Dyads Learning in Movement Tracking

By
ADRIANO GENDY
B.S., Politecnico di Milano, Milano, Italy 2018

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Master of Science in Bioengineering
in the Graduate College of the
University of Illinois at Chicago, 2021

Chicago, Illinois

Defense Committee:

James Patton, Chair and Advisor

Cristian Luciano

Simona Ferrante, Politecnico di Milano

ACKNOWLEDGMENTS

I would like to thank my family that always supported me during all my life and allowed me to have this amazing experience here in Chicago. This thesis is yours. I hope to have made you proud and I hope to be able to return all the sacrifices you did for me. I hope I turned out to be the son you always wanted.

Thanks to my brother Alberto for always being there for me; thanks to my cousins Silvia and Julia for being like my sisters; thanks to Benny and Seba for being my friends for I don’t know how many years and for supporting and for tolerating me every time I did not know what to do; thanks to Antonio for being an older brother; thanks to Federica for being there from Venice to Milan to Chicago; thanks to Davide for being an amazing roommate and friend for the last year in Chicago; thanks to Alice, Ila and Mich to have made Milan as a home even if that city never belonged to me. Thanks to Testa who is always there to remind me how my soccer team is better than his, thanks to all of my Milan’s friends: Luca, Marco, Gio, Carlo and thanks to all my new Chicago friends, you made this experience away from home an unforgettable memory.

The list could go on and on, but the number of pages for the thesis is limited so I will just say a big thank you to every person that helped me, even involuntarily or with just a smile, during this journey

Last but not least, I want to thank also Professor Patton and all my lab mates that gave me the opportunity to work in a special environment and to achieve this important goal in my life. I want to thank my friend Mattia, we overcame a lot of challenges to build our device and I am happy to have been working with you.

Again, Thank you.

Adriano Gendy

TABLE OF CONTENTS

CHAPTER **PAGE**

1 INTRODUCTION......1

1.1 Motivation of this Thesis 1

1.2 Possible Applications of this Research Study 2

1.3 Stroke Rehabilitation......3

1.4 Robots Rehabilitation 4

1.5 Human-Human Interaction Studies 7

2 IMPLEMENTATION 16

2.1 Design and Building of the Pantograph 16

2.2 Data Acquisition......21

2.3 GUI 22

2.4 Data Processing and Analysis 24

2.5 Protocol 27

3 RESULTS 29

4 DISCUSSION 33

APPENDICES 36

Appendix A 37

Appendix B 47

Appendix C 49

Appendix D 53

CITED LITERATURE......54

VITA......56

TABLE OF FIGURES

FIGURE **PAGE**

1 The Lokomat device 8

2 Dual System for Reed and Patton's study 10

3 Completion times for dyads and for individuals alone 11

4 Dual Robotic System for Ganesh's study 12

5 Trend of error with respect to trials 13

6 Representation of the motor improvement in different types of experiments 14

7 Solidworks Mockup of the Pantograph 17

8 Disasembled Pantograph 18

9 Top View of the Pantograph 20

10 Side View of the Pantograph 21

11 GUI 23

12 Position Signal Before and After Resampling 30

13 Learning Curve 31

14 Exponential Fitting Curve of the Data 32

15 Permission grant to use Springer Nature Material 49

CONTRIBUTION OF AUTHORS

The part related to the implementation of the device has been done thanks to a teamwork that includes my advisor, Professor Patton, and my coworker Mattia Demasi.

SUMMARY

Human-human interaction has been a subject of great interest in the last years. The focus has been posed on the responses that these types of interactions bring in the term of motor behavior. Moreover, the studies on interpersonal communication are aiming to find new paradigms to exploit the signals that are interchanged between humans for the purpose of enhancing rehabilitative outcomes.

In the first part of this thesis we will present the motivation behind this study, how our device is inserted in the state of the art and the future aims that this specific field wants to achieve. Considering the latter, an a very brief overview of rehabilitation is presented to better understand how the connection of humans to a robot and the connection between humans through a robot can change the way of doing rehabilitation in the clinical environment. Finally, some related works are presented. We aim to verify the work of Ganesh et al (2018) using our experimental system. In the second part, the implementation and the design of our new device called Pantograph is presented with a detailed explanation of the techniques used, the design specifications, the structure of the data acquisition system, the functioning of the data analysis software and the protocol followed for this pilot study. In the third part, the results obtained in this pilot study and how the robot behaves in an actual experiment will be presented. Finally, the results obtained will be discussed. Since the device is a result of this study, in this part also the limits and the possible improvements that can be applied to the device are suggested for future works. This study represents a first step used to demonstrate the functioning of the device, the development of a first paradigm for future studies on human-human interaction by using this device and another proof that interpersonal interactions can be deployed in the clinical environment.

CHAPTER 1

INTRODUCTION

Little is known on humans interacting and learning together, and yet human-human interaction and learning is common in everyday life.

The study of dyads, a couple of interacting humans, has been of large interest over the years. This is because the human-human interactions processes can lead to different types of responses that can also affect the motor behavior of the subjects. These responses can be the result of two types of signals:

- Cognitive signals, that is the recognition of the individual with which an interaction is occurring
- Sensory feedback, such as visual, haptic and auditory signals

This study mainly focuses on the second type of feedback, particularly on haptic signals that are exchanged between two subjects coupled with a passive robot called Pantograph. However, the subjects involved in this study will notice the presence of another agent performing the task with them. This will allow us, with respect to other studies made in this field, to understand the effect of cognitive signals on the motor learning process. Hence, the interaction of humans is studied here through a cyber-human system (CHS).

These systems have been found, in recent studies, to enhance the motor performance of subjects when coupled with respect to the same task but performed individually.

As it will be shown in this thesis, a lot of studies have been done on human-human interaction and more of them are focusing on CHS and the results are promising. Other studies have the aim of studying the human-human interaction principles to use them in the design of the control system of robots that are engaged, for example, in the rehabilitation process in which is important to obtain an interaction between the robot and the user that allows variability of movements and that is intuitive and natural. This shows how promising and important is the investigation of the principles that rule human physical interactions.

1.1 Motivation of this Thesis

This research study on CHS and dyads was first developed by Dr. James Patton at Shirley Ryan AbilityLab (that was called Rehabilitation Institute of Chicago till 2016), which is a research hospital that is specialized in rehabilitation of patients that were subjected to severe impairments that can occur after spinal cord injury, stroke, amputation and other conditions.

The main idea is to exploit the potential of the CHS to define a new paradigm of rehabilitation in which more than one patient is trained at a time by one therapist. Indeed, it has been found that by the year 2030 there will not be enough therapists to oversee the required physical therapies in over 40 states [1]. Hence, having two subjects undergoing the therapy at once can lead to an important help for shortage of therapists. Not only that, a lot of studies have found how dyads actually perform better than individuals [2], how the improvement in performance varies with respect of the components of the dyads (it has been proven that when two novices, who are two individuals that have no experience on the dyadic task that they are asked to perform, are coupled together the outcome in improvement is better than when a novice is coupled with an expert [2]) and how the dyadic performance does not increase when a human is coupled with a robot [3].

All these results are a proof of how powerful the human-human interaction could be in the clinical field of rehabilitation. However, all these concepts can be applied also in other fields. So, a CHS can be used for training purposes, such as in the surgery field in which two apprentices can learn while using a robot that is among them and that provides optimal learning. Or a novice surgeon can be coupled with an expert that would lead his or her movements to teach the proper steps that must be performed while exploiting the precision and speed of a robot.

Hence, the main hypothesis of this research study is that human-human interaction through a robot can enhance motor skill learning. We, at the Robotics Lab of Shirley Ryan AbilityLab, conceived a new human-robot-human pantograph system that combines sensors and robotics to empower dyadic skill learning. The goal of my thesis is to build and demonstrate the mechanical system in a preliminary experiment.

1.2 Possible Applications of this Research Study

Motor learning is an important aspect of different fields such as rehabilitation, sports and surgery. However, even if motor learning has been largely studied through different methods, in a lot of fields (in particular in rehabilitation) there is need for more effective motor learning technologies. For this reason, exploiting CHS and human-human interactions can provide those results that individual tasks have not been able to give.

For example, in rehabilitation, even if robots for individual tasks have been proven to be as effective (or even better) as human therapists, dyadic CHS could give even more promising results in motor learning. Moreover, this paradigm can diminish the therapy time, the expenses of rehabilitation and improve patients' quality of life.

In the same way, interactive tasks that are driven by a robot can be very useful also in learning skills in sports. In this field dyadic CHS can be seen as a tool that can be used by professionals and amateurs that want to improve their game in a faster and more efficient way with respect to traditional sport training.

Finally, this type of learning modality can be used to train new surgeons to execute complex and very precise movements. In this case, the interaction can be seen as a "master-slave" type of interaction. Where a surgeon shows a novice how to perform the motion and then asks the trainee to reproduce it. The surgeon can then feel the haptic feedback given by the novice and correct him or her through the connection given by the CHS.

These examples show how important studying dyadic CHS is and how it can be deployed. However, how human-human interaction influences motor behavior is still largely unknown. Thus, our study wants to be a first step for future research in which the device we built will be studied in one of these specific fields. Hence, our investigation will mainly focus on the device and on a preliminary experiment done on healthy subjects. But since one of the main reasons for which this study has been started is to have, in future, a new rehabilitation therapy (even if it is only one of the possible uses of the device) that deploys dyadic CHS, in the following section there will be a brief overview on robotic therapy (and what are the positive effects of this type of paradigm with respect to traditional care) and on how the latter can be improved with dyadic interaction.

1.3 Stroke Rehabilitation

There are several factors that can define a rehabilitative process as successful or not. One of these is related to if the patient is treated or not in a stroke unit right after the stroke occurred. Indeed, it has been found that the patients treated in a stroke unit are, with higher probability, able to live in their houses after a year from the stroke and then to be independent. This is due to the fact that a stroke unit is an environment that gathers experts from different disciplines and technology that can not be found elsewhere. Another aspect that is considered during therapy is also how intense and with which frequency the therapy is given to the patient. Indeed, high repetitive and intensive therapies improves motor function after stroke. The problem of this aspect is that it requires a lot of effort from the caregivers which are not always able to follow the patients to ensure that they provide the therapy with so many repetitions. This is where robots can help the therapists in fulfilling this high repetition requirement without making the therapist work too much. Moreover, robots represent a reliable way to do some tasks (grasping, reaching, releasing etc.) in a repeatable way (which may be lacking in conventional therapy) with also the possibility of changing some variables (such as the resistance given by the robot to the movement) or implementing some experimental conditions that may enhance the outcome of the therapy (such as error augmentation, that has been discovered to enhance the adaptation, only if the error is augmented below a certain level [4]).

One of the most used therapy in the last decades and that can be considered a conventional therapy for upper extremity rehabilitation is the constrained-induced movement therapy (CIMT). This therapy has been found to be very effective in the outcome of the paretic arm. The protocol is constituted of three main phases:

- Intensive training of the impaired limb to promote task-specific use of the paretic limb.
- Constraining the non-impaired limb with a mitt to force the patient to use the paretic limb throughout the day.
- Train the patient to use the learned skills in the clinical environment and in the patient’s environment (so outside the clinical unit).

However, even if this therapy has been giving positive outcomes in terms of motor impairment recovery, the main limitation of CIMT is that high intensity trainings are difficult to be provided. That is where the robot therapy can be very effective and helpful in the clinical environment.

1.4 <u>Robots Rehabilitation</u>

The use of robots in the clinical field has been of great interests in the last decade. This technology provides high dosage and intensity training to the patients avoiding a direct contribution of the caregivers that can just follow how the patient is performing the task with the robot. This point really defines the main advantage that robots bring to the conventional rehabilitation of stroke: high dosage and intensity training have been shown to improve the impairment with a reduction of 5 points of the Fugl-Meyer scale (which is used to assess the limb impairment in stroke patients) and at the same time the workload on the caregivers is diminished significantly.

In the rehabilitation field, robots can be classified into two main categories:

- Exoskeletons: these can be considered as wearable robots since there is a direct contact between the patient and the robots' surfaces. The main aim of these devices is to reproduce as much as possible the joint movement. To do so, the joints and the axes of the exoskeleton should be aligned, as much as possible, with the anatomical ones. Considering these design specifications, an exoskeleton could be useful to manipulate limbs movement in their natural range of motion. In rehabilitation this technology can be exploited to reproduce the movements that the patient would do in a session with a therapist. Moreover, these devices are embedded with sensors that can provide quantitative information to the clinicians about the development of the therapy.
- End-effectors: these are connected to the patient just at the distal part of the impaired limb. Hence, the joints of the device do not match the anatomical ones. This leads to less control of the proximal segment of the joints but allows forces to be applied at the distal interface. This type of technology is easy to be set up and allows the implementation of different algorithms [5] that can alter the therapy condition (such as modifying the amount of assistance given to the patient or introduction of error augmentation factors).

In general terms, robotic therapy has been found to have better outcomes, with respect to usual care therapy, in terms of reduction of the impairment, in particular in chronic stroke patients [6], but it does not give significant improvements on motor function. Hence, using robot therapy alongside with usual care training, to translate impairment reduction into function, can be even more effective than using robot therapy alone. In fact, when the robotic and the conventional therapy are combined the outcomes in terms of Fuegl-Meyer scale are significantly better than the case of conventional therapy alone [7]. This type of paradigm allows to have a high intensity and duration training, given by the robot, enhanced with an improvement of the motor function, given by the traditional therapy.

Another advantage of using robot as rehabilitation tools is that, thanks to the advancements in controllers and to the implementation of different feedbacks (haptic, visual auditory etc.), different modalities can be realized:

- Active assisted mode, this type of modality brings to higher benefits with respect to a totally passive machine that does not show any real functional gain with respect to usual care therapy. This is a consequence of the lack of active movements and attention that the patient has during the training. Actually, attention is a main component in the beginning of brain reorganization and learning. However, passive machines might be exploited by the clinician as tools used to avoid secondary complications.
- Alteration of treatment conditions, a clear example of this is the use of error augmentation during training. In this case robot technology is operated alongside with an algorithm that exploits the natural processes of learning in the nervous system. In fact, error signals can increase learning (and re-learning) of motor skills through stimulation and activation of the neuroplasticity processes. Hence, error augmentation may lead to higher gains in terms of function recovery. The error augmentation used with repetitive training has been shown to produce better outcomes with respect to the repetitive training used alone [8]. Moreover, the gains were not only impairment-based but also functional-based with participants claiming a more frequent use of the impaired limb [8]. This shows how robot in rehabilitation can be exploited in different ways allowing a variety of modalities that can be chosen to maximize the outcomes of the therapy.
- Use of robot controllers to change adaptive settings. In this case robots are set to vary their behavior based on the patient, hence providing the best possible interaction with the patient to enhance as much as possible motor recovery. However, everything should be implemented to avoid passive guidance that has been shown to have no impact in the rehabilitation of stroke survivors.

Robots in this field can also be used as measuring tools. This allows clinician to have an objective evaluation of the patients' recovery. Furthermore, it permits researchers to have more insights on the processes involved in stroke recovery. For example, a study conducted through the MIT-Manus robot and the recording of kinematics of hand movement showed how continuous movements in stroke survivors can be seen as the composition of different sub-movements [9]. This has given more insights on the hypothesis that human movement is the result of primitive movements that overlap. This overlapping has been shown to be minimal in the first stages of recovery and this gives a better understanding on how movements are controlled at the central nervous system level and on how a specific damaged area of the central nervous system can affect the execution of a single sub-movement [9].

However, even if robots bring a lot of advantages in the clinical field, there is not a widespread use of these devices. The main reason for this is the cost. Their cost can vary between $75,000-$350,000 US dollars [5]. Hence, the development of new robots should have the aim of reducing the costs allowing an increase use of this type of devices in the clinical field (also in low and middle-income countries where the burden of stroke is very high).

In the actual state of the art our device, the Pantograph, can be considered a low-cost robot since it is designed without any controller and it just has two sensors. Moreover, it allows the creation of a new paradigm in which the human-human interaction complexity is exploited. This could lead to a further enhancement of rehabilitation outcome through robots.

1.5 Human-Human Interaction Studies

As seen before in this chapter, the human-robot paradigm has been proven to have positive effects on stroke rehabilitation outcomes. However, robots still present some drawbacks. Some of them may constrain too much the movement of the patients. In fact, a rehabilitative robot that does not allow to have a natural movement does not bring any improvement to the therapy with respect to the traditional care. For example, a robot called Lokomat (designed for gait rehabilitation) has been studied in a 5-year randomized clinical trial to compare it with traditional methods for gait rehabilitation [10]. It showed lower or equal outcomes with respect to the usual care and this is due to the fact that the robot does not allow a physiological movement of the pelvis (with lower frontal and sagittal plane movement) and patients have reported to have difficulty in swinging their arms when walking with the device [10].

Figure 1 The Lokomat device – Credits https://www.hocoma.com/solutions/lokomat-2/. This rehabilitation exoskeleton is used for lower limb therapies. The structure of this device constrains a lot the movement of the limbs, hence not allowing the patient to exploit the benefits of movement variability in motor learning. Picture: Hocoma, Switzerland.

The results of the study demonstrated how this type of devices, that limit too much the natural movement of the user, may perform equally or worse than traditional therapy for rehabilitation [10].

The limitations that may be imposed from a robot brings to this type of not significant outcomes because they limit two main factors in learning:

- Errors, as cited before the error augmentation has produced promising results in the clinical field [8] and error seems to be an important component of the learning process. With a robot that strictly fixes the possible movements, the patient is not able to complete the task trying new strategies.
- Variability, this is important in rehabilitation because fixed movement paths showed less normal spinal cord circuitry activation hence resulting in a diminished activity between the sensorimotor and spinal neural control system. This would hinder the re-learning process [11].

One solution to these drawbacks is to exploit human-human interactions. Robots that are designed based on interpersonal interaction principles could introduce that variability that can enhance therapies' outcomes. The problem is that a very few number of studies have been done on this subject, even if its importance may be remarkable in enhancing the already good results of robots by changing their design in order to make them behave more naturally when interacting with a human.

The first thing that has been defined to develop robots that reacts as similarly as possible to humans a classification of the different motor interactions that can occur between humans. This taxonomy allows to understand the mechanisms that are present in these interactions, then permitting a better understanding on how the agents' behavior can be modeled. The classification presented here has been defined by Jarrassé [12] and it is based on minimizing effort and error of each individual and on role assignment [12]:

- Competition: in this type of interaction every component minimizes its own error and effort without considering the ones of the partner (or partners). This type of interaction generally occurs in antagonist tasks.
- Collaborative: in this case each component also considers the error and the effort of the other components so that the task is completed in a way that benefits each component. In this case, there is not any role assignment prior to the beginning of the task. So roles are defined during the execution of the task, so there is an equal distribution of effort between the participants [13].
- Cooperative: as in the collaborative case the subjects work together to minimize effort and error but in this case the roles of each subject is defined before the beginning of the task and do not change during the execution of the task. In this case there is an uneven distribution of effort since, even if every participant is working to reach the same goal, each component is performing a different part of the same task. This type of interaction can be divided in other two subtypes: assistance, where the effort and error that are being minimized are the one of the person that is receiving assistance (in human-human interaction paradigms the haptic signals exchanged between the subjects can be used as a measure to understand the assistance that the other subject needs), and education, where there is a teacher-student role assignment where the teacher tries to minimize the error of the student so that the subject can perform the task alone (this type of interaction represents more the therapist-patient interaction that is present in the traditional therapy where the aim is to have functional improvements of the impaired limb).

This taxonomy is useful to understand the different types of interactions and how these can be implemented by using a robot.

A research study that has the aim of understanding more how human-human interaction occurs in dyadic type of interaction is the one by Reed and Patton [14]. Moreover, they also wanted to understand if two motor control systems are better than one. To do so, they developed a crank that connected two different individuals through two spinning handles. The task that the individuals were asked to perform was to move the handle to the target that was displayed on the screen and to hold the handle until a new target would appear. This task was done by dyads and by individuals alone.

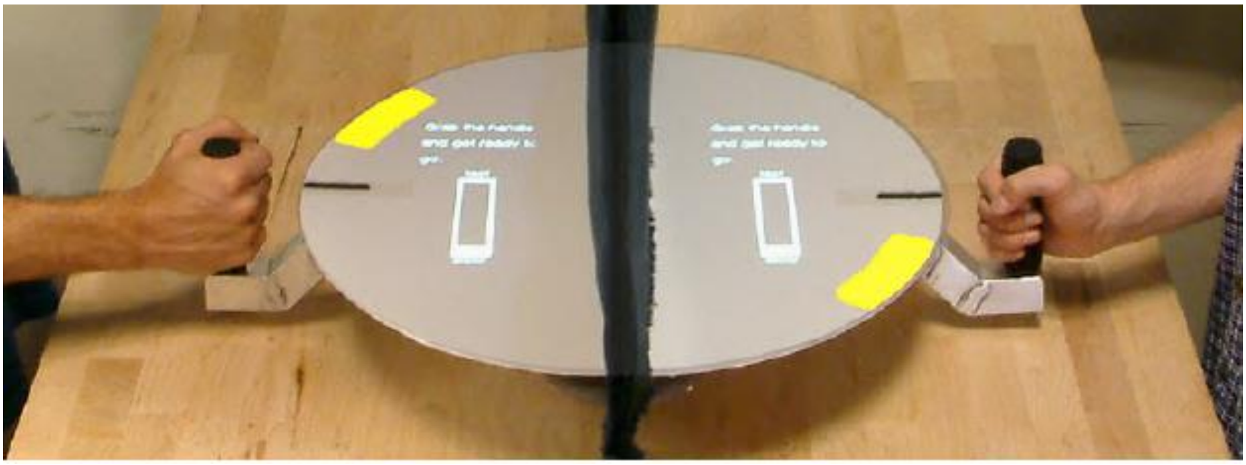

Figure 2 Device used in Reed and Patton's research study. It is constituted by two freely spinning handles that allow for 1D movements. On the top of the crank there is a screen that allows for the implementation of visual feedback for the execution of the task. Through this device the subjects were asked to perform a targeted reaching type of task. Credits for the picture: K. Reed and J. Patton et al. "Haptic cooperation between people, and between people and machines" International Conference on Intelligent Robots and Systems (IROS), 2006

To understand if dyads work together better than individuals alone, what they measured is the completion time of the task. What they found is that, except for two cases, the completion times for dyads were lower than the one found for individual tasks. This means that human-human interaction is beneficial to motor performance. And since the only way for the two subjects to communicate was through the device (during the task they could not see or talk to each other), this confirms the hypothesis for which the haptic communication that was engaged between the participants allow them to attain more information about the task, hence improving the performance.

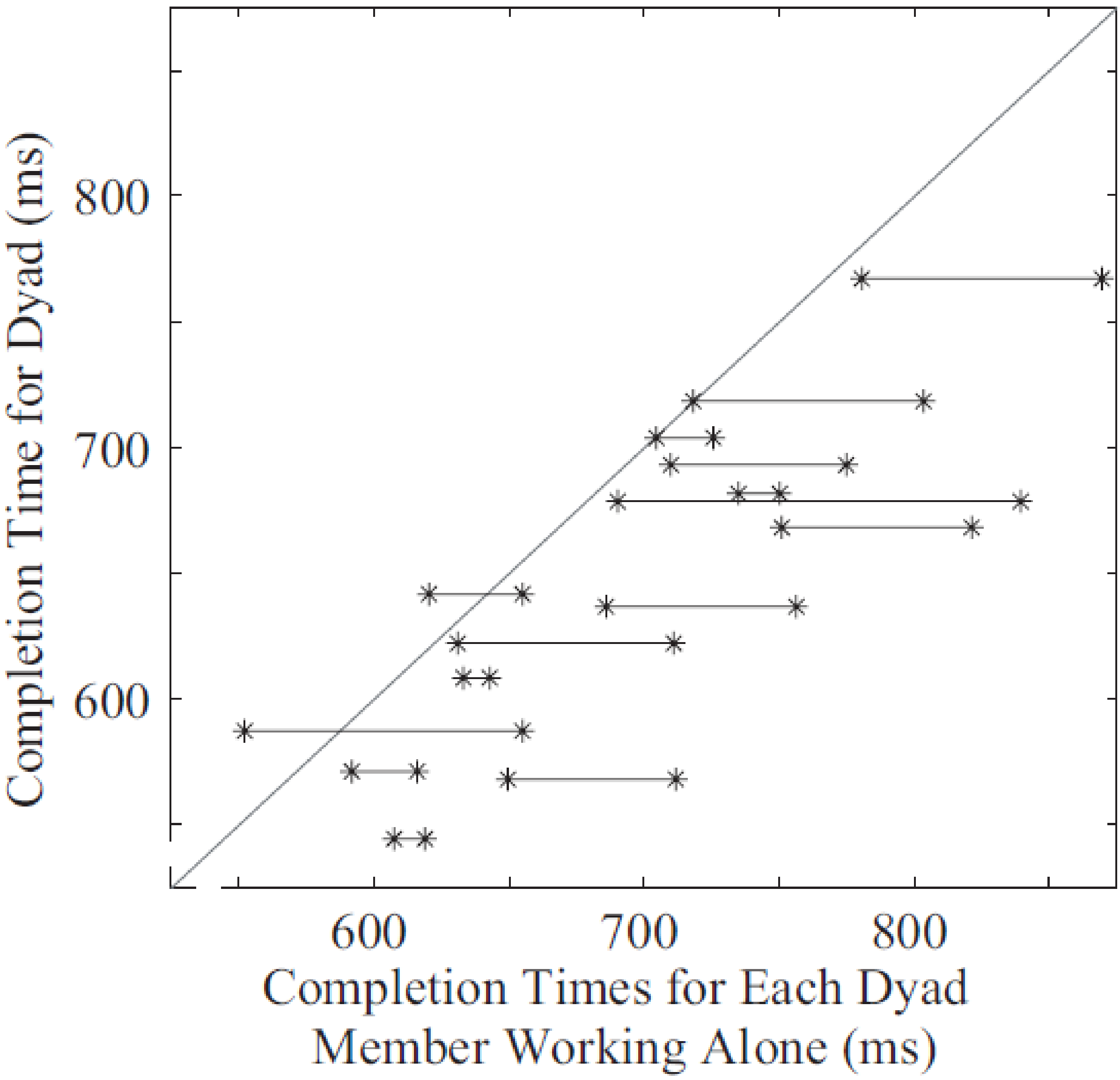

Figure 3 Completion times for dyads and for individuals alone (each individual is also a dyad member). Just two points are above the bisector line, this means that just in two cases the dyad performed worse than the individual. Hence, the motor performance benefits are higher in the dyadic type of interaction. These enhanced benefits are the consequence of the haptic communication that is engaged between the components of the dyads and that allows them to attain additional information about the task. Credits for the picture: K. Reed and J. Patton et al. "Haptic cooperation between people, and between people and machines" International Conference on Intelligent Robots and Systems (IROS), 2006

"Moreover, the force profiles recorded show that when working together, many dyads developed a new strategy that was not available to the members when they were working alone. Because the only interaction between subjects was a haptic one, they must have used this channel to develop a cooperative strategy" (K. Reed et al., 2006) [14].

One of the main research study that has been done on dyads is the one made by Ganesh [2]. This work has the aim of investigating motor responses that are given by the sensory feedback through haptic signals when human-human interaction occurs. The other type of response that is present in these interactions is the cognitive one which is related to the fact that the subject recognizes that there is an actual interaction with another person. The latter's effect has been excluded from this study to focus only on the motor adaptation driven by inter-personal physical interaction.

To quantitatively study the haptic signals' effect on dyads movement, a cyber human system was developed so that there was a connection between the individuals through a virtual elastic band which compliance could be changed. Each component of the dyad did not notice the presence of the other person. The task that was required to be performed was to track a target with a cursor controlled with the dual robotic system. The performance of the dyad is measured by analyzing the error which is defined as the mean distance, throughout the trial, between the cursor and target.

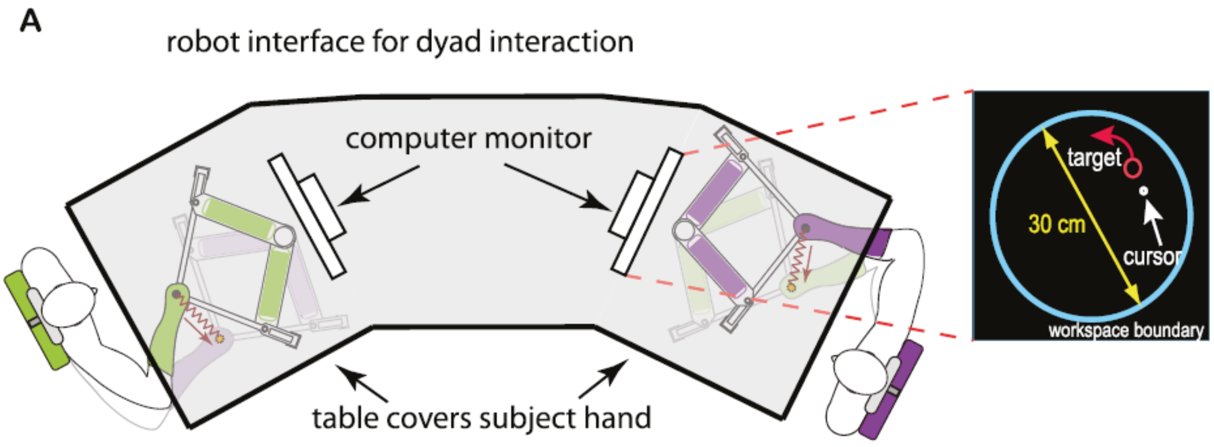

Figure 4 In this figure the dual robotic system created for Ganesh's study is depicted in this figure. A virtual elastic band connects the two subjects with a compliance that can be changed. In front of them a screen is placed, it shows the movement of the target and of the cursor (which is controlled by the dyadic interaction given by the dual robotic system). The task that the participants were asked to perform was a tracking target task. Credits for the picture: Ganesh, G. et al. Two is better than one: Physical interactions improve motor performance in humans. Sci. Rep. 4, 3824; DOI:10.1038/srep03824 (2014).

This new device has been used in different ways:

- Individual task, in this case the person involved is performing the task alone. This experiment was used to understand how much the motor performance improved and it is then used as a control experiment.
- Expert-Novice and Novice-Novice connection, in this case "expert" is considered a person that already know the task and can perform it with little error (lower than 1.3cm). These experiments were used to understand the influence of the nature of the interacting partner.

- Trajectory and force playback, where the interaction was between a human and a robot and it was predefined: the robot response emulated the one of a human that previously performed the task. These experiments were used to understand if the haptic cues given by the human interaction could be replaced.

This study shows that the human-human interaction is beneficial to the motor performance, regardless of the nature of the partner. Moreover, the benefits were higher than the ones registered when the task was performed individually. This means that the haptic communication between the two subjects allow to have more information about the task, hence improving the performance.

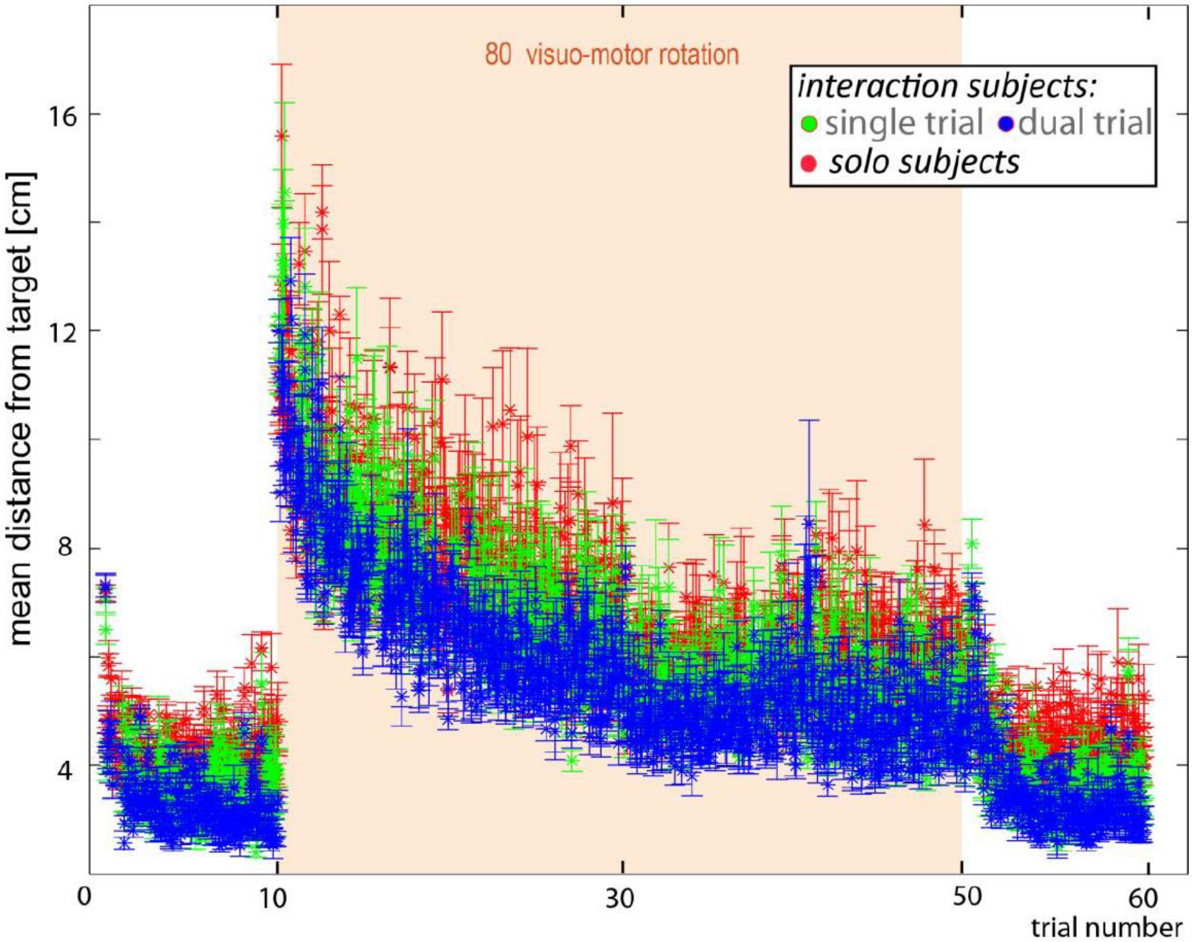

Figure 5 Trend of error (defined as mean distance between target and cursor during a trial) with respect of different trials. The blue trace shows how the performance in dyads is better than the one showed in the single trials. Moreover, dyadic tasks demonstrated the fastest way of learning. This means the time constants of the dual trials exponential decays are the lowest with respect to the ones of the single trials. Credits for the picture: Ganesh, G. et al. Two is better than one: Physical interactions improve motor performance in humans. Sci. Rep. 4, 3824; DOI:10.1038/srep03824 (2014).

What was also shown is that the nature of the partner may influence the outcome: more gains in terms of motor performance has been measured when the two subjects were similar to each other (which is in the case of the novice-novice experiment).

Finally, the trajectory and force playback experiments proved to be degrading the performance of the dyad [2]. Hence, the human-human interaction signals cannot give the same outcomes with only the robot design. What arose is that in this type of experiment is that there is a one-way connection which means that the subject could feel and respond to the action performed by the virtual partner but the latter could not change its response based on the one of the other component of the dyad. Hence, the user did not receive any haptic feedback and interaction. So, a two-way connection seems to be an essential factor that induces mutual motor benefits. This suggests that humans, during physical interactions, have implicit expectations in terms of the haptic forces from a partner and this allows to attain additional information about the task.

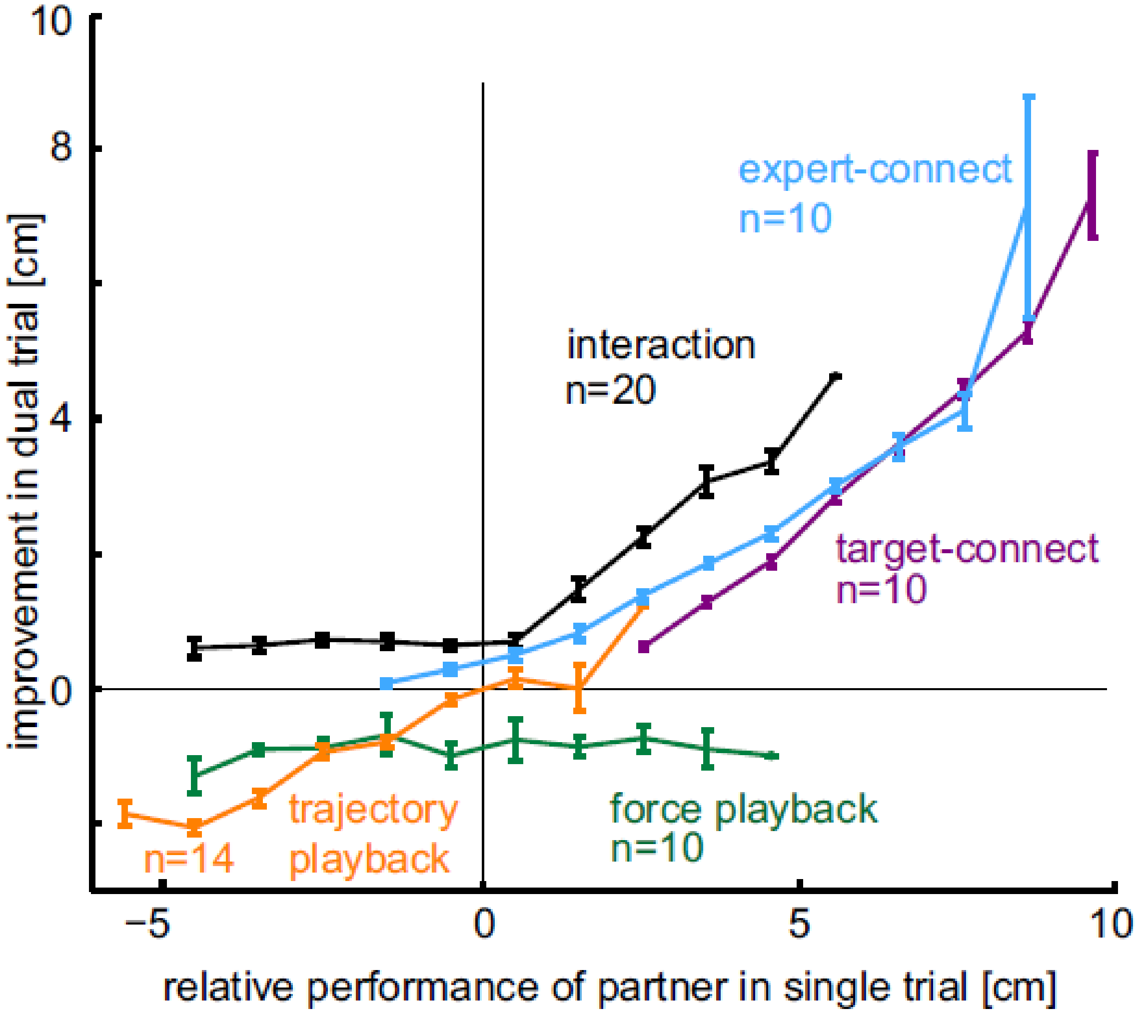

Figure 6 Representation of the motor performance improvement during different types of experiments. The green and orange traces show how the force and trajectory playback experiments mostly degrade the motor improvement. This is explained by the lack of haptic feedback in these types of experiment, where a one-way connection type of communication is used. Indeed, the subject could change his or her responses based on the shaptic signals that were applied on the handle but the latter would not change based on the subject's behavior. Credits for the picture: Ganesh, G. et al. Two is better than one: Physical interactions improve motor performance in humans. Sci. Rep. 4, 3824; DOI:10.1038/srep03824 (2014).

In conclusion, this study shows how dyadic interaction may be more beneficial than individual training for motor learning. It can be exploited not only in the rehabilitation field but also during sport practicing. This type of paradigm can offer also different approaches based on Jarrassé's classification [12], hence providing the best modality based on the impairment level of the patient.

Our study and device is a simpler version of this experiment and the aim of it is to understand if the same results can be obtained with a simpler device and how cognitive responses may influence the motor learning process that occurs in dyads.

CHAPTER 2

IMPLEMENTATION

Here the methods used to design the experiment for this study will be presented. There are two main components that has been designed:

- The Pantograph mechanical structure, which has been built using CAD software and machining of parts and that includes also two optical incremental rotary encoders.
- The data acquisition system that includes an Arduino board that is connected to the encoders, an algorithm that connects Matlab and Arduino, and a GUI implemented in Matlab that tracks the position of the handle through a virtual cursor showed on the screen.

In this chapter the design methods used for building the Pantograph and the reasons behind them will be presented first. After that, the connection between the mechanical part and the electronics (and its implementation) will be explained.

2.1 <u>Design and Building of the Pantograph</u>

The Pantograph is a mechanical system that presents two handles that are placed at the extremities of the device and are mechanically connected through six steel-tubes linked one another with seven steel-pins. There are four tubes (three for each handle) that are long 25.5cm and two center tubes that are long 51cm. The last two tubes are connected to each other by a common central pin which can be considered the center of rotation of the whole Pantograph, the shorter six tubes, instead, are connected to each other using three different pins: one is connected to the handle (and it used also as a support of the handle), while the other two link the tubes to the central ones. All these pins are conceived not only as a way to connect the different structures of the system to each other but also as the joints that allow the movement of the whole pantograph (and this is addressed by the central pin) and the relative movement between two different segments (which is addressed by the pins that link the central tubes to the shorter ones). The central pin is then constrained (through a press-fit method) to the rotation of the lower central tube so that it can rotate with the lower tube and then, through the use of a coupling, the pin is connected to the encoder. In this way it is possible to read accurately the rotation of the lower tube through the encoder, which rotation axis is aligned to the rotation axis of the Pantograph. The same is done for the upper tube, but in this case the constrained pin is used only for acquiring the rotation of the upper tube and it is not an element around which the pantograph rotates.

There are two incremental optical encoders that are connected to the Pantograph, as said before. One of them is linked to the lower central tube and it is blocked on the base on which the pantograph lays. The other one is instead connected to the upper tube and it is kept in place through a bridge made of 80/20 aluminum on which the support base of the encoder is attached. This avoids the translation of the encoders while the system is in use, hence providing a proper measurement of the angle of rotation.

The handles have been obtained by wooden balls in which a hole has been made to insert it into the pin that is then rigidly constrained to it. This choice allows also stroke patients, that may have a diminished ability in grasping a handle, to use the device. In fact, with this sphere-shaped handle the user just needs to lay the hand on it to be able to control the Pantograph.

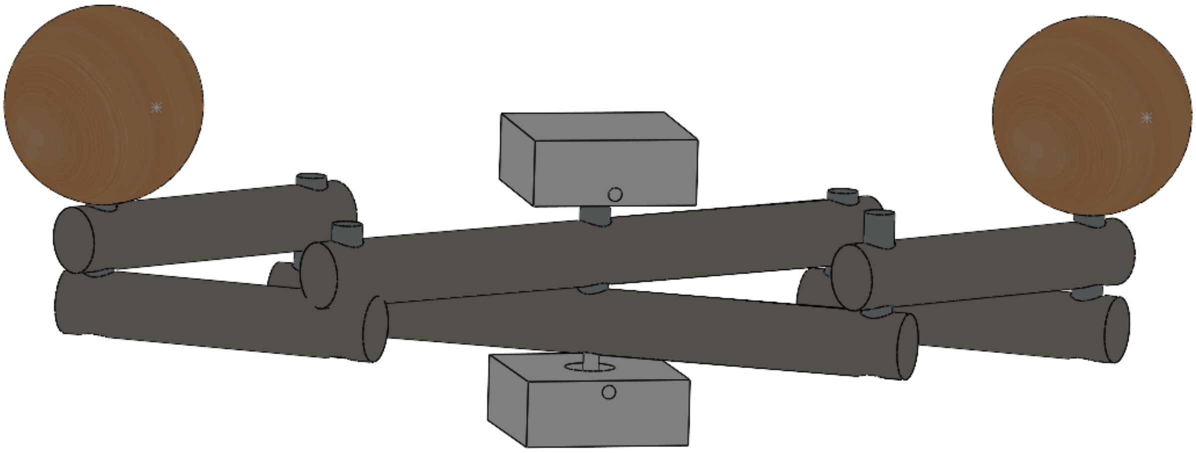

Figure 7 SolidWorks mockup of the pantograph. In this structure the six-steel tubes mechanism, that allows for high compliance between the users, can be seen clearly. The gray boxes represent the structures in which the encoders are inserted and at the extremities of the device the spherical wooden handles are fixed to the device through pins.

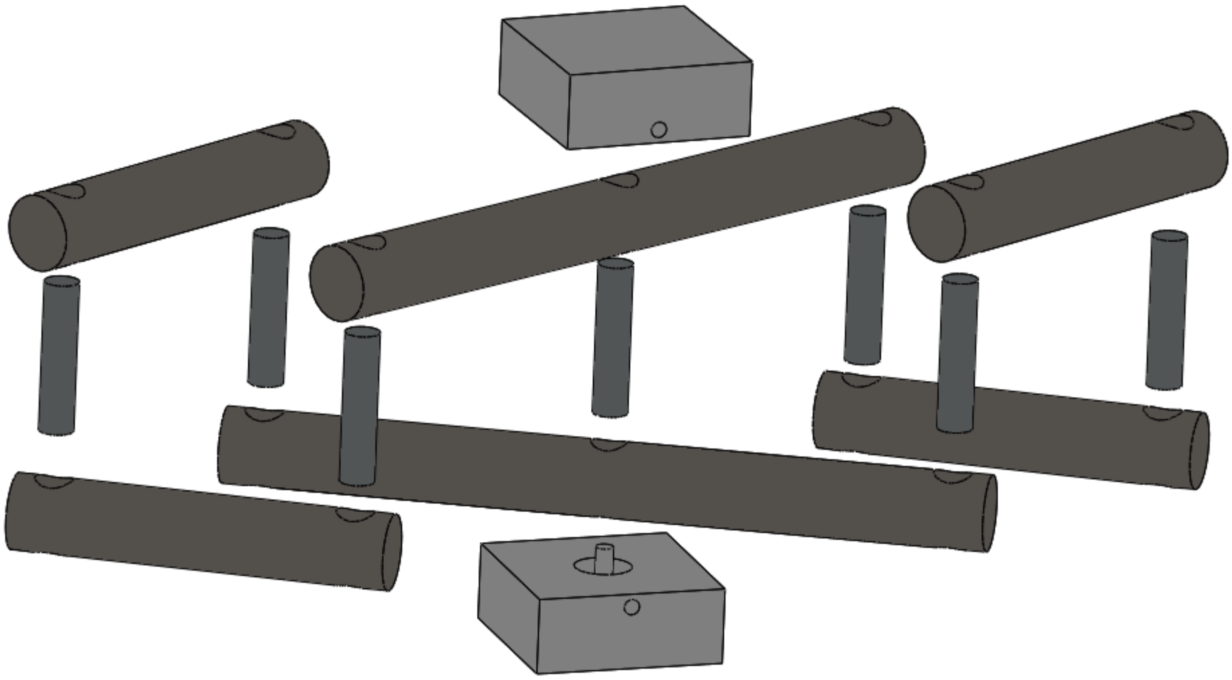

Figure 8 Disassembled Pantograph. In this figure the seven pins that connect the different beams are shown. The central one is the one around which the device rotates, the other ones (that connect different beams to each other) allows for relative rotation between different beams.

Everything then is supported by 80/20 aluminum bars. The pantograph is then attached to it and the base is heavy enough to avoid the device from moving while it is used.

The machinery used to build the Pantograph is:

- Band saw, to cut the tubes and the pins at the desired length.
- Drill press laser, to create the holes in the tubes.

The characteristics of the Pantograph have been defined considering the following design specifications:

- Friction between pins and beams has to be minimized. This has been achieved by creating holes in the tubes that have a diameter of 0.58cm. This allows to have some space between the pin and the tube (hence minimizing the friction) but creates some backslash that can cause the whole system to fail when sudden changes in direction occur. To have a null backlash, the pin and the tube should be rigidly constrained, but this would not allow the rotation of the CHS system around the pin. While to have null friction, there should not be any contact between the pin and the tube, but this would create a very high backlash. So, the diameter of the holes of 0.58cm has been found to be optimal.

- Lightweight design, so that it can be easily carried. The use of steel tubes has been found to be a good choice for this. At the same time, it must be safe for the user, that could be also a patient. The latter has been achieved by using steel (that even with relatively small sections, the outer diameter of the tubes is of 2.6cm, can provide enough resistance to the Pantograph) and minimizing the backlash. Moreover, the device is sitting on a 80/20 aluminum bar structure that gives the device enough strength to support the arms of the subjects during the execution of the experiment.
- Easy to use, which is a very important specification particularly in the clinical environment. For inpatients, this is necessary because an easy device saves a lot of time for the clinician and the patient. For outpatients, it makes the training sessions less difficult and that would involve more the patient in completing the required tasks at home. Also, it diminishes the experiment time.
- 2D movements must be allowed so that subjects can explore also more complex movements. This is accomplished by how the mechanical system has been designed and built.
- The resolution of the encoders must be around 1°. Since these are the only sensors that are embedded in the device it is essential that they have a high resolution, allowing to have more values of the angle of rotations, hence a more possible positions of the cursor on the screen. The encoders used on the CHS have a resolution of 0.15 degrees.
- The device must provide visual feedback to the user. This is done by mounting a screen on the device that shows the position of a cursor that is controlled by the user through the handle. The visual feedback is implemented in Matlab and it exploits the signal given by the encoders.
- Low cost. This specification is fulfilled considering that the device is mainly made by steel tubes (that cost around 8$ each) and 2 encoders (that cost 16$ each).

The following figures show the resulting robot that has been developed. On the top of it there is a screen through which the virtual reality, implemented in Matlab, is shown.

Figure 9 Top view of the Pantograph. In this figure the screen on which the virtual feedback is displayed can be seen. The screen is wide enough to avoid the users from seeing their own hands while executing the target tracking task. This does not permit them to have any point of reference while they are executing the task.

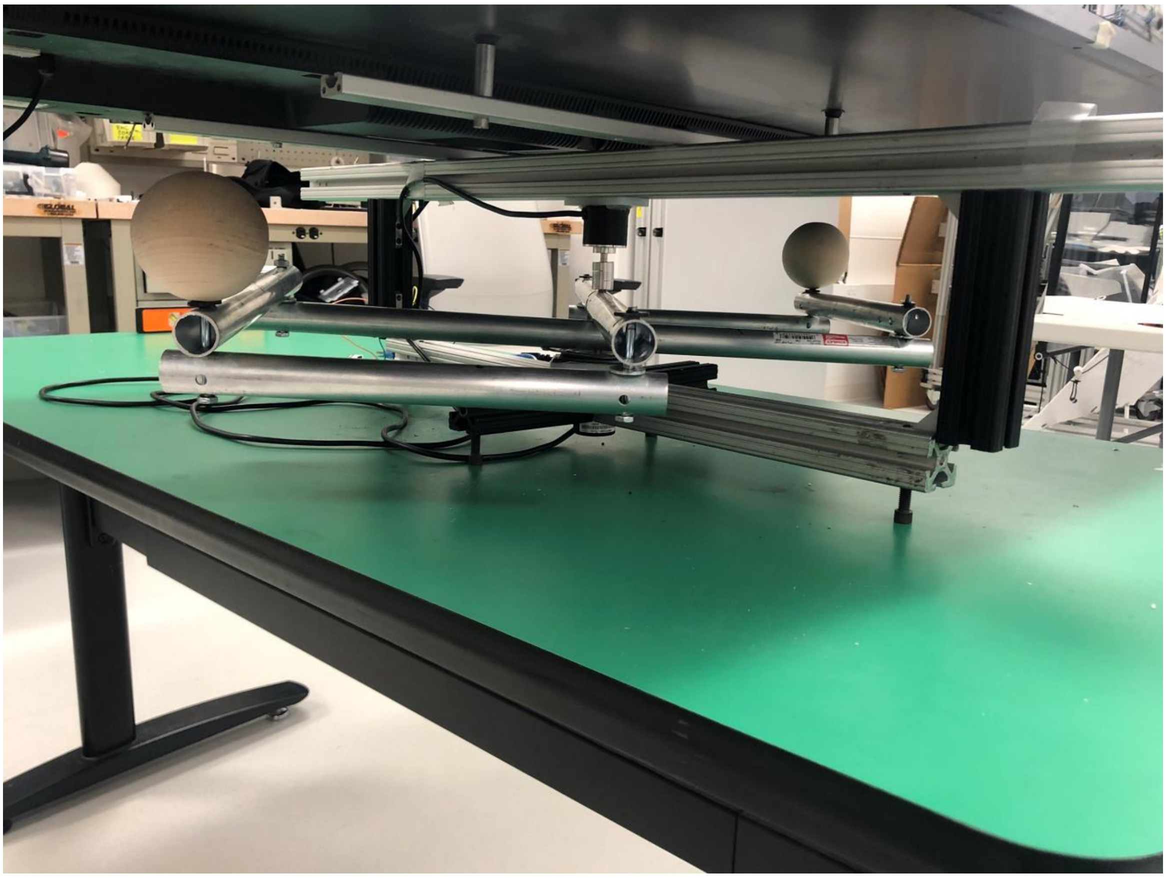

Figure 10 Side view of the Pantograph. The 80/20 aluminum bars are the basis on which the whole structure of the device sits. It has enough strength to support the weight of the subjects' limbs while they are performing the tracking task. Between the tubes and the encoders (in black in the figure) there is a coup-link (one of each encoder) that connects the pin to the encoder. This allows for higher accuracy in reading the rotation of the beams.

2.2 <u>Data Acquisition</u>

The data acquisition is made by an Arduino that reads the encoders' signals at an average sampling rate of 50Hz.

The incremental optical encoders provide a digital signal that counts how many times the rotational axis rotates. This signal is then sent to Matlab through serial communication and from Matlab the angle of rotation of the central tubes (that are connected to the encoders) is computed.

Using the angles that are given by the encoders it is then possible to solve the following forward kinematic equations from which it is possible to obtain the position of the cursor:

$$\boldsymbol{x} = 2 * \mathbf{a} * \mathbf{cosd}(\boldsymbol{\theta_2}) * \mathbf{sind}(\boldsymbol{\theta_1} + \boldsymbol{\theta_2})$$

$$\boldsymbol{y} = 2 * \mathbf{a} * \mathbf{cosd}(\boldsymbol{\theta_2}) * \mathbf{cosd}(\boldsymbol{\theta_1} + \boldsymbol{\theta_2})$$

Where:

- x and y are the position of the handle in the 2D space.
- a is the length of the shorter tube and it is equal to 25.5cm.
- θ_1 is the angle between the lower central pipe and the x-axis and it is computed by multiplying the resolution of the encoder (0.15°) with the digital signal of the lower encoder. Then 45° are subtracted to the results of the product to consider the angle at the initial condition.
- θ_2 is half of the angle between the central tubes and it is computed as half of the difference between the digital signals of the two encoders multiplied by the resolution (0.15°). Then 45° are added to the results of the product to consider the angle at the initial condition.

2.3 GUI

The GUI presents a start button that starts the application and a stop button that stops the application before the task is completed if needed.

When the start button is clicked the target will start moving on the screen and its position in time is defined by these functions:

$$\boldsymbol{x_T = 3 * sin(1.8 * t) + 3.4 * sin(1.8 * t) + 2.5 * sin(1.82 * t) + 4.3 * sin(2.34 * t)}$$

$$\boldsymbol{y_T = 3 * sin(1.1 * t) + 3.2 * sin(3.6 * t) + 3.8 * sin(2.5 * t) + 4.8 * sin(1.48 * t)}$$

Where:

- x_T and y_T are the position of the target in the 2D space
- t is the time vector and is defined as: $[t_i, t_i + 60]$ and $0 < t_i < 20$.

The equations that define the target movement have been obtained by [2].

The user will be asked to follow the target position with the cursor for a duration of 60 seconds for 60 different trials. After the 10^{th} trial a visuomotor rotation of 80° is switched on. This means that when the subjects move the device linearly, they will not see the cursor moving accordingly but it will instead rotate of 80° in a clockwise direction. After the 50^{th} trial the visuomotor rotation is turned off again.

The application is then connected to the Matlab code that computes the position of the handle in the 2D space so that these positions are then passed to the cursor that is shown on the screen.

While the application is running the position of the cursor is saved in a txt file with an average sampling frequency of 50Hz. With the same frequency the position of the target is saved in another file, even if the position of the target is already known. This permits to have the position of both cursor and target sampled at the same time moment, hence allowing to compare the positions of the two objects at the same time when analyzing the data. The txt files are organized in three columns:

- The first column is the time at which the position is sampled.
- The second column is the x-position.
- The third column is the y-position.

These files are then used as an input of another Matlab algorithm that has the purpose of analyzing the acquired data. The files names are given in this way: "CursorPosition_TrialNumber" for the cursor, while for the target is "TargetPosition_TrialNumber".

The following figure shows the GUI that has been implemented. The blue point represents the cursor that is controlled with the Pantograph while the red point it the target.

Figure 11 GUI. The red point is the target, and the blue point is the cursor. The cursor is controlled through the Pantograph, while the target is moving according to a multi-sine function that changes between different trials. There is a start and stop button, the latter allows to stop the task even before its completion. This can be useful in situations in which the users need to stop.

2.4 Data Processing and Analysis

The data analysis is based on the files that has been saved during the trials and that contain the position of the cursor, the position of the target (saved in a different file) and the time at which the position sample has been acquired.

The first thing that the algorithm does is to load the x and y positions of the target and the cursor. To do so, an optimized way to retrieve the data must be implemented to have a faster process. So, two vectors of dimension 60x1 is filled with 60 strings, one per each trial that has been done. Each string represents the name of the 60 files that has been saved with the position of target and cursor while the task was being performed. Then all these files are saved in two different structs. One is called "TargetData" (where the target positions files are saved) and the other is called "CursorData" (where the cursor positions files are saved). These two structs have 60 fields each. In each one a position file is saved and turned into a numeric array. In this way the data can be retrieved in an easier way throughout the code.

Then the data is pre-processed. Since the data acquisition system can be perturbed by different factors that may be related to the environment such as the computer to which the system is connected, the other application running on the same computer etc., the sampling frequency is not constant throughout the different trials and throughout the same trial. This may lead to different time steps during the same trial. To avoid this, resampling is being applied on the data. The latter allows to resample the data to a uniform rate. In this case the rate has been set to 50Hz. This new frequency allows us to have enough data points and it has been chosen considering that human motion occurs at a frequency between 10Hz and 20Hz. Hence, to avoid the aliasing event a sampling frequency higher that 40Hz must have been chosen. Moreover, a higher resampling frequency has not been chosen because the average sampling frequency of the data acquisition system is of 50Hz. Hence, having a higher resampling frequency would just slow down the code.

After that, the algorithm computes the velocity of motion of the cursor in the x and y directions by analyzing the change of position between two subsequent position samples and dividing it by the time interval that occurred between the two samples (which is constant). This computation leads to the creation of two (one for the x and one for the y direction) velocity vectors where each component of the matrix is computed with the following equation:

$$\boldsymbol{v_i} = \frac{\Delta \boldsymbol{s}}{\Delta \boldsymbol{t}}$$

Where:

- Δs is the change in position between subsequent samples.
- Δt is the time interval between subsequent samples.
- i is the direction of the velocity component (hence, x or y).

Then for each time instant the velocity is computed as:

$$v = \sqrt{v_x^2 + v_y^2}$$

Where:

- v_x is the x component of the velocity.
- v_y is the y component of the velocity.

The mean of the velocity vector is then computed, and this is done for each trial. So, a velocity threshold is set (and is lower than $0.5\,\frac{cm}{s}$) and all the velocities components that are lower than this threshold are used to recognize those trials that should be discarded. This is because those trials represent situation in which the user has not moved throughout the experiment.

Then, the error is computed as the absolute value of the difference between the cursor position and the target position (as defined in [2]):

$$\boldsymbol{error_i} = |\boldsymbol{s_{c,i}} - \boldsymbol{s_{T,i}}|$$

Where:

- s_c is the cursor position.
- s_T is the target position.
- i is the direction of the error component (hence, x or y).

Then the distance, which is used as a parameter to evaluate learning throughout different trials, is computed as:

$$distance = \sqrt{err_x^2 + err_y^2}$$

Where:

- err_x is the error on the x direction.
- err_y is the error on the y direction.

So, a “distance” vector is defined for each trial at each time sample. But to follow the method by Ganesh et al. [2], the average distance per each trial should be considered. Before calculating the mean of the distance, a filter is applied to the data. This averaging process is performed by averaging 5 seconds of data. Hence, having 12 data points per trial. This is performed thanks to the “reshape” function that creates a matrix in which each column represents 5 seconds of data, then the mean of each column of this matrix is computed and the result is saved in a vector. However, it must be considered that between different trials the acquisition frequency may vary. Then the number of points per each trial may not be a multiple of 5, hence the resulting matrix given by the “reshape” function may have some “NaN” values. So, to avoid this, the last column of the matrix is saved in a vector and the NaN values are erased from it. Hence allowing to compute the average of the last 5 seconds of data without giving any error.

Finally, the mean and standard deviation of the 12 data points (per each trial) is computed and saved in a vector. Hence, the components of this two columns vector will represent the mean and the standard deviation of each trial.

Then, the variation of the error between different trials is analyzed to see and define if there is an enhancement of the interaction between the two subjects, which means to investigate if the average distance between the cursor and the target decreases while performing going on with the trials.

So, what is done is to plot the tendency and spread of the data. Hence, per each trial the mean and the standard deviation of the distance between cursor and target is plotted through the “errorbar” function. On the y-axis the mean distance from the target is plotted while in the x-axis the number of trials is plotted. This curve will allow us to understand if learning have really occurred.

However, we need a quantitative parameter that allows us to understand which is the speed of learning. An exponential fitting of the data will permit us to retrieve this information through the reciprocal of the time constant of the curve. The exponential fitting has been done through a function that has no assumptions on the data points that are considered as the input, as opposed to the built-in functions of Matlab that needs data that goes to zero and that is not very noisy (which is not the case for the data we collected). The function that has been used finds the necessary parameters to best fit the following function:

$$f(x) = p_1 + p_2 * \exp\left(-\frac{x}{p_3}\right)$$

The exponential regression is computed for three different curves: one before the visuo-motor rotation is activated, one while the visuo-motor is activated and one after is turned off. After the three parameters p_1, p_2, p_3 have been computed, the interval of confidence at 95% is calculated. This permits us to understand, at a 95% confidence level, if our data is likely to be true. The formula used, per each parameter, is:

$$p_j = \tilde{p} \mp se * CritT$$

Where:

- p_j is the parameter computed (so it can represent the high or low limits of the parameter).
- $\tilde{p}$ is the parameter p_1 or p_2 or p_3.
- se is the standard error and has been computed as the ratio between the standard deviation of the data and the square root of the number of data points.
- $CritT$ is the critical T value of the data.

Another parameter that has been considered to evaluate the fitting curve is the sum of squares error. Through this evaluation is possible to understand how good the fitting curve is and it is useful to compare it with other fitting functions, hence to understand which one is better to be used.

2.5 <u>Protocol</u>

Participants will perform the tasks seated on a chair, since this study is focused only on the upper extremities. They will be asked to perform each trial of 60 seconds, “after each trial the target will be switched off and their hand will be passively returned to the center of the screen followed by a 20-30 seconds break” (G.

Ganesh, A. Takagi, 2014) [2]. After the 10^{th} trial the visuo-motor rotation will be switched on and turned off again after the 50^{th} trial. The participants will be asked to follow the movement of the target (which path will be pre-defined by a multi-sine function that changes between different trials) in the best way possible, which means they have to reduce as much as they can the distance between the cursor (that they are controlling) and the target.

The participants were not allowed to see their own hands, that were covered by the screen mounted on the Pantograph, so that they could not develop a strategy to complete the task that was different from exploiting the dyadic interaction. Moreover, they were not allowed to talk to each other.

To avoid the risk that one of the components of the dyad would execute the movement passively, they were told the had to complete the task as they were performing the task alone. This allowed to have data related to true human-human interaction, and not to a human passively following the movement of the other subject's limb.

CHAPTER 3

RESULTS

In this chapter the results retrieved from our preliminary experiment are presented. This data can not represent a scientific proof of how our device works, however it is a first step and a basis for future studies in which an IRB will be approved and in which a clear protocol will be followed.

The data has been acquired through an experiment done on my coworker, Mattia Demasi, and me.

The first result that can be analyzed is the device itself. The mechanism that connects the two subjects works so that if one moves, also the other one moves. The design specifications were all fulfilled, and this resulted in a device that has:

- Low friction between the moving parts of the robot. The movement of the device is fluid and allows to have a continuous movement of the arms of the subjects.
- A lightweight design with at the same time enough mechanical resistance to ensure safety during the execution of the task.
- Easy set up process. This allows to reduce the experiment time and get the subjects to be more involved in the use of the device.
- Allowance of 2D movements in space.
- Encoders with enough resolutions to provide a visual feedback to the subjects that is accurate.
- Low cost. The material used for the implementation of the device cost 112$.
- Handles that do not need too much grip to use the device. This could be important in future use of the device in the clinical environment.
- A range of motion of about 40cm on the vertical direction and of 60cm on the horizontal direction.

After completing and testing the device, the Pantograph has been used in a preliminary experiment. The experiment has been done following the protocol that has been described in chapter 2. The users were asked to perform 60 trials. The first 10 trials without any visuomotor rotation, then 40 trials with a visuomotor rotation of 80° in a clockwise direction and then the last 10 trials with the visual rotation switched off again.

The filtering process and the resampling process bring to a smooth position signal without any distortion of the signal. The resampling method created points that follow the signal.

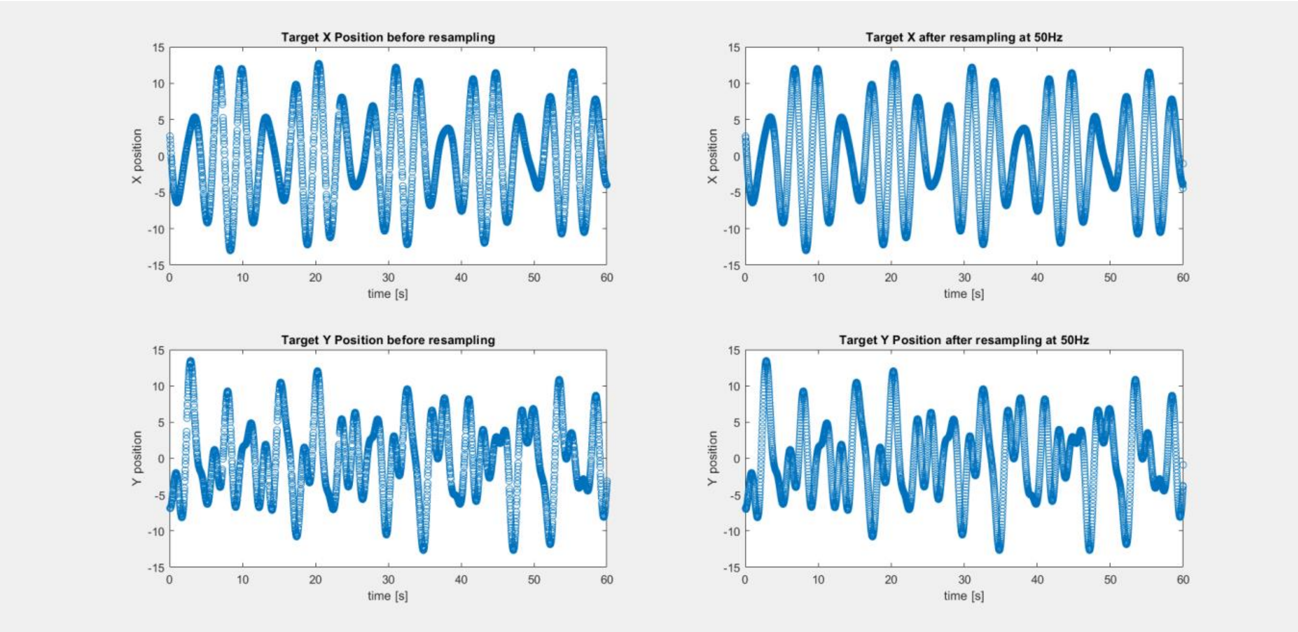

Figure 12 Position signal before and after the resampling process at 50Hz. The resampling process, as shown from the figure, allow to have a time step (between subsequent data points) that is constant during the whole duration of the experiment. Moreover, in the left part of the figure, it can be seen how some of the data points at certain time steps are missing. Hence, this means that in those instants the data acquisition system is missing some data that are then recovered thanks to the resampling method.

The data collected with the device showed that the acquisition system was able to give one cursor data point for each target point that was defined by the multi-sine function defined in section 2.3.

The code did not report any bad trial. The velocity threshold was set to 0.5 $\frac{cm}{s}$ and in none of the 60 trials the velocity was below this limit.

The measures of errors, through which performance is defined, show that there is a decreasing trend of the mean distance between cursor and target throughout the different trials. This decay is typical of learning processes and what has been considered is the average and the standard deviation of the error (defined as the distance between target and cursor during the experiment) per each trial. This allows to understand the tendency and the spread of the data during the 60 trials and hence to define if learning occurred or not.

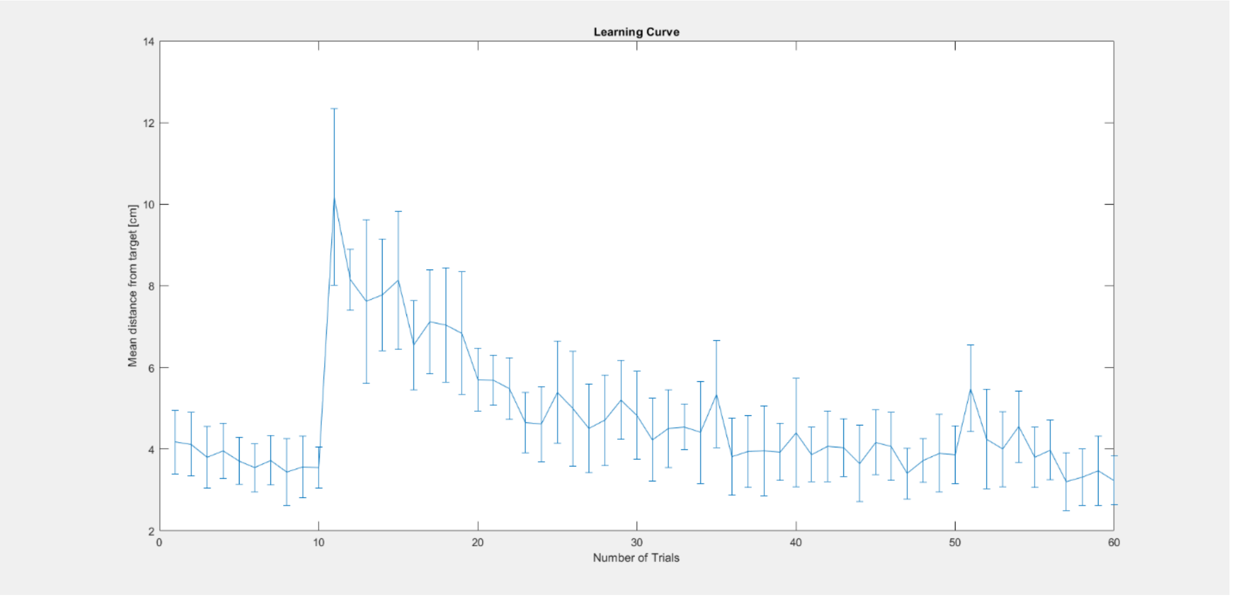

Figure 13 Mean distance between target and cursor throughout different trials. This plot allows to understand how error, which is a measure of performance in this study, varies with respect of different trials. The graph shows a decreasing error trend. Moreover, the shape of the curve is the one of a typical learning curve in which there is the activation of visuomotor rotation.

The shape of the curve is the one of a typical learning curve with the activation of visuomotor rotation in the middle of the experiment. However, the curve plotted like this does not give any further information about the learning process. Hence, fitting the curve to an exponential decay can give use more quantitative parameters. The exponential fitting process was done for three different curves:

- One for the first 10 trials (before the visuomotor rotation).
- One for the trial number 11 to the trial number 50 (during the visuomotor rotation).
- One for the last 10 trials (after the visuomotor rotation).

The three exponential curves that has been obtained are:

- $f_1 = 3.40 + 1.00 * \exp\left(-\frac{x}{4.48}\right)$
- $f_2 = 3.77 + 17.43 * \exp\left(-\frac{x}{9.69}\right)$
- $f_3 = 3.17 + 7020128.99 * \exp\left(-\frac{x}{3.39}\right)$

The main parameter that can be obtained by this analysis is the speed of learning. This can be computed by considering the inverse of the time constant of the exponential functions:

- $\tau_1 = 4.48$ trials
- $\tau_2 = 9.69$ trials

- $\tau_3 = 3.39$ trials

These values show how in the three different phases, learning occurred at different speeds.

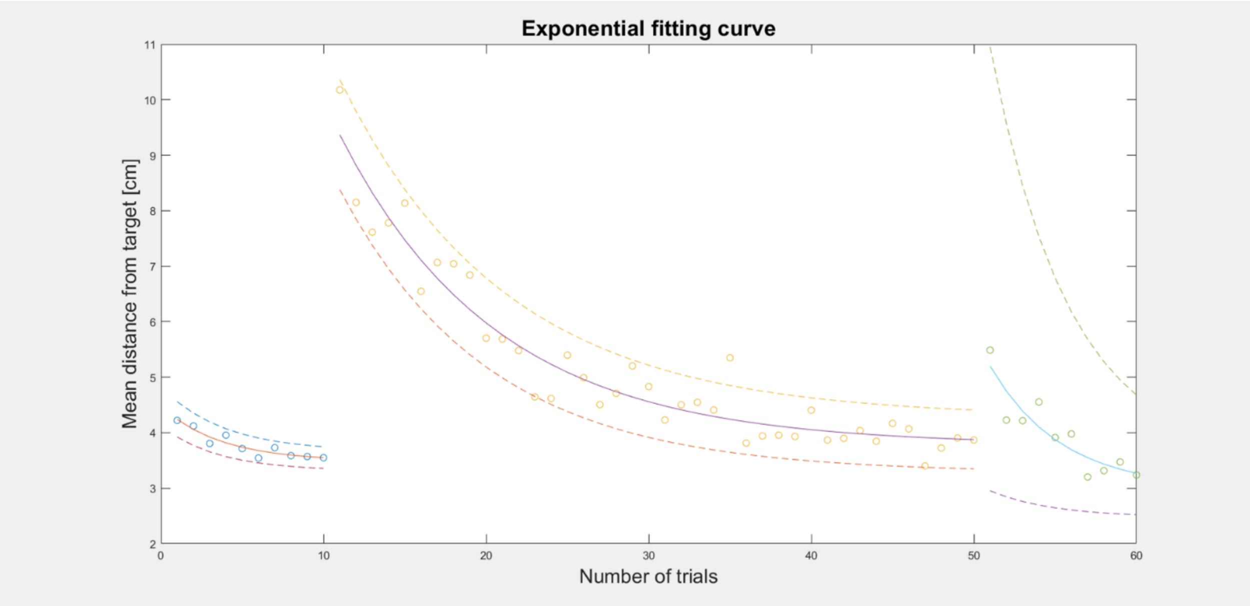

Figure 14 Exponential fitting curve of the data. The fitting process allows to understand if the data is decaying exponentially, and as it is shown from this figure the data, we attained is following an exponential decaying curve (as is typical of learning processes). Moreover, the interval of confidence at 95% is plotted with the dashed curves. Most of the data we collected belongs to the interval of confidence area, which shows how well the fitting process represents our data. Finally, this representation shows how the three different curves tend to a lower asymptote at different speeds.

The sum of squares error for each function is:

- $sse_1 = 0.087$
- $sse_2 = 7.0096$
- $sse_3 = 0.9028$

These three residuals values have been compared with the ones given by the fitting process performed by the built-in function of Matlab. The costs given by this latter are:

- $sse_1 = 0.12$
- $sse_2 = 16.00$
- $sse_3 = 1.12$

This comparison allowed us to understand that the built-in Matlab function is based on certain assumptions that are not fulfilled by the data we have.

CHAPTER 4

DISCUSSION

With this thesis we wanted to design and test a novel human-human interaction paradigm. We implemented a new device that mechanically connects two humans with the highest compliance possible. The system, called Pantograph, allows to:

- Have haptic feedback between the users.
- Have visual feedback: through a cursor the users can track the movement of the handle of the device.
- Acquire data on the position of the cursor through a data acquisition system made with an Arduino and two 16-bit absolute encoders.

With this study we wanted to give some insights on how human-human interaction can be beneficial to motor learning and we mainly wanted to demonstrate the usability of the new device we designed by using it in a pilot study.

In the experiment we performed, the device showed that the design specifications were good enough to ensure that all the trials were completed. The friction between the tubes is low enough to have a continuous and fluid movement of the device. Hence, the users do not need to apply a high force to move the arms of the device. The wooden ball handles allow the movement of the device without any need of gripping them. That can be a very good feature for future use of the device in the rehabilitation environment, in which some patients may not be able to grip the handle.

The protocol that has been used in the pilot experiment was taken from the paper written by Ganesh [2]. This will allow us in future studies to compare the results given by our device, with the ones that was obtained by Ganesh's experiment. This can lead to a higher degree of understanding on how the devices that are used to connect two or more subjects can influence the outcome of the motor learning process.

The data we found is very promising because it shows how even with our novel device, learning process occurs. As is it possible to see in figure 14 there are three different curves that are associated with three different speeds of learning. The last curve presents the lowest time constant, which means it was the phase in which subjects learned the motor skill in the fastest way possible. This means that in the last 10 trials the subjects were able to fully exploit the device and the human-human interaction signals to execute the task in the most efficient way. Indeed, it is the phase in which the error has its lowest values. However, the first part of the curve shows higher values. That is because there is a wash-out phase at the beginning in which the subjects have to re-learn how to use the device without the involvement of the visuo-motor rotation. In fact, the curve that goes from trial 11 to trial 50 has the highest values of error. In this part of the experiment the subjects have to learn how to use the device efficiently and how to move the cursor with the activation of the visuo-motor rotation to decrease as much as possible the distance between cursor and target. The summation of these two learning tasks brought to the highest errors and also to the lowest speed of learning as it is highlighted by the value of τ_2.

So, as shown by the results, a very simple device, as the one we developed, can lead to motor learning by interchanging haptics signals between two subjects. Even if cognitive signals were involved in this study, the dyadic interaction brings to motor adaptation. This supports the theory for which these haptic signals, trigger internal forward models that may benefit motor learning. Moreover, these results have been obtained with a very simple device with a very easy set-up process. This is an aspect that can be particularly important in the rehabilitation field, where the high cost of rehabilitation robots did not allow for their widespread use.

The main limit of the device is that it has not been made with high precision techniques. This has led to some minor imperfections that are the consequence of a slight vertical movement of the handles. This movement should not be present, but it can be easily fixed with finer machinery.

Moreover, the device can be improved with a more performant acquisition system. Even if the Arduino shows the capability of acquiring data at a rate higher than 40Hz (since human motion occurs between 10Hz and 20Hz, at least 40Hz of sampling rate must be ensured), the sampling rate is not constant between different trials. This happens because the system is easily susceptible to different factors since it is connected to a laptop. This led us to use the resampling method to have a homogenous time step between different data samples.

Another limitation of the device is that the virtual reality with which the users have to interact is not very interactive. And this can be critical since it may not involve the subject for the whole experiment, hence having a participant that is not actively moving the cursor. This is particularly important also in rehabilitative care where intensity and attention have been shown to be key factors in therapy outcomes.

A further improvement can be the addition of actuated pins that can change the level of coupling between the two subjects. This would allow to use the device to implement novel paradigms in which the interaction between the subjects is changed during the experiment intermittently. That could help understand if different levels of compliance bring to a better or worse motor adaptation.

Finally, strain gauges could be added to the center beams to measure the forces applied on them and subsequently to compute the forces applied to the handles. This would bring extra information on how the haptic signals are interchanged and which cooperative strategies are developed between the dyads' components.

For what concerns the study itself, the main limitation has been the fact the data was collected on us. Hence this data can not prove any hypothesis, but it can be used to prove the functioning of the new device.

Moreover, this study represents a first paradigm that can be used in future studies that involve a larger number of participants and that can focus not on the device itself but on how motor adaptation occurs in dyadic interaction and if this interaction could be beneficial enough to use such paradigm in a clinical context.

In conclusion, further studies should be done on this device to understand more consistently if the motor learning benefits found in this pilot study can be generalized. Also, more studies should be done to understand how the variability of the subjects (which could be two novices, or a novice coupled with an expert) that compose the dyad can affect the motor learning and performance.

Our aim is to continue this study by improving the device first and by using it in experiments that allow to collect more data and more information on how human-human interaction occurs and can be deployed.

APPENDICES

APPENDIX A

DATA ANALYSIS CODE – Matlab

```
%% Load Data
%Path to get to the results folder
Path = 'Results/Trial3MayWithPantograph/';
%Vectors of strings containing the results file
numTrials = 60;
CursorFileNames = strings(numTrials,1); %Creating a matrix with strings
TargetFileNames = strings(numTrials,1);
CursorBase = 'CursorPosition_';
TargetBase = 'TargetPosition_';
%filling the string matrix with the file names
for i = 1:numTrials
      CursorFileNames(i) = strcat(CursorBase,num2str(i),'.txt');
      TargetFileNames(i) = strcat(TargetBase,num2str(i),'.txt');
end
%Get the data and order it in a struct (one for the Target and one
%for the Cursor) in which each element represents a trial
Field = 'Trial';
CursorData = [];
TargetData = [];
average_err = zeros(numTrials,1);
for i = 1:numTrials
      CursorData1 =
         struct(Field,str2num(caseread(strcat(Path,CursorFileNames(i)))));
      CursorData = [CursorData, CursorData1];
      TargetData1 =
         struct(Field,str2num(caseread(strcat(Path,TargetFileNames(i)))));
      TargetData = [TargetData, TargetData1];
end
%% Process the data
%average_sampling_f is a vector in which the average sampling frequency of
```

```
%each trial is saved.

m_s = zeros(numTrials,2);

average_sampling_f = zeros(60,1);
f_variation = zeros(numTrials,1);
f_variation_noR = zeros(numTrials,1);

%This part of random numbers is used for the plots (look at the plot section)

r1 = round((10-1)*rand(1) + 1); %Random number between 1 and 10

r2 = round((50-11)*rand(1) + 11); %Random number between 11 and 50
r3 = round((60-51)*rand(1) + 51); %Random number between 51 and 60

%This vectore is used in the “bad trials check” section

average_velocity = zeros (numTrials,1);

for i = 1:numTrials

    %Save the position data in a Matrix

    CursorPos = CursorData(i).Trial;

    TargetPos = TargetData(i).Trial;

    %Read the matrix Results which is structured in this way

    %Fisrt column: time
    %Second column: x points
    %Third Point: y points

    time = TargetPos(:,1);

    %Resampling: in this way I am going to obtain a constant sampling

    %frequency throughout the whole experiment

    [Xpos_C, tx_c] = resample (CursorPos(1:end,2), (CursorPos(:,1)), 100);

    [Ypos_C, ty_c] = resample (CursorPos(1:end,3), (CursorPos(:,1)), 100);

    [Xpos_T, tx_t]  = resample (TargetPos(1:end,2), (TargetPos(:,1)), 100);
```

```
n. [Ypos_T, ty_t]  = resample (TargetPos(1:end,3), (TargetPos(:,1)), 100);

o. %Compute the error in X and Y, defined as the X and Y distance

p. %of the cursor from the target

q. errX = zeros(size(tx_c));

r. errY = zeros(size(tx_c));

s. for idx=1:length(tx_c)

      i.  errX(idx) = abs(Xpos_C(idx) - Xpos_T(idx));
     ii.  errY(idx) = abs(Ypos_C(idx) - Ypos_T(idx));

t. end

u. %Define then the distance between cursor and target,

v. %which will be the value that at the end you are going to use as the
w. %error and as the parameter to assess the learning process

x. distance = zeros(size(errX));

y. for idx = 1:length(errX)

      i.  distance(idx) = sqrt((errX(idx)^2)+(errY(idx)^2));
z. end

aa.if (i == r1)

      i.  distance_r1 = distance;
     ii.  time_r1 = tx_c;
    iii.  Xpos_C_r1 = Xpos_C;
     iv.  Ypos_C_r1 = Ypos_C;
      v.  Xpos_T_r1 = Xpos_T;
     vi.  Ypos_T_r1 = Ypos_T;
    vii.  else if (i == r2)
   viii.  distance_r2 = distance;
     ix.  time_r2 = tx_c;
      x.  Xpos_C_r2 = Xpos_C;
     xi.  Ypos_C_r2 = Ypos_C;
    xii.  Xpos_T_r2 = Xpos_T;
   xiii.  Ypos_T_r2 = Ypos_T;
    xiv.  else if (i == r3)
            1. distance_r3 = distance;
            2. time_r3 = tx_c;
            3. Xpos_C_r3 = Xpos_C;
            4. Ypos_C_r3 = Ypos_C;
            5. os_T_r3 = Xpos_T;
            6. Ypos_T_r3 = Ypos_T;
     xv.  end
    xvi.  end
bb.end

cc.%Filtering the data
```

```
dd.%Perform a 5 seconds averaging process: for every 5 seconds of data an
ee.%average of that data is performed
ff.%In this part of the code the "out" matrix is defined. In this matrix
gg.%each column represent 5 seconds of data

hh.average_sampling_f(i) = size(distance,1)/60;


ii.n_samples = round(average_sampling_f(i) * 5);
jj.out = reshape([distance;nan(mod(-
   numel(distance),n_samples),1)],n_samples,[]);
kk.err_averaged = zeros(size(out,2),1);

ll.%So each column of out represents 5 seconds of data, these 5 seconds of


mm.%data are saved in an element of err_averaged. So the first component
nn.%of err_averaged represents the average value of the first 5 seconds of
oo.%data and so on.

pp.for idx = 1☹size(out,2)-1)


      i.  err_averaged(idx,1) = mean(out(:,idx));
qq.end

rr.%Since the frequency may vary,


ss.%then the number of point per each trial may not be a multiple of the
tt.%number of samples in 5 seconds (so the last column of the "out" matrix
uu.%may have some NaN values).
vv.%Hence, to avoid errors in the code, the vector "last_vector" is used
ww.%to save the last column of "out". Then the NaN values are erased from
xx.%the vector. Then the mean of last_vector is computed and substituted
yy.%as the last value of "err_averaged"

zz.last_vector = [];


aaa.      for idx = 1:size(out,1)
      i.  if (~isnan(out(idx, end)))
     ii.  last_vector = [last_vector out(idx,end)];
    iii.  else
     iv.  break
      v.  end
bbb.      end

ccc.      err_averaged(end,1) = mean(last_vector);



ddd.      %So err_averaged is a vector with different rows (one per each
   column


eee.      %of "out") and each value is the average of 5 seconds of data.
fff.      %From this vector an overall average and standard deviation of its
ggg.      %values are computed (so we will end up with one average value and
   with
hhh.      %one std value per each trial) and saved in m_s.
iii.      %m_s (mean_standarddeviation) is a numTrialsx2 matrix. In the
   first
```

```
jjj.     %column the average of each trial is saved, while in the second
kkk.     %column the std per each trial is saved

lll.     avg = mean(err_averaged);


mmm.     stn_dev = std(err_averaged);
nnn.     m_s (i,1) = avg;
ooo.     m_s (i,2) = stn_dev;

ppp.     %In average_err the values of the average distance, computed
   without


qqq.     %performing a 5 seconds averaging, are saved to check the
   difference
rrr.     %with the error computed with the 5 seconds averaging

sss.     average_err(i) = mean(distance);



ttt.     %% Frequency dropout



uuu.     %Here we are going to check how the frequency changes throughout
   the


vvv.     %trial with resampling

www.     freq = 1./(diff(tx_c));


xxx.     min_freq = min(freq);
yyy.     max_freq = max(freq);
zzz.     f_variation (i) = max_freq - min_freq;

aaaa.    %Here we are going to check how the frequency changes throughout
   the


bbbb.    %trial without resampling

cccc.    freq_noR = 1./(diff(time));


dddd.    min_freq_noR = min(freq_noR);
eeee.    max_freq_noR = max(freq_noR);
ffff.    f_variation_noR (i) = max_freq_noR - min_freq_noR;

gggg.    %% Velocity of the cursor computation



hhhh.    deltaX = zeros(size(tx_c)-1);


iiii.    deltaY = zeros(size(tx_c)-1);
jjjj.    deltaT = zeros(size(tx_c)-1);
kkkk.    velocity = zeros(size(tx_c)-1);

llll.    for idx = 1:length(tx_c)-1
```

```
deltaX(idx) = Xpos_C(idx+1) - Xpos_C(idx);
deltaY(idx) = Ypos_C(idx+1) - Ypos_C(idx);
deltaT(idx) = tx_c(idx+1) - tx_c(idx);

Vx = (deltaX(idx))/deltaT(idx);

Vy = (deltaY(idx))/deltaT(idx);

velocity(idx) = sqrt((Vx^2) + (Vy^2));

end

average_velocity(i) = mean(velocity);

end

%% Plots

% Error vs time in 3 trials

% Which trials do we choose? One between trial 1 and trial 10 (before VM)
% One between trial 11 and trial 50 (during VM)
% One between trial 51 and trial 60 (after VM)

figure

plot(time_r1,distance_r1)
title (['Error plot at trial number ' num2str(r1)])
xlabel('time [s]')
ylabel('error [cm]')
figure
plot(time_r2,distance_r2)
title (['Error plot at trial number ' num2str(r2)])
xlabel('time [s]')
ylabel('error [cm]')
figure
plot(time_r3,distance_r3)
title (['Error plot at trial number ' num2str(r3)])
xlabel('time [s]')
ylabel('error [cm]')

% Cursor and Target X and Y positions vs time in 3 trials

% Which trials do we choose? One between trial 1 and trial 10 (before VM)
% One between trial 11 and trial 50 (during VM)
% One between trial 51 and trial 60 (after VM)

figure

subplot(1,2,1)
plot(time_r1,Xpos_T_r1)
hold on
plot(time_r1,Xpos_C_r1)
```

```
title (['X position of target and cursor at trial number ' num2str(r1)])
xlabel('time [s]')
ylabel('X position [cm]')
legend ('Target X Position', 'Cursor X Position')
subplot(1,2,2)
plot(time_r1,Ypos_T_r1)
hold on
plot(time_r1,Ypos_C_r1)
title (['Y position of target and cursor at trial number ' num2str(r1)])
xlabel('time [s]')
ylabel('Y position [cm]')
legend ('Target Y Position', 'Cursor Y Position')
```

```
figure
```

```
subplot(1,2,1)
plot(time_r2,Xpos_T_r2)
hold on
plot(time_r2,Xpos_C_r2)
title (['X position of target and cursor at trial number ' num2str(r2)])
xlabel('time [s]')
ylabel('X position [cm]')
legend ('Target X Position', 'Cursor X Position')
subplot(1,2,2)
plot(time_r2,Ypos_T_r2)
hold on
plot(time_r2,Ypos_C_r2)
title (['Y position of target and cursor at trial number ' num2str(r2)])
xlabel('time [s]')
ylabel('Y position [cm]')
legend ('Target Y Position', 'Cursor Y Position')
```

```
figure
```

```
subplot(1,2,1)
plot(time_r3,Xpos_T_r3)
hold on
plot(time_r3,Xpos_C_r3)
title (['X position of target and cursor at trial number ' num2str(r3)])
xlabel('time [s]')
ylabel('X position [cm]')
legend ('Target X Position', 'Cursor X Position')
subplot(1,2,2)
plot(time_r3,Ypos_T_r3)
hold on
plot(time_r3,Ypos_C_r3)
title (['Y position of target and cursor at trial number ' num2str(r3)])
xlabel('time [s]')
ylabel('Y position [cm]')
legend ('Target Y Position', 'Cursor Y Position')
```

```
%% Bad trials check
```

```
%A velocity vector stores the average velocity per each trial
```

```
velocity_threshold = 20; %which means no movement 43onfiden
```

```
bad_trial_number = zeros(length(average_velocity),1);
```

```
for idx = 1:length(average_velocity)
if (average_velocity (idx) < velocity_threshold)
bad_trial_number(idx) = idx;
end
end

%% Plot of tendency and spread

% Error and std vs Number of trial

% This plot represents the learning curve

figure

errorbar(1:numTrials, m_s(:,1), m_s(:,2));
title('Learning Curve')
xlabel('Number of Trials')
ylabel('Mean distance from target [cm]')

%% Fitting the learning curve (with Jim's library)

[p1, cost1, f1Exp] = expRegression([1,1,1],(1:10)',m_s(1:10,1));

f1 = @(x) p1(1) + p1(2)*exp(-x/p1(3));

[p2, cost2, f2Exp] = expRegression([1,1,1],(11:50)',m_s(11:50,1));

f2 = @(x) p2(1) + p2(2)*exp(-x/p2(3));

[p3, cost3, f3Exp] = expRegression([1,1,1],(51:60)',m_s(51:60,1));

f3 = @(x) p3(1) + p3(2)*exp(-x/p3(3));

figure

plot((1:10)', m_s(1:10,1), 'o')
hold on
plot ((1:10), f1(1:10))
hold on

plot((11:50)', m_s(11:50,1), 'o')

hold on
plot ((11:50), f2(11:50))
hold on

plot((51:60)', m_s(51:60,1), 'o')

hold on
plot ((51:60), f3(51:60))
hold on

% Compute 44onfidence interval at 95%
```

```
alfa = 0.05;
p1_CI = zeros(2,3);
p2_CI = zeros(2,3);
p3_CI = zeros(2,3);
se_1 = std(m_s(1:10,1))/sqrt(length(m_s(1:10,1)));
se_2 = std(m_s(11:50,1))/sqrt(length(m_s(11:50,1)));
se_3 = std(m_s(51:60,1))/sqrt(length(m_s(51:60,1)));
% For the first curve
p1_CI(1,1) = p1(1) - se_1*CritT(alfa, length(m_s(1:10,1)) - 1, 'two');
p1_CI(2,1) = p1(1) + se_1*CritT(alfa, length(m_s(1:10,1)) - 1, 'two');
p1_CI(1,2) = p1(2) - se_1*CritT(alfa, length(m_s(1:10,1)) - 1, 'two');
p1_CI(2,2) = p1(2) + se_1*CritT(alfa, length(m_s(1:10,1)) - 1, 'two');
p1_CI(1,3) = p1(3) - se_1*CritT(alfa, length(m_s(1:10,1)) - 1, 'two');
p1_CI(2,3) = p1(3) + se_1*CritT(alfa, length(m_s(1:10,1)) - 1, 'two');
%For the second curve
p2_CI(1,1) = p2(1) - se_2*CritT(alfa, length(m_s(11:50,1)) - 1, 'two');
p2_CI(2,1) = p2(1) + se_2*CritT(alfa, length(m_s(11:50,1)) - 1, 'two');
p2_CI(1,2) = p2(2) - se_2*CritT(alfa, length(m_s(11:50,1)) - 1, 'two');
p2_CI(2,2) = p2(2) + se_2*CritT(alfa, length(m_s(11:50,1)) - 1, 'two');
p2_CI(1,3) = p2(3) - se_2*CritT(alfa, length(m_s(11:50,1)) - 1, 'two');
p2_CI(2,3) = p2(3) + se_2*CritT(alfa, length(m_s(11:50,1)) - 1, 'two');
%For the Third curve
p3_CI(1,1) = p3(1) - se_3*CritT(alfa, length(m_s(51:60,1)) - 1, 'two');
```

```
p3_CI(2,1) = p3(1) + se_3*CritT(alfa, length(m_s(51:60,1)) - 1, 'two');
p3_CI(1,2) = p3(2) - se_3*CritT(alfa, length(m_s(51:60,1)) - 1, 'two');
p3_CI(2,2) = p3(2) + se_3*CritT(alfa, length(m_s(51:60,1)) - 1, 'two');
p3_CI(1,3) = p3(3) - se_3*CritT(alfa, length(m_s(51:60,1)) - 1, 'two');
p3_CI(2,3) = p3(3) + se_3*CritT(alfa, length(m_s(51:60,1)) - 1, 'two');
%Interval of confidence functions
f1_min = @(x) p1_CI(1,1) + p1_CI(1,2)*exp(-x/p1_CI(1,3));
f1_max = @(x) p1_CI(2,1) + p1_CI(2,2)*exp(-x/p1_CI(2,3));
hold on
plot ((1:10), f1_min(1:10), '-')
hold on
plot ((1:10), f1_max(1:10), '-')
hold on
f2_min = @(x) p2_CI(1,1) + p2_CI(1,2)*exp(-x/p2_CI(1,3));
f2_max = @(x) p2_CI(2,1) + p2_CI(2,2)*exp(-x/p2_CI(2,3));
hold on
plot ((11:50), f2_min(11:50), '-')
hold on
plot ((11:50), f2_max(11:50), '-')
hold on
f3_min = @(x) p3_CI(1,1) + p3_CI(1,2)*exp(-x/p3_CI(1,3));
f3_max = @(x) p3_CI(2,1) + p3_CI(2,2)*exp(-x/p3_CI(2,3));
hold on
plot ((51:60), f3_min(51:60), '-')
hold on
plot ((51:60), f3_max(51:60), '-')
hold on
```

APPENDIX B

DATA ACQUISITION CODE – Arduino

```
// Red - 5V
// Black - GND
const int encoder_a = 2; // Green - pin 2 - Digital
const int encoder_b = 3; // White - pin 3 - Digital
long encoder = 0;
float newEnc = 0;
```

```
void setup() {
```

```
Serial.begin(9600);
pinMode(encoder_a, INPUT_PULLUP);
pinMode(encoder_b, INPUT_PULLUP);
attachInterrupt(0, encoderPinChangeA, CHANGE);
attachInterrupt(1, encoderPinChangeB, CHANGE);
}
```

```
void loop() {
```

```
// a = (P/PPR)*360
// a = angle in degrees
//P = data we are acquiring (encoder)
// PPR = pulses per revolution, in this case 600
// 360 = 360 degrees
if (encoder > 2400 || encoder < -2400 )
encoder = 0;
if (encoder != newEnc){
newEnc = encoder;
```

```
float degree = (encoder/2400.00)*360.00;

Serial.println(degree);

}

}
```

```
void encoderPinChangeA() {
```

```
encoder += digitalRead(encoder_a) == digitalRead(encoder_b) ? -1 : 1;

}
```

```
void encoderPinChangeB() {
```

```
encoder += digitalRead(encoder_a) != digitalRead(encoder_b) ? -1 : 1;

}
```

APPENDIX C

GUI – Matlab App Designer

```
% Code that executes after component creation
function startupFcn(app)
Target = zeros(2,1);
Cursor = zeros(2,1);
ad.Tpoint = line( app.UIAxes, ...
‘Xdata’,Target(1),...
‘Ydata’,Target(2), ‘Marker’, ‘o’, ...
‘MarkerFaceColor’, ‘r’,...
‘Color’, ‘r’, ...
‘MarkerSize’, 30,...
‘Tag’, ‘TargetPoint’);
ad.Tpoint1 = line( app.UIAxes2, ...
‘Xdata’,Target(1),...
‘Ydata’,Target(2), ‘Marker’, ‘o’, ...
‘MarkerFaceColor’, ‘r’,...
‘Color’, ‘r’, ...
‘MarkerSize’, 30,...
‘Tag’, ‘TargetPoint’);
ad.Cpoint = line(app.UIAxes, ...
‘Xdata’,Cursor(1),...
‘Ydata’,Cursor(2), ‘Marker’, ‘o’, ...
‘MarkerFaceColor’, ‘b’,...
‘Color’, ‘b’, ...
‘MarkerSize’, 15, ...
‘Tag’, ‘CursorPoint’);
ad.Cpoint1 = line(app.UIAxes2, ...
‘Xdata’,Cursor(1),...
‘Ydata’,Cursor(2), ‘Marker’, ‘o’, ...
‘MarkerFaceColor’, ‘b’,...
‘Color’, ‘b’, ...
‘MarkerSize’,15,...
‘Tag’, ‘CursorPoint’);
%Define the files in which the data will be saved
fid = fopen(‘TargetPosition_1.txt’,’wt’);
fidC = fopen(‘CursorPosition_1.txt’,’wt’);
ad.fid = fid;
ad.fidC = fidC;
% Define Path of the Target
a = 0;
b = 20;
n =  a + (b-a)*rand(1,1);
x = @(t) 3*sin(1.8*(t+n)) + 3.4*sin(1.8*(t+n)) + 2.5*sin (1.82*(t+n)) +
   4.3*sin(2.34*(t+n)) ;
y = @(t) 3*sin(1.1*(t+n)) + 3.2*sin(3.6*(t+n)) + 3.8*sin(2.5*(t+n)) +
   4.8*sin(1.48*(t+n)) ;
t1 = 25.5;
ad.x = x;
ad.y = y;
ad.t1 = t1;
```

```
%Define Cursor position variables
ad.xC = 0;
ad.yC = 0;
% Activate Arduino and define variables related to it
[ad.encoder1, ad.encoder2] = Activate_Arduino();
ad.count1 = 0;
ad.count2 = 0;
ad.time = 0;
ad.theta1 = 0;
ad.theta2 = 0;
%Define Visual-Motor rotation parameter
fi = 80;
ad.fi = fi;
%Set Handles so that they are visible to all function
ad.handles = guihandles(app.UIFigure);
guidata(app.UIFigure,ad);
end
% Button pushed function: StartButton
function StartButtonPushed(app, event)

ad = guidata(app.UIFigure);

%Define the flag to start and stop the update of the plot

%If flag is equal to 1 then the plot is updating
global flag;
flag = 1;

% % Turn off the Start button

set(app.StartButton,'Enable','off');
% % Turn on the Stop button
set(app.StopButton,'Enable','on');

tic

while (toc<60 && flag == 1)

    set(ad.Tpoint, 'Xdata', ad.x(toc), 'Ydata', ad.y(toc));
    set(ad.Tpoint1, 'Xdata', -ad.x(toc), 'Ydata', -ad.y(toc));

    [ad.count1, ad.count2, ad.time] =
    Counting(ad.encoder1,ad.encoder2);

    ad.theta1 = ad.count2 * 0.15 - 45;

    ad.theta2 = (((ad.count1 - ad.count2) * 0.15)/2) + 45;
    ad.xC = (2*ad.t1*cosd(ad.theta2)*sind(ad.theta1+ad.theta2));
```

```
7. ad.yC = (-2*ad.t1*cosd(ad.theta2)*cosd(ad.theta1+ad.theta2))
   + 36.062445840513924;

8. fprintf(ad.fid,'%f, %d, %d\n', toc ,ad.x(toc),  ad.y(toc));
   %fid->Target

9. fprintf(ad.fidC,'%f, %d, %d\n', toc, ad.xC, ad.yC);
10.%fidC->Cursor

63.fprintf(ad.fidC,'%f %d %d \n', toc, ad.xC*cosd(ad.fi) +
   ad.yC*sind(ad.fi),...
64.ad.yC*cosd(ad.fi) - ad.xC*sind(ad.fi));

65.set (ad.Cpoint, 'Xdata', ad.xC*cosd(ad.fi) + ad.yC*sind(ad.fi), ...

66.'Ydata', ad.yC*cosd(ad.fi) - ad.xC*sind(ad.fi));
67.set (ad.Cpoint1, 'Xdata', -(ad.xC*cosd(ad.fi) + ad.yC*sind(ad.fi)), ...
68.'Ydata', -(ad.yC*cosd(ad.fi) - ad.xC*sind(ad.fi)));

1. set (ad.Cpoint, 'Xdata', ad.xC, ...
      a. 'Ydata', ad.yC);
2. set (ad.Cpoint1, 'Xdata', -ad.xC, ...
      a. 'Ydata', -ad.yC);

3. drawnow limitrate

  ii. end

 iii. if (flag == 1)

1. fclose(ad.fid);
2. fclose(ad.fidC);
  iv. end
   v. end
  vi. % Button pushed function: StopButton
 vii. function StopButtonPushed(app, event)

viii. %If flag is equal to 0 then the plot stops updating

  ix. global flag;
   x. flag = 0;

  xi. fclose(ad.fid);

 xii. fclose(ad.fidC);

xiii. % % Turn on the Start button

 xiv. set(app.StartButton,'Enable','on');
  xv. % % Turn off the Stop button
 xvi. set(app.StopButton,'Enable','off');
```

```
end

% Close request function: UIFigure
function UIFigureCloseRequest(app, event)
delete(app)
end
end
```

APPENDIX D

Springer Nature – Material Permission

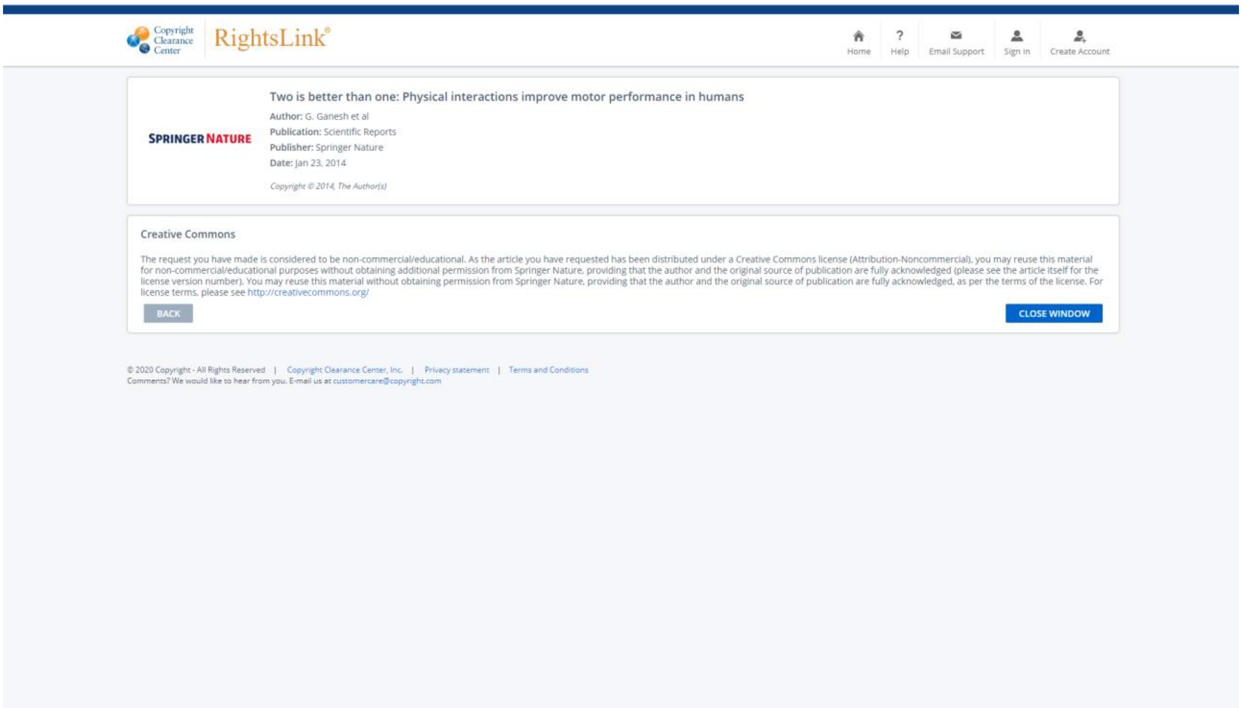

Figure 15 Permission grant to use Springer Nature Material

CITED LITERATURE

[1] J. L. Zimbelman, S. P. Juraschek, X. Zhang, and V. W. H. Lin, "Physical Therapy Workforce in the United States: Forecasting Nationwide Shortages," *PM R*, vol. 2, no. 11, pp. 1021–1029, 2010, doi: 10.1016/j.pmrj.2010.06.015.

[2] G. Ganesh, A. Takagi, R. Osu, T. Yoshioka, M. Kawato, and E. Burdet, "Two is better than one: Physical interactions improve motor performance in humans," *Sci. Rep.*, vol. 4, pp. 1–7, 2014, doi: 10.1038/srep03824.

[3] K. B. Reed and M. A. Peshkin, "Physical collaboration of human-human and human-robot teams," *IEEE Trans. Haptics*, vol. 1, no. 2, pp. 108–120, 2008, doi: 10.1109/TOH.2008.13.

[4] J. L. Patton, M. E. Stoykov, M. Kovic, and F. A. Mussa-Ivaldi, "Evaluation of robotic training forces that either enhance or reduce error in chronic hemiparetic stroke survivors," *Exp. Brain Res.*, vol. 168, no. 3, pp. 368–383, 2006, doi: 10.1007/s00221-005-0097-8.

[5] C. Duret, A. G. Grosmaire, and H. I. Krebs, "Robot-assisted therapy in upper extremity hemiparesis: Overview of an evidence-based approach," *Front. Neurol.*, vol. 10, no. APR, pp. 1–8, 2019, doi: 10.3389/fneur.2019.00412.

[6] R. Bertani, C. Melegari, M. C. De Cola, A. Bramanti, P. Bramanti, and R. S. Calabrò, "Effects of robot-assisted upper limb rehabilitation in stroke patients: a systematic review with meta-analysis," *Neurol. Sci.*, vol. 38, no. 9, pp. 1561–1569, 2017, doi: 10.1007/s10072-017-2995-5.

[7] C. Zhang, C. W. P. Li-Tsang, and R. K. C. Au, "Robotic approaches for the rehabilitation of upper limb recovery after stroke: A systematic review and meta-analysis," *Int. J. Rehabil. Res.*, vol. 40, no. 1, pp. 19–28, 2017, doi: 10.1097/MRR.0000000000000204.

[8] F. Abdollahi *et al.*, "Error augmentation enhancing arm recovery in individuals with chronic stroke: A randomized crossover design," *Neurorehabil. Neural Repair*, vol. 28, no. 2, pp. 120–128, 2014, doi: 10.1177/1545968313498649.

[9] N. E. H. Ogan, "Quantization of continuous arm movements in humans with brain injury," vol. 96, no. April, pp. 4645–4649, 1999.

[10] J. Hidler *et al.*, “Multicenter randomized clinical trial evaluating the effectiveness of the Lokomat in subacute stroke,” *Neurorehabil. Neural Repair*, vol. 23, no. 1, pp. 5–13, 2009, doi: 10.1177/1545968308326632.

[11] M. D. Ziegler, H. Zhong, R. R. Roy, and V. R. Edgerton, “Why variability facilitates spinal learning,” *J. Neurosci.*, vol. 30, no. 32, pp. 10720–10726, 2010, doi: 10.1523/JNEUROSCI.1938-10.2010.

[12] N. Jarrassé, T. Charalambous, and E. Burdet, “A Framework to Describe, Analyze and Generate Interactive Motor Behaviors,” *PLoS One*, vol. 7, no. 11, 2012, doi: 10.1371/journal.pone.0049945.

[13] A. Sawers and L. H. Ting, “Perspectives on human-human sensorimotor interactions for the design of rehabilitation robots,” *J. Neuroeng. Rehabil.*, vol. 11, no. 1, pp. 1–13, 2014, doi: 10.1186/1743-0003-11-142.

[14] K. B. Reed, M. Peshkin, M. J. Hartmann, J. Patton, P. M. Vishton, and M. Grabowecky, “Haptic cooperation between people, and between people and machines,” *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 2109–2114, 2006, doi: 10.1109/IROS.2006.282489.

VITA

NAME	Adriano Gendy
EDUCATION	
	Master of Science in “ Bioengineering ”, University of Illinois at Chicago, May 2021, USA
	Bachelor’s Degree in Bioengineering, Jul 2018, Politecnico di Milano, Italy
LANGUAGE SKILLS	
Italian	Native speaker
English	Full working proficiency
	A.Y. 2019/20 One Year of study abroad in Chicago, Illinois
	A.Y. 2018/19. Lessons and exams attended exclusively in English
Arabic	Basic communication skills