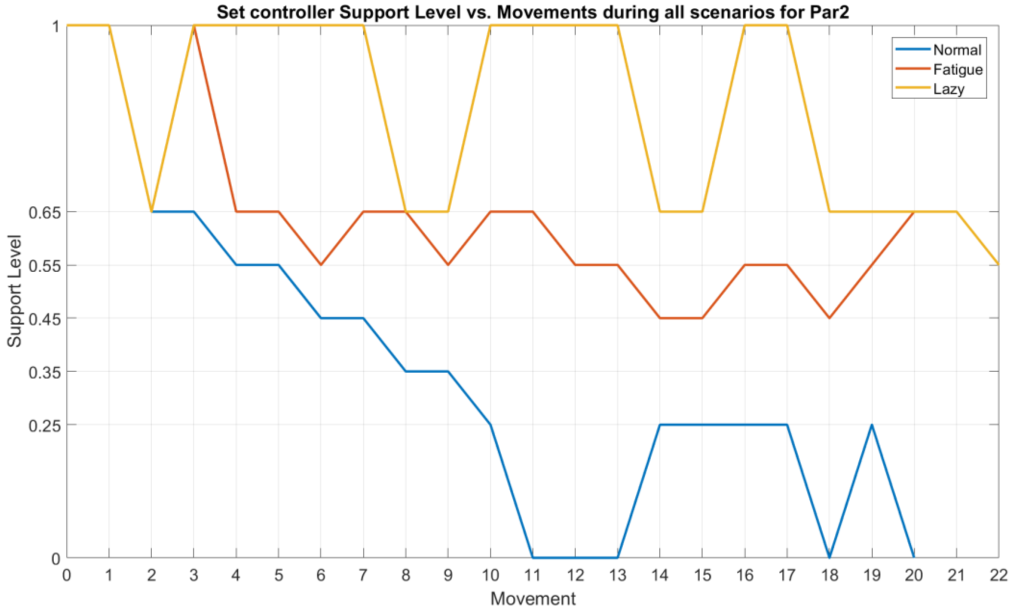


Figure 52 Appendix - Scenario phase results for participant 2.

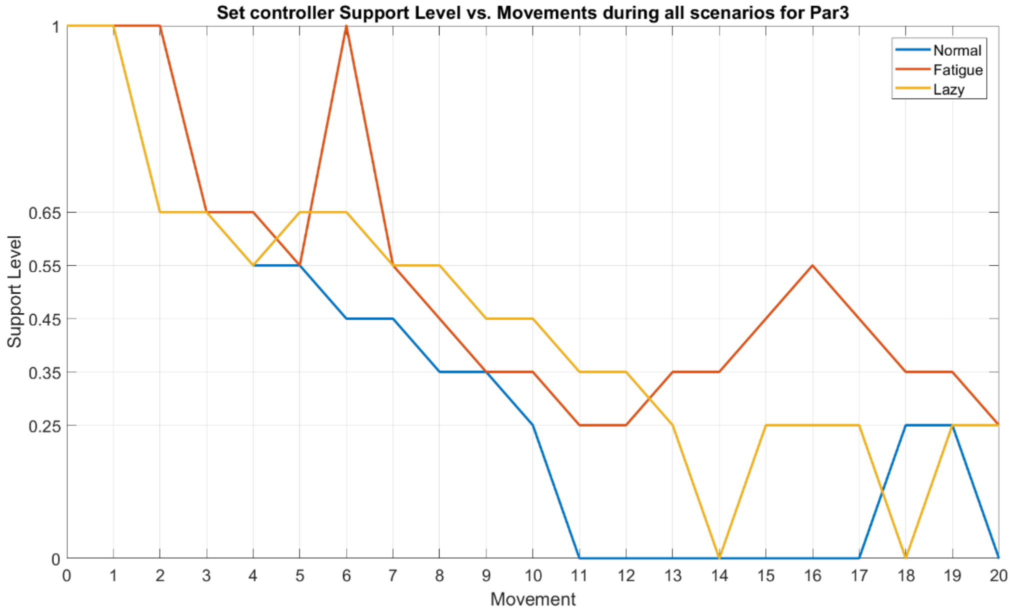


Figure 53 Appendix - Scenario phase results for participant 3.

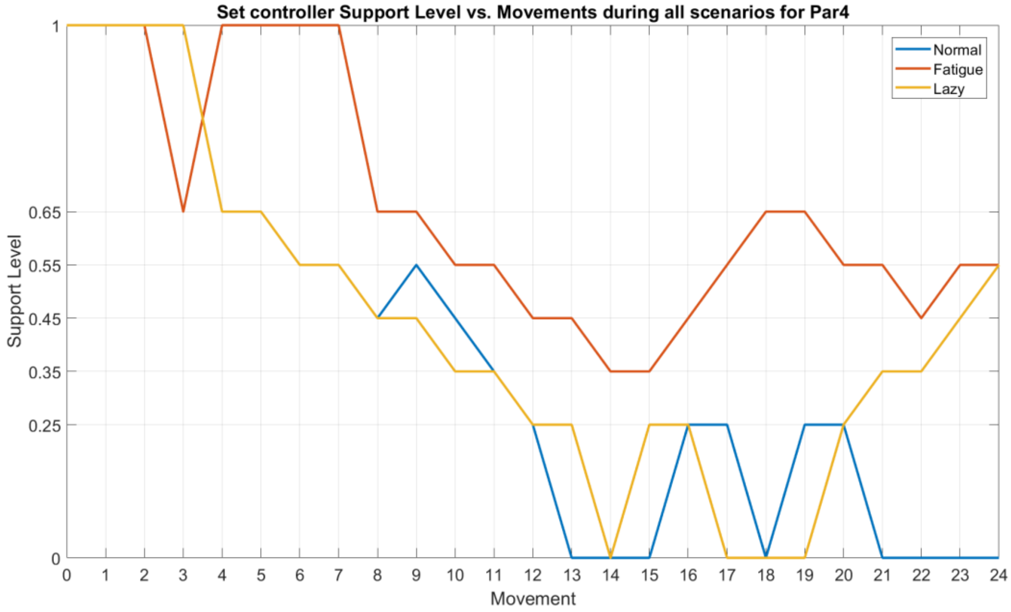


Figure 54 Appendix - Scenario phase results for participant 4.

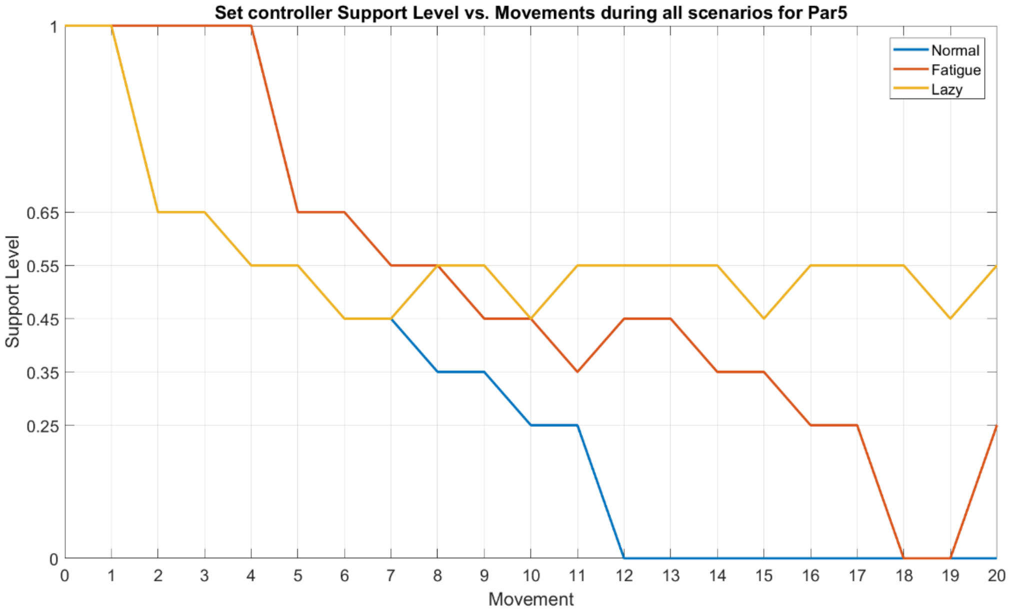


Figure 55 Appendix - Scenario phase results for participant 5.

References

[1] "Stroke | National Heart, Lung, and Blood Institute (NHLBI)", Nhlbi.nih.gov, 2018. [Online]. Available: https://www.nhlbi.nih.gov/health-topics/stroke. [Accessed: 16- Nov- 2018].

[2] "WHO | The world health report 2002 - Reducing Risks, Promoting Healthy Life", Who.int, 2018. [Online]. Available: http://www.who.int/whr/2002/en/. [Accessed: 16- Nov- 2018].

[3] C. Han, M. Arbib and N. Schweighofer, "Stroke Rehabilitation Reaches a Threshold", PLoS Computational Biology, vol. 4, no. 8, p. e1000133, 2008.

[4] "Poststroke Rehabilitation: Assessment, Referral, and Patient Management", Topics in Stroke Rehabilitation, vol. 3, no. 2, pp. 1-26, 1996.

[5] P. Langhorne, J. Bernhardt and G. Kwakkel, "Stroke rehabilitation", The Lancet, vol. 377, no. 9778, pp. 1693-1702, 2011.

[6] S. Wolf, C. Winstein, J. Miller, E. Taub, G. Uswatte, D. Morris, C. Giuliani, K. Light, D. Nichols-Larsen and f. EXCITE Investigators, "Effect of Constraint-Induced Movement Therapy on Upper Extremity Function 3 to 9 Months After Stroke", JAMA, vol. 296, no. 17, p. 2095, 2006.

[7] N. Maclean, P. Pound, C. Wolfe and A. Rudd, "The Concept of Patient Motivation", Stroke, vol. 33, no. 2, pp. 444-448, 2002.

[8] W. Chang and Y. Kim, "Robot-assisted Therapy in Stroke Rehabilitation", Journal of Stroke, vol. 15, no. 3, p. 174, 2013.

[9] P. Langhorne, F. Coupar and A. Pollock, "Motor recovery after stroke: a systematic review", The Lancet Neurology, vol. 8, no. 8, pp. 741-754, 2009.

[10] B. Volpe, H. Krebs, N. Hogan, L. Edelstein, C. Diels and M. Aisen, "A novel approach to stroke rehabilitation: Robot-aided sensorimotor stimulation", Neurology, vol. 54, no. 10, pp. 1938-1944, 2000.

[11] R. Gassert and V. Dietz, "Rehabilitation robots for the treatment of sensorimotor deficits: a neurophysiological perspective", Journal of NeuroEngineering and Rehabilitation, vol. 15, no. 1, 2018.

[12] R. Bertani, C. Melegari, M. De Cola, A. Bramanti, P. Bramanti and R. Calabrò, "Effects of robot-assisted upper limb rehabilitation in stroke patients: a systematic review with meta-analysis", Neurological Sciences, vol. 38, no. 9, pp. 1561-1569, 2017.

[13] R. Loureiro, F. Amirabdollahian, M. Topping, B. Driessen and W. Harwin, Autonomous Robots, vol. 15, no. 1, pp. 35-51, 2003.

[14] M. Guidali, A. Duschau-Wicke, S. Broggi, V. Klamroth-Marganska, T. Nef and R. Riener, "A robotic system to train activities of daily living in a virtual environment", Medical & Biological Engineering & Computing, vol. 49, no. 10, pp. 1213-1223, 2011.

[15] L. Dipietro, M. Ferraro, J. Palazzolo, H. Krebs, B. Volpe and N. Hogan, "Customized Interactive Robotic Treatment for Stroke: EMG-Triggered Therapy", IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 13, no. 3, pp. 325-334, 2005.

[16] L. Marchal-Crespo and D. Reinkensmeyer, "Review of control strategies for robotic movement training after neurologic injury", Journal of NeuroEngineering and Rehabilitation, vol. 6, no. 1, p. 20, 2009.

[17] "eNHANCE-motion – eNHANCE-motion", Enhance-motion.eu, 2018. [Online]. Available: http://www.enhance-motion.eu/. [Accessed: 16- Nov- 2018].

[18] T. Chai and R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)?", Geoscientific Model Development Discussions, vol. 7, no. 1, pp. 1525-1534, 2014.

[19] L. Hall, X. Liu, K. Bowyer and R. Banfield, "Why are neural networks sometimes much more accurate than decision trees: an analysis on a bio-informatics problem", SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance (Cat. No.03CH37483).

[20] L. Breiman, "Random Forest", Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.

[21] P. Geurts, D. Ernst and L. Wehenkel, "Extremely randomized trees", Machine Learning, vol. 63, no. 1, pp. 3-42, 2006.

[22] I. Maglogiannis, K. Karpouzis and M. Wallace, Emerging Artificial Intelligence Applications in Computer Engineering. Amsterdam: IOS Press, 2007, pp. 15-17.

[23] J. Friedman, R. Tibshirani and T. Hastie, "Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors)", The Annals of Statistics, vol. 28, no. 2, pp. 337-407, 2000.

[24] "Documentation scikit-learn: machine learning in Python — scikit-learn 0.20.0 documentation", Scikit-learn.org, 2018. [Online]. Available: https://scikit-learn.org/stable/documentation.html. [Accessed: 18- Nov- 2018]

[25] V. Huang and J. Krakauer, "Robotic neurorehabilitation: a computational motor learning perspective", Journal of NeuroEngineering and Rehabilitation, vol. 6, no. 1, p. 5, 2009.

[26] P. Lum, C. Burgar and P. Shor, "Evidence for Improved Muscle Activation Patterns After Retraining of Reaching Movements with the MIME Robotic System in Subjects with Post-Stroke Hemiparesis", IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 12, no. 2, pp. 186-194, 2004.

[27] J. Patton, M. Stoykov, M. Kovic and F. Mussa-Ivaldi, "Evaluation of robotic training forces that either enhance or reduce error in chronic hemiparetic stroke survivors", Experimental Brain Research, vol. 168, no. 3, pp. 368-383, 2005.

[28] L. Marchal-Crespo and D. Reinkensmeyer, "Review of control strategies for robotic movement training after neurologic injury", Journal of NeuroEngineering and Rehabilitation, vol. 6, no. 1, p. 20, 2009.