Learning and Robotic Imitation of Therapist's Behavior for Rehabilitation Therapy

by

Carlos Manuel Martínez

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

CONTROL SYSTEMS

Department of Electrical and Computer Engineering
University of Alberta

ABSTRACT

The demand for rehabilitation services has increased in recent years due to population aging. Due to the limitations of therapists' time and healthcare resources, robot-assisted therapy is becoming an appealing, powerful and economical solution. In this thesis, and in order to reach a long term goal, we propose different solutions that combine Learning from Demonstration (LfD) algorithms and robotic rehabilitation to save the therapist's time and reduce the therapy costs as well as the patient's recovery time. The target of this work is to show how medical robotics can be used in combination with LfD algorithms to learn and reproduce the therapist's behavior during therapy based on ADL's. In this thesis, three different tasks and experiments are presented. First, a telerehabilitation system to perform a unimanual cooperative task using LfD algorithms is presented. The second experiment targets ADLs that involve periodic motion in a 2D space. Later, a 2D reaching motion control task, as well as a 2D force control task are presented using a different LfD algorithm that helps to ensure the global asymptotic stability of the system. This thesis presents a step forward in the robotics rehabilitation context. By using LfD algorithms, we show that there is a new paradigm in the rehabilitation field where the robots can learn the therapist's behavior and reproduce it even for complex tasks.

DEDICATION

To my parents, Eloisa and Rubén:
Thank you for your support, help, love, and for allowing me to follow my dreams.

To my brother, Ulises:
My best friend in this journey called life.

To my grandparents Guadalupe, Fortinita, and Carlos:
Two of you left us while I was following my dreams and I could not say goodbye to you. Thank you for all the memories together. We will be together again someday.

To Lili:
My companion of a thousand battles that has been by my side all this time supporting me and giving me reasons to keep fighting.

To God:
Thank you for allowing me a new achievement in my life.

A mis padres, Eloisa y Rubén:
Gracias por su apoyo, ayuda, amor, y por permitirme seguir mis sueños.

A mi hermano, Ulises:
Mi mejor amigo en este viaje llamado vida.

A mis abuelos Guadalupe, Fortinita y Carlos:
Dos de ustedes partieron mientras yo estaba siguiendo mis sueños y no tuve la oportunidad de despedirme. Gracias por todos los recuerdos juntos. Estaremos juntos otra vez algún día.

A Lili:
Mi compañera de mil batallas que ha estado a mi lado por todo este tiempo apoyándome y dándome motivos para seguir luchando.

A Dios:
Gracias por permitirme un nuevo logro en mi vida.

ACKNOWLEDGEMENTS

An achievement such as a graduate degree requires hard work, an unmeasurable effort, nights without sleeping, and last but not least, an entire group of people to help, support, cheer and believe in the applicant along the way. In this thesis, I would like to show you all the memories, love, and support that I accumulate during my graduate studies.

With that being said, I would like to first and foremost thank my advisor, Dr. Mahdi Tavakoli, who gave me the opportunity to achieve a new goal in my life. You inspired me to give my best every day of my life and reminded me that everything is possible. Thank you for believing in me and for your invaluable guidance and patience throughout this research project.

Special thanks to Dr. Mahdi Tavakoli, Dr. Kim Adams, and Dr. Hossein Rouhani for serving on the examination committee.

My family has been my support, strength and main source of inspiration throughout not only my graduate studies but my life as a whole. Even though it has been eight years since I had to move out to keep following my dreams, you were always by my side. Thank you for believing in me, there is no way to pay you back for all your love, dedication, passion and generosity that you have invested in my life.

Thank you to my friends and labmates for helping me in my personal and professional life. This achievement would not be possible without you. You made this period in my life an amazing experience that I will never forget.

Funding for this research has been provided in part by the Mexico's National Commission of Science and Technology (CONACYT); the Canada Foundation for Innovation (CFI); the Alberta Innovation and Advanced Education Ministry under Small Equipment Grant; the Natural Sciences and Engineering Research Council of Canada (NSERC) under the

Collaborative Health Research Projects (CHRP), the Quanser Inc.; and the University of Alberta.

TABLE OF CONTENTS

1 INTRODUCTION 1

1.1 Motivation 1

1.2 Organization of Thesis 2

1.3 Publications 3

1.4 Contribution of the Thesis 3

1.4.1 Proposal of new robotic rehabilitation systems 4

1.4.2 LfD applied to ADLs that involve periodic motions 4

1.4.3 LfD applied to force tasks 4

1.4.4 Use of SEDS to follow a complex path by segmenting it into small trajectories 4

2 BACKGROUND 5

2.1 Therapeutic Robotics 5

2.1.1 Effect of robotic therapy on motor function recovery 6

2.2 Telerehabilitation 7

2.3 Learning from Demonstration (LfD) 8

2.4 Smart rehab robots 8

3 LEARNING AND IMITATION OF A COOPERATIVE THERAPY EXERCISE 10

3.1 Introduction 10

3.1.1 Prior Art 13

3.2 Learning from Demonstration 14

3.2.1 Gaussian Mixture Model 15
3.2.2 Gaussian Mixture Regression 16
3.3 Experiments Materials and Methods 17
3.3.1 Materials 17
3.3.2 Methods 17
3.3.2.1 Task 17
3.3.2.2 GMM and GMR Design 18
3.3.2.3 Experiments 19
3.4 Results and Discussion 21
3.4.1 GMR Output for Different Scenarios 21
3.4.2 Evaluation of Training Data Quality 27
3.5 Conclusion and summary 30
4 LEARNING AND IMITATION OF SEMI-PERIODIC MOTIONS 31
4.1 Introduction 31
4.2 Related Work 34
4.3 Task Description 35
4.4 Proposed Framework 37
4.4.1 First Scenario 38
4.4.1.1 Demonstration Phase 38
4.4.1.2 Reproduction Phase 39
4.4.2 Second Scenario 41
4.4.2.1 Demonstration Phase 41
4.4.2.2 Reproduction Phase 41

4.4.3 Third Scenario 42
4.4.3.1 Demonstration Phase 42
4.4.3.2 Reproduction Phase 45
4.5 Experiments 46
4.6 Results 49
4.6.1 First Scenario 50
4.6.2 Second Scenario 53
4.6.3 Third Scenario 58
4.7 Conclusion and summary 64
5 LEARNING AND ROBOTIC IMITATION OF THERAPIST'S MOTION AND FORCE 66
5.1 Introduction 66
5.2 Related Work 68
5.3 Task Description 69
5.3.1 Task 1 70
5.3.2 Task 2 71
5.4 Learning From Demonstration Algorithms 71
5.5 Experiments 76
5.5.1 Task 1 76
5.5.2 Task 2 77
5.6 Results 78
5.6.1 Task 1 78
5.6.1.1 Sub-Task 78
5.6.1.2 Complex task 83

5.6.2 Task 2 84
5.7 Conclusion and summary 88
6 CONCLUSIONS AND FUTURE WORK 90
6.1 Conclusions 90
6.2 Future Work 91
6.2.1 Implement more complex ADLs with sophisticated rehabilitation robots 91
6.2.2 SEDS applied to periodic motions 91
6.2.3 LfD algorithms applied to patient's and therapist's assessment 92
6.2.4 Reinforcement learning applied to the current systems 92
6.2.5 Perform experiments with people with disabilities 92
REFERENCES 94
APPENDIX A: REHAB ROBOT INVERSE AND FORWARD KINEMATICS 103
APPENDIX B: MATLAB CODE 107
B.1 Code for Chapters 3 and 4 107
B.2 Code for Chapter 6 110
APPENDIX C: DATA ANALYSIS 114
C.1 Correlation Coefficient 114
C.2 Mean Square Error 114
C.3 Euclidean Distance 115

LIST OF TABLES

Table 3-1 Error between GMR output and recorded therapist position, averaged over the duration of the demonstration. Three trials are provided for each therapist behavior, which represents the removal of one of the three training demonstrations. 29

Table 4-1 Numerical analysis of 1D periodic motion 52

Table 4-2 Numerical analysis of 2D periodic motion 58

Table 4-3 Numerical analysis of 2D periodic motion with GMM and GMR algorithms 64

Table 5-1 Numerical analysis of position control task using SEDS 82

Table 5-2 Numerical analysis of force control task using SEDS 88

LIST OF FIGURES

Figure 3-1 (Tao, 2014) Illustrations for the TIL phase (a) where the patient interacts with the therapist, and TOOL phase (b) where the patient interacts with a slave robot that emulates the therapist's behavior. 13

Figure 3-2 (a) Experiment setup and demonstration; (b) HD2 High Definition Haptic Device (Quanser Inc., Markham, Ontario, Canada) used as the master robot by the therapist; (c) a Motoman SIA-5F (Yaskawa America, Inc., Miamisburg, Ohio, USA) industrial robot. 14

Figure 3-3 Design of the cooperative task. The slave robot holds one side of the bar, while the patient holds the bar from its other side. The bar's inclination can be altered in a 180° range. 17

Figure 3-4 The two phases of LfD are shown in diagrams 1-2 and 3-4, respectively. In diagram 1, the therapist is present (making this the TIL phase). The patient will initiate movement as they lift the bar, to which the therapist will respond as shown in diagram 2. The data from both robots will be recorded and used to generate the GMM. Then in diagram 3, the patient is practicing in the absence of the therapist (the TOOL phase). The robot utilizes GMR to emulate the therapist and respond to the patient's movements, as shown in diagram 4. 19

Figure 3-5 This figure summarizes the system's components needed to execute the task. The top figure shows a block diagram for the demonstration phase while the bottom figure shows a block diagram for the reproduction phase. 20

Figure 3-6 Obtained results for the slow scenario during the reproduction phase, the inputs are the patient's position (red solid line), patient's velocity (black dotted line), and the output is the therapist's expected position (blue dashed line). 22

Figure 3-7 Obtained results for the medium scenario during the reproduction phase, the inputs are the patient's position (red solid line), patient's velocity (black dotted line), and the output is the therapist's expected position (blue dashed line). 23

Figure 3-8 Obtained results for the fast scenario during the reproduction phase, the inputs are the patient's position (red solid line), patient's velocity (black dotted line), and the output is the therapist's expected position (blue dashed line). 23

Figure 3-9 Obtained results for the back scenario during the reproduction phase, the inputs are the patient's position (red solid line), patient's velocity (black dotted line), and the output is the therapist's expected position (blue dashed line). 24

Figure 3-10 Obtained results for the simulated scenario during the reproduction phase, the inputs are the patient's position (red solid line), patient's velocity (black dotted line), and the output is the therapist's expected position (blue dashed line). 24

Figure 3-11 Obtained results for the multi-behavioral scenario during the reproduction phase, the inputs are the patient's position (red solid line), patient's velocity (black dotted line), and the output is the therapist's expected position (blue dashed line). 25

Figure 3-12 Obtained results for the plus 45 scenario during the reproduction phase, the inputs are the patient's position (red solid line), patient's velocity (black dotted line), and the output is the therapist's expected position (blue dashed line). 25

Figure 3-13 Obtained results for the negative 45 scenario during the reproduction phase, the inputs are the patient's position (red solid line), patient's velocity (black dotted line), and the output is the therapist's expected position (blue dashed line). 26

Figure 3-14 Obtained results for the zero scenario during the reproduction phase, the inputs are the patient's position (red solid line), patient's velocity (black dotted line), and the output is the therapist's expected position (blue dashed line). 26

Figure 3-15 Analysis of the dataset quality for sample trials of back scenario in phase two. 27

Figure 3-16 Analysis of the dataset quality for sample trials of medium scenario in phase two. 28

Figure 3-17 Analysis of the dataset quality for sample trials of medium scenario in phase two. 28

Figure 3-18 Analysis of the dataset quality for sample trials of slow scenario in phase two. 28

Figure 4-1 The left figure shows a diagram for task 1, while right figure shows a diagram for task 2. In both tasks, the patient has to follow a desired trajectory while moving the robot's end effector. 36

Figure 4-2 First, during the demonstration phase, the therapist performs the task for a short time. Then, using robot learning from demonstration, the task is modeled as an average trajectory and variations in trajectory. The controller assists the patient by two varying impedance models (spring-damper) to follow the demonstrated trajectory and remain in the demonstrated range of variability. 37

Figure 4-3 Robot's end-effector representation in both fix (black) frame (X,Y) and dynamic (red) frame (X,Y) 41

Figure 4-4 Block diagram of the system used for Scenarios 1 and 2. 42

Figure 4-5 Block Diagram of EM algorithm 44

Figure 4-6 Block diagrams of the system used for task three. The top figure shows a block diagram to describe the learning phase. The bottom figure shows a block diagram that describes the reproduction phase. 47

Figure 4-7 Parallel training mode. During the training phase (stage 1 & 2) the therapist and the patient interact with the robot to teach it about the task performance. Later, during the reproduction phase (stage 3 & 4), the therapist is no longer involved in the therapy. The robot helps the patient to perform the task as taught by the therapist. 48

Figure 4-8 Sequential training mode. During the training phase (stage 1 & 2) the therapist teaches the robot about the task performance. Later, during the reproduction phase (stage

3 & 4), the therapist is no longer involved in the therapy. The robot helps the patient to complete the task as taught by the therapist. 48

Figure 4-9 Figure showing the demonstration phase. The therapist interacts with the robot to train the system. 49

Figure 4-10 Figure showing the reproduction phase. The patient interacts with the robot to execute the given task, in the therapist's absence, the robot assists the patient to complete and perform the task just as the therapist did. 49

Figure 4-11 Therapist's demonstration and patient's reproduction with and without robotic assistance in X-axis. 50

Figure 4-12 Cross-correlation between the patient's reproduction with and without robotic assistance and the therapist's demonstration in X-axis. 51

Figure 4-13 Power Spectrum analysis of patient's reproduction with and without robotic assistance and therapist's demonstration. 52

Figure 4-14 Therapist's demonstration and patient's reproduction with and without robotic assistance in X-axis 54

Figure 4-15 Therapist's demonstration and patient's reproduction with and without robotic assistance in Y-axis 54

Figure 4-16 Therapist's demonstration and patient's reproduction with and without robotic assistance in 2D. 55

Figure 4-17 Cross-correlation between the patient's reproduction with and without robotic assistance and the therapist's demonstration in X-axis. 55

Figure 4-18 Cross-correlation between the patient's reproduction with and without robotic assistance and the therapist's demonstration in Y-axis. 56

Figure 4-19 Power Spectrum analysis of patient's reproduction with and without robotic assistance and therapist's demonstration in X-axis. 57

Figure 4-20 Power Spectrum analysis of patient's reproduction with and without robotic assistance and therapist's demonstration in Y-axis. 57

Figure 4-21 Therapist's demonstration and patient's reproduction with and without robotic assistance in X-axis. 60

Figure 4-22 Therapist's demonstration and patient's reproduction with and without robotic assistance in Y-axis. 60

Figure 4-23 Therapist's demonstration and patient's reproduction with and without robotic assistance in 2D. 61

Figure 4-24 Cross-correlation between the patient's reproduction with and without robotic assistance and the therapist's demonstration in X-axis. 61

Figure 4-25 Cross-correlation between the patient's reproduction with and without robotic assistance and the therapist's demonstration in Y-axis. 62

Figure 4-26 Power Spectrum analysis of patient's reproduction with and without robotic assistance and therapist's demonstration in X-axis. 62

Figure 4-27 Power Spectrum analysis of patient's reproduction with and without robotic assistance and therapist's demonstration in Y-axis. 63

Figure 5-1 Task 1 graphical description. 70

Figure 5-2 Task 2 virtual springs along X-axis and Y-axis. 72

Figure 5-3 Task 2 graphical description. 72

Figure 5-4 Block diagram showing the learning and reproduction phase. 76

Figure 5-5 Rehab Robot and setup used in these experiments. 77

Figure 5-6 Task 2 setup with the springs' array. 78

Figure 5-7 Sub-task reproductions. 79

Figure 5-8 Cross-correlation plots of patient's reproduction with assistance (Top) and patient's reproduction without assistance (Bottom) along the X-axis. 80

Figure 5-9 Cross-correlation plots of patient's reproduction with assistance (Top) and patient's reproduction without assistance (Bottom) along the Y-axis. 80

Figure 5-10 Power Spectrum analysis of patient's reproduction with and without robotic assistance and therapist's demonstration in X-axis. 81

Figure 5-11 Power Spectrum analysis of patient's reproduction with and without robotic assistance and therapist's demonstration in Y-axis. 81

Figure 5-12 Complex task patient's reproduction. 83

Figure 5-13 Measured forces along X-axis. 85

Figure 5-14 Measured forces along Y-axis. 85

Figure 5-15 Cross-correlation plots of patient's reproduction with assistance (Top) and patient's reproduction without assistance (Bottom) along the X-axis. 86

Figure 5-16 Cross-correlation plots of patient's reproduction with assistance (Top) and patient's reproduction without assistance (Bottom) along the Y-axis. 87

Figure 5-17 Power Spectrum analysis of patient's reproduction with and without robotic assistance and therapist's demonstration in X-axis. 87

Figure 5-18 Power Spectrum analysis of patient's reproduction with and without robotic assistance and therapist's demonstration in Y-axis. 88

Figure A0-1 Rehab robot components. 104

Figure A0-2 Rehab robot links and joint angles. 104

Figure A0-3 Model used to solve inverse kinematics. 106

LIST OF SYMBOLS AND ABBREVIATIONS

$\hat{\Sigma}_{\Delta}$	Conditional covariance
$\dot{X}_{Pa}$	Patient velocity in the vertical axis
$\hat{\xi}_{\Delta}$	Conditional expectation
C_1	Selected threshold for convergence
X_{Pa}	Patient position in the vertical axis
X_{Th}	Therapist position in the vertical axis
a_0	Constant term of Fourier Series Coefficient
a_n	Vector of cosine amplitudes of Fourier Series Coefficient
b_n	Vector of sine amplitudes of Fourier Series Coefficient
$\hat{\xi}$	Estimated D-dimensional continuous-value data vector
$\prec 0$	Negative definiteness of a matrix
Δ	Vector of Fourier Series Coefficients
1D	1 Dimension
2D	2 Dimensions

3D	3 Dimensions
ADL	Activities of Daily Living
ANN	Assist-as-Needed
CC	Cross-Correlation
DOF	Degrees of Freedom
E	Error
ED	Euclidean Distance
EM	Expectation Maximization
F	Force
FFT	Fast Fourier Transform
FS	Fourier Series
GAS	Global Asymptotic Stability
GMM	Gaussian Mixture Model
GMR	Gaussian Mixture Regression
$\mathcal{L}$	Log-likelihood
LAR	Learn and replay
LfD	Learning from Demonstration

m	Slope between two consecutive points
MIT	Massachusetts Institute of Technology
MSE	Mean Square Error
SEDS	Stable Estimator of Dynamical Systems
T	Period of the signal
t	Time
TENS	Transcutaneous electric nerve stimulator
TIL	Therapist-In-Loop
TOOL	Therapist-Out-Of-Loop
w	Fundamental frequency
θ	Angle in degrees
Σ	Covariance matrix
Ω	Vector with means, priors, and covariance matrices of the Gaussian components
J	Jacobian Matrix
μ	Mean
ξ	D-dimensional continuous-valued data vector

$\boldsymbol{\pi}$	Priors

1 INTRODUCTION

1.1 Motivation

In recent years, aging population is causing a growing demand for movement rehabilitation therapy following a stroke, joint injury, and many other accidents causing motor impairment of human limbs (Adamson, 2004). The most severe problem emerges due to the strokes; strokes have become one of the most common causes of disability not only in Canada but also worldwide (Federation, 2017), (Foundation, 2017). Currently, in Canada, there are more than 62,000 stroke cases each year, and 405,000 Canadians are living with a long-term stroke disability (Bogart, 2017). This problem impacts the Canadian economy by as much as $21B a year (Canadians, 2017).

Patients living with a long-term disability require personalized therapy to regain strength and mobility. However, due to the increasing demand for therapists' time and healthcare resources, therapy sessions are not offered as frequently as expected. This demand has motivated the development of new techniques and technologies for rehabilitation therapy. One of the most significant solutions rises from the robotic field. The inclusion of robots into the rehabilitation field has become a powerful solution to target the existing problems. Robots have brought therapeutic benefits to physiotherapy programs (Krebs & Hogan, 2006). Their ability to execute precise and repetitive motions without fatigue as well as the capability to collect precise information such as position, force, performance turn robots into the perfect assistant for therapists.

Even though robotic therapy has helped to improve traditional therapy, the current robotics systems are designed to execute fixed and known tasks. It is clear that the current technology does not take advantage of all of the robots' capabilities such as adaptability to different tasks and patients.

Robot programming by demonstration is a new tool in the rehabilitation robotics field that allows robot reprogramming to execute any given task without any previous robot programming knowledge. This tool is emerging as a popular solution in the robotics field; its inclusion in medical robotics is a new paradigm that we are pursuing in this thesis and that should be explored to bring benefits to people with disabilities.

1.2 Organization of Thesis

Chapter 2 presents previously developed work in the robotic rehabilitation field including using Learning from Demonstration (LfD) techniques for rehabilitation. Basic concepts and ideas behind LfD are presented, followed by a brief description of related work. This chapter helps the readers to understand the context in which the research is carried out.

Chapter 3 proposes a system where a therapist and a patient interact in a cooperative task through a telerobotic medium. In this chapter, we demonstrate how LfD and teleoperation can be combined to save patient's travel time as well as therapist's time. The proposed task targets a common problem in post-stroke patients, which is the compensation of the affected limb using the non-affected limb. The patient is instructed to only use the affected limb to cooperate with the therapist. This chapter was developed together with the MSc student Jason Fong. We had equal contributions to developing the systems, programming the algorithms and writing the paper that resulted in this chapter.

Chapter 4 proposes a system that uses an LfD algorithm to learn and reproduce a semi-periodic motion trajectory using Fourier Series (FS). The advantage of this system is the fact that it uses a simple and well-developed algorithm such as FS and learns the motion based on the frequencies and amplitudes.

Until now, the previously developed systems in the rehabilitation field using robotic devices and LfD are not interested on the stability of the system. These systems have shown an excellent performance in the given tasks, but there was no discussion of stability. Ensuring stability could help to create safer devices for therapists and patients.

Chapter 5 uses a previously developed LfD robotic rehabilitation system paradigm called Stable Estimator of Dynamical Systems (SEDS) and combines it with a robotic rehabilitation system to perform force and position control tasks in 2 degrees-of-freedom (DOFs). In the proposed tasks, the system converges to the origin; in the force task, the origin is considered a force equal to 0 Newton while in the position task, the origin is where the X-axis and Y-axis are equal to zero. We also test a 2-DOF force and a 2-DOF position control task under the SEDS paradigm. In this chapter, we show the potential of SEDS in the robotic rehabilitation field. This chapter shows SEDS is probably one of the best algorithms for the robotic rehabilitation field.

Finally, Chapter 6 summarizes the research findings and presents suggested future directions of research.

In this thesis, therapists and patients were *simulated*. Along this thesis, the word *therapist* refers to a person taking the role of the therapist. Notice that this person does not have therapeutic skills. The word *patient* refers to a non-disabled person that took the role of a person with disabilities. Different devices such as springs and transcutaneous electric nerve stimulator (TENS) were used to simulate the motor impairments.

1.3 Publications

A condensed version of Chapter 3 was presented in the 2018 International Symposium on Medical Robotics, Atlanta, USA, March 2018. A short version of Chapter 5 was presented in the 2017 IEEE International Conference on Systems, Man, And Cybernetics, Banff, Canada, October 2017.

1.4 Contribution of the Thesis

This thesis makes several new contributions in the areas of medical robotics for post-stroke rehabilitation using LfD, LfD applied to semi-periodic motion and cooperative tasks, and LfD applied to force and position control tasks with dynamic origins in 1-DoF and 2-DoF.

1.4.1 Proposal of new robotic rehabilitation systems

The combination of LfD algorithms and medical robotics for post-stroke rehabilitation therapy proposed in this thesis is a novel rehabilitation paradigm aiming at providing a solution to the increasing demand for therapist's time and healthcare resources. LfD allows for time sharing a therapist between multiple patients. Even though these algorithms have been previously used in other fields, this thesis explores methodologies, tasks, and algorithms for using LfD in rehabilitation therapy.

1.4.2 LfD applied to ADLs that involve periodic motions

The learned semi-periodic motion trajectory using LfD algorithms presented in Chapter 4 is a new paradigm in the robotics rehabilitation field. While most of the previously developed robot learning systems are used for non-periodic motions, the proposed system in this thesis targets periodic motions. In this way, we show a novel system capable of helping therapists during therapy to develop more complex tasks.

1.4.3 LfD applied to force tasks

So far, most of the previously developed robotic rehabilitation systems have been implemented to learn position trajectory-based tasks. Learning force based tasks using an LfD algorithm introduces new possibilities for choosing therapy tasks.

1.4.4 Use of SEDS to follow a complex path by segmenting it into small trajectories.

Even though the implementation of SEDS is not new in the robotic field, the novelty and contribution emerge in Chapter 5, where the SEDS algorithm is no longer constrained to a fixed origin. This new paradigm can be used in trajectory-based tasks to follow a given path. In force-based tasks, it can be used to implement tasks with varying forces.

2 BACKGROUND

This chapter presents the needed background to understand the basis and principles present in this thesis. Section 2.1 provides an overview of therapeutic robotics including a brief history. Section 2.2 discusses the telerehabilitation robotics field. The effect of robotic therapy on motor function recovery is reviewed in Section 2.3. In Section 2.4, a brief explanation of learning from demonstration and machine learning is provided. Finally, in Section 2.5, an explanation of smart rehab robots, as well as the previously developed work, is presented.

2.1 Therapeutic Robotics

A robot is a re-programmable multifunctional manipulator designed to accomplish a programmed task (Skillicorn, 2018). In the rehabilitation field, robots are often thought of as aids to assist people with disabilities (Krebs, 2003). Due to the ability to deliver high-intensity and high-dosage training during therapy, robotics systems are becoming a popular tool in rehabilitation to help patients with motor disorders. Therapeutic robots allow patients to train independently of a therapist and to improve their functional level. These robots enhance traditional therapy techniques by enabling more precise and consistent therapy (Krebs, 2006), especially in therapies that involve highly repetitive movement training. By using these technologies, the systems can collect data that can be used to measure the patient's progress through the therapy, enabling therapists to optimize treatment techniques.

A new field that will be covered and discussed later in this section covers the combination of machine learning and therapeutic robots; the resulting systems have the potential to provide extended periods of unsupervised therapy, which could increase efficiency and reduce costs by decreasing the amount of time that a therapist must spend with a patient.

Even though robots have been present in the recent history almost since the beginning of our modern civilization (Hagis, 2003), they were not incorporated into the rehabilitation field until the last century, more specifically in 1989 (Krebs, 2004).

The first robots used in rehabilitation were known as assistive robots (Krebs, 2003), which are robots used to assist people with disabilities to perform activities of daily living (ADLs). These robots were thought as a tool to make their life easier, but not as a tool to help them to regain motor function or to recover mobility. Most of these robots were industrial robots adapted to patients' needs.

Before 1989, there were no available robotic technologies specifically built for rehabilitation. The pioneer of this new class was the MIT-MANUS, a 2-DOF robot developed at the Massachusetts Institute of Technology (MIT) for planar shoulder-and-elbow therapy (Hogan, 1992).

After the MIT-MANUS established the rules for a completely new paradigm in robotics, new devices were developed to fulfill the therapeutic needs. The most common or popular are the MIME (Lum, 2004) and the ARM Guide (Reinkensmeyer, 2000). In the beginning, most of the rehabilitation robotics systems were developed for the upper limb. In early 2000's, new devices were developed for therapy involving other parts of the human body, such as gait therapy (Colombo, 2000), wrist movement therapy (Williams, 2001), ankle training (Deutsch, 2001), and hand therapy (Worsnopp, 2007).

Rehabilitation robots have changed the way humans perform therapy. Their advantages complement the human skill. New technologies, ideas, and materials have to be created to keep improving our lives.

2.1.1 Effect of robotic therapy on motor function recovery

Even though traditional therapy has been used with reliable results, in the last decades and based on the new technologies, it has been shown that rehabilitation robotics helps to improve the motor outcome and disability of chronic post-stroke patients (Colombo, 2005).

Some studies performed with an MIT-MANUS were conducted on patients with a post-stroke disability. The obtained results showed that patients had reduced shoulder and elbow motor impairment compared to the patients that did not use the robotic device (Volpe, 1999). A different study using a MIME system showed that the patients that worked with the robot obtained better results regarding the Fugl-Meyer score, gains in muscle strength, and reach extent compared to conventional rehabilitation therapy (Lum, 2002). Although these studies show promising results, it is not clear that rehabilitation robots can be comparable to or even more effective than traditional therapy.

One promising technology to improve the current rehabilitation systems is based on LfD. This technology uses machine learning algorithm where the therapist's behavior is learned to later have the robot interact with the patient during the therapist's absence. These new technologies can be used to increase the productivity of therapists and consequently help in reducing healthcare costs as well as increasing therapy availability.

2.2 Telerehabilitation

Telerehabilitation is a subcomponent of the broader area of telemedicine, and it is used for the provision of distance support, assessment and intervention to patients with disabilities using a telecommunication channel.

Telerehabilitation is considered a new technology in the rehabilitation field. It was introduced in 1997 by the National Institute of Disability and Rehabilitation Research (U.S. Department of Education) (Movahedazarhouligh, 2015). Originally it was presented with four main properties:

- Delivery of training, education, and counseling rehabilitation services at a distance.
- Assessment and monitoring of progress and results of rehabilitation at a distance.
- Performing therapeutic interventions at a distance.
- Implementing virtual reality technologies for rehabilitation.

Current telerehabilitation technologies can be classified as unilateral and bilateral, where the former involves just one robot and the later uses two robots. In the unilateral mode, the patient interacts with a rehabilitation robot while the therapist provide instructions to the robot or the patient. Some examples of this system can be found in JavaTherapy (Reinkensmeyer, 2000) and Rutgers Master II (Popescu, 2000) systems. In the bilateral mode, the therapist and the patient interact with each other through a master-slave robotic system. Some examples of this system can be found in (Tao, 2014).

Given that the telerehabilitation allows interaction between the patient and the therapist while both are in different locations, telerehabilitation is mainly used in therapies to save therapist's time as well as patient's time. This technology can also be used to improve patient's recovery time and to reduce the therapy costs. Finally, an emerging technology where the robots are trained to interact with the patients while the therapists are not available using machine learning algorithms is becoming a powerful tool in this field; details about this new technology will be introduced in this thesis.

2.3 Learning from Demonstration (LfD)

Learning from demonstration, also known as "programming by demonstration," "imitation learning," and "teaching by showing" is a concept that started in the robotics field approximately 40 years ago (Atkeson, 1997). Given some data and information about the nature of a robotic system, machine learning algorithms extract the structure of the data by minimizing an error criterion. The goal is to learn policy − the right action in response to an observed event − to guide a dynamical system to perform a task.

The main purpose of this technology is to save the time to program or reprogram a robot by an automatic programming process. A human expert is needed to train or teach the robot how to execute a given task based on some demonstrations. Then, the system learns and reproduces the task automatically and without any manual coding/programming or knowledge about robotics or math.

2.4 Smart rehab robots

As mentioned before, LfD is has the potential to become a popular tool for rehabilitation robots (Tavakoli, 2018). This is especially due to its robustness, friendliness, and intuitive nature. Most of the times, therapists do not have programming or robotics knowledge. The robots used for rehabilitation task are usually programmed to execute predefined tasks. As covered in previous sections, one of the most important advantages of robots is the fact that they can be reprogrammed and they can adapt their behavior to different tasks. Since generally speaking therapists cannot reprogram rehab robots, LfD has the potential to become a powerful tool in the robotic rehabilitation field. This tool allows the therapist to reprogram the robots without the skills of an expert roboticist. Therefore, more personalized therapies, as well as different and more complex tasks, can be implemented by clinicians. Also, due to the nature of LfD, the rehabilitation robots can perform exercises with far more repetitions than therapists. This can be translated into time-saving for the therapist and economic benefits for the healthcare system. Examples about the systems that have been implemented before are explained in Chapters 3, 4 and 5.

3 LEARNING AND IMITATION OF A COOPERATIVE THERAPY EXERCISE

3.1 Introduction

Stroke is the second leading cause of death globally, where 15 million people suffer a stroke every year, causing 6 million deaths and leaving nearly 5 million survivors with a permanent disability (Federation, 2017). In Canada, there are more than 62,000 stroke cases each year, and 405,000 Canadians are living with long-term stroke disability (Bogart, 2017). Cardiovascular disease has an economic impact on Canada of more than $20.9B a year (Canadians, 2017). It is therefore crucial that rehabilitation for stroke survivors reduces these impacts as well as that it increases their quality of life. Research has found that by actively engaging stroke patients in repetitive exercises, the brain can rewire neurological pathways for motor functions to relearn movements, which is known as neuroplasticity (Lum, 2002).

Over the past decades, due to the increase of the stroke population, there has been a growing demand for rehabilitation services, motivating robotics technologies for assisting recovery following disability. As a result, the use of robots to reproduce repetitive rehabilitation tasks and, for therapeutic purposes, to assist or resist the performance of these tasks by the patient has become popular (Loos, 2008) (Voelker, 2005). Traditionally, haptics-enabled robotic rehabilitation has facilitated two categories of movement therapies: assistive therapy and resistive therapy. Assistive therapy involves the use of a haptic device to assist the patient to complete the task, while in resistive

therapy the device will oppose the patient's actions by applying resistive forces to build muscle strength. The focus has more recently shifted towards functional therapies, in which the tasks reproduced during therapy are meant to directly emulate Activities of Daily Living (ADLs) such as open a door, grab a glass of water, eat with a spoon, etc., translating into more meaningful and efficient recovery for patients (Guidali, 2011).

Often, the behaviors of robots during rehabilitative therapies are pre-programmed, which is highly restrictive in the presence of unstructured task environments and given the variation in patients and therapists' abilities. This is in contrast to the flexibility with which a skilled therapist can adjust the parameters of therapy such as the therapy intensity for every situation and based on years of experience. To directly incorporate the therapist's skills in robotic therapy, the field of telerobotic rehabilitation, where there is one robot for the patient and one robot for the therapist, has received increasing interest (Mcclure, 2012). In this chapter, our focus is on telerehabilitation through a bilateral (haptics-enabled) telerobotic system. Haptic feedback, which provides a human who operates a robot/tele-robot with a sense of touching a virtual/physical environment, allows interactions that are gentle, safe, reliable, and precise and is of high importance in rehabilitation robotics and telerobotics (Atashzar, 2015). The main strength of haptics-assisted telerehabilitation is its ability to simulate the so-called "hand-over-hand" therapy over a distance (Atashzar, 2012). Haptic tele-robots are also the ideal tools for moving the rehabilitation process to the home and remote areas for increased access to and reduced costs of healthcare (Carigman, 2006).

Telerobotic rehabilitation also allows therapist-administered therapies to take the form of cooperative tasks exercised collaboratively by the therapist and the patient. We define cooperative tasks as tasks that require the use of two hands to complete (Sainburg, 2013), such as holding a jar and unscrewing its lid or lifting an object with two hands. Allowing for cooperative tasks to be practiced not only provides more variations of therapy tasks to administer, but also provides therapists an opportunity to monitor and guide patients in situations where they may undesirably compensate for their affected limb with the contralateral limb.

We refer to the situation where the therapist is haptically interacting with the patient remotely as Therapist-In-Loop (TIL). While TIL bilateral telerehabilitation has many advantages over unilateral telerehabilitation, a therapist may not always be available to interact with the patient over the telerehabilitation medium. In fact, since the number of patients afflicted with strokes has increased in recent years (Zariffa, 2012), the number of therapists and the hospital resources may become tightly stretched across patient caseloads in the future. A solution to this problem proposed here is to have the patient-side robot first learn the therapy administered by the therapist during the live telerehabilitation session, and then imitate it. As a result, in the absence of the therapist, the patient can continue to practice the task in cooperation with the patient-side robot. We refer to this situation where the therapist is absent as Therapist-Out-Of-Loop (TOOL). The paradigm to transition from TIL to TOOL will be based on learning from demonstration (LfD) techniques (Schaal, 2003). Figure 3-1 depicts the TIL and TOOL phases.

In this chapter, we are interested in creating a variable-difficulty cooperative task through the use of haptic telerehabilitation. The task will be cooperatively performed, with the therapist controlling the slave robot through the master robot to intervene in the task and the patient interacting with the task directly. The master and slave robots, as well as the placement of the therapist, the patient, and the task, are shown in Figure 3-2. LfD based on GMM and GMR will then be implemented to learn how the therapist interacts with the patient to complete the task. We hypothesize that the combination of these distinct techniques can provide a suitable middle ground between hand-over-hand and fully autonomous therapy, and allow a rehabilitation robot to imitate a therapist interaction for a demonstrated task accurately.

In this chapter, therapists and patients were *simulated*. Along this thesis, the word *therapist* refers to a person taking the role of the therapist. Notice that this person does not have therapeutic skills. The word *patient* refers to a non-disabled person that took the role of a person with disabilities.

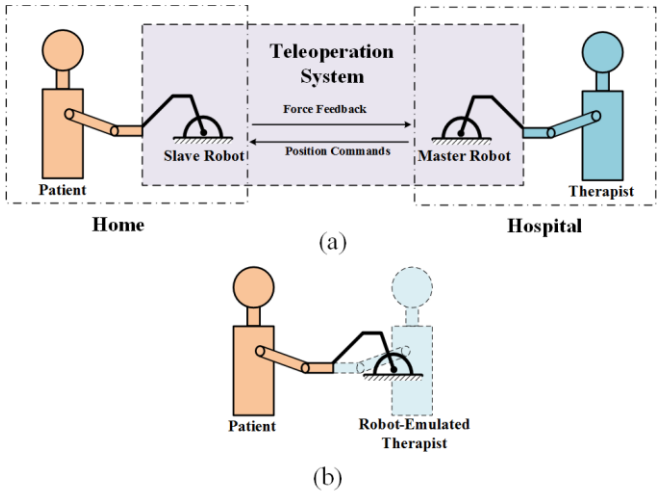

Figure 3-1 (Tao, 2014) Illustrations for the TIL phase (a) where the patient interacts with the therapist, and TOOL phase (b) where the patient interacts with a slave robot that emulates the therapist's behavior.

3.1.1 Prior Art

The concept of LfD has seen extensive research in the past few decades. Application of LfD principles to human-robot interaction has naturally led to the exploration of cooperative tasks. (Evrard, 2009) taught a robot to cooperatively lift a beam in a setup similar to what we propose here. (Gribovskaya, 2010) built upon the same work to ensure global asymptotic stability (GAS) of the system. (Peternel, 2013) created a variant to learn motion and compliance during a highly dynamic cooperative sawing task. However, few groups have applied LfD techniques towards the practice of physical therapy in rehabilitation medicine. (Maaref, 2016) described the use of LfD as the underlying mechanism for an assist-as-needed paradigm. (Lydakis, 2017) learned and classified demonstrations of therapy tasks through EMG measurements. (Lauretti, 2017) optimized a system built on dynamic motor primitives for learning therapist-demonstrated paths for activities of daily living. (Najafi, 2017) learned the trajectory and interaction impedance provided by a simulated therapist and provided user experiment evaluations. These previous works show well-developed innovations in human-robotic interaction, robot-cooperative tasks, and LfD in separate and different contexts. Our work gathers these different ideas into a single system to create a new way to provide human post-stroke

therapy. We propose to combine the best and most important contributions of each of these works to show that robotic therapy can be streamlined for greater practicality.

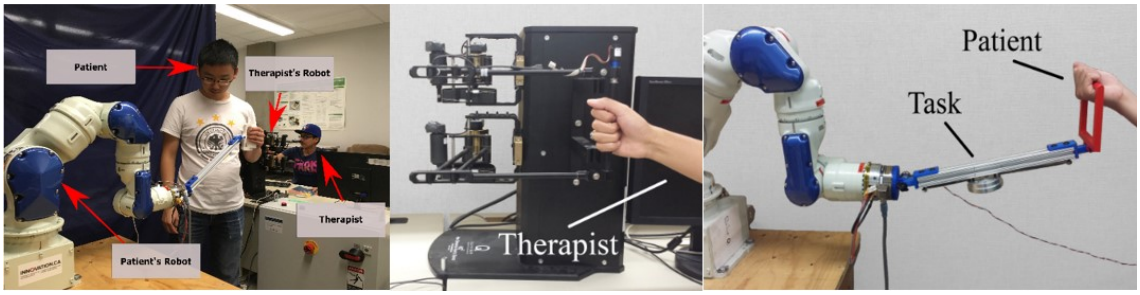

Figure 3-2 (a) Experiment setup and demonstration; (b) HD2 High Definition Haptic Device (Quanser Inc., Markham, Ontario, Canada) used as the master robot by the therapist; (c) a Motoman SIA-5F (Yaskawa America, Inc., Miamisburg, Ohio, USA) industrial robot.

3.2 Learning from Demonstration

LfD is a paradigm focused on allowing a human user to program a robot through demonstration of desired behaviors (Tavakoli, 2018). In other words, a trainer (which can be a human or even another machine) physically demonstrates the behaviors to be imitated by the robot (Devautl, 2015). In general, the behaviors are actions or movements to be later imitated by the robot.

A cornerstone and driver of our LfD-based approach is the assumption that programming know-how is limited in clinical settings. This requires that reprogramming the robotic system between different tasks must be made as simple and user-friendly as possible. State-of-the-art LfD techniques allow for this and facilitate robot learning based on only a few real demonstrations of the task by a human without any additional computer programming overhead.

LfD is divided into two phases, known as the demonstration and imitation phases. In the demonstration phase, a trainer interacts with the robot and performs an action that is to be learned by the robot. Multiple demonstrations of the task can be completed to provide a wider knowledge base for the robot. The imitation phase then reproduces the learned

behavior based on the inputs the robot receives in real time. There are different approaches to learning and imitating a desired behavior (Calinon, 2010).

In this chapter, due to their low memory requirements, an extended use in the literature, and easy implementation, GMM and GMR are used as the underlying learning and imitation algorithms for the LfD paradigm. The GMM algorithm takes multiple demonstrations and extracts the most important values to reconstruct the output. This process avoids redundancy of data in memory. The GMR algorithm uses the stored data and, based on the inputs, retrieves the general form of the output (Reynolds, 2017).

3.2.1 Gaussian Mixture Model

GMM is a probability density function widely used for clustering data (Calinon, 2009). In the robotics field, GMM is extensively used to encode temporal and spatial components of continuous trajectories and behaviors (Calinon, 2007). The model classifies the data by assigning each component to the more similar cluster. To cluster the data, the model relies on a weighted sum of Gaussian component densities, with each Gaussian having its own mean and covariance. Because of these properties and advantages, GMM is widely used for LfD (Calinon, 2007).

GMM is a weighted sum of K component Gaussian densities given by the equation,

$$p(\xi_j) = \sum_{k=1}^{K} p(k)p(\xi_j|k) \tag{3-1}$$

where $p(k)$ = Priors, $p(\xi_j|k)$ = Conditional density function, and ξ_j = D-dimensional continuous-valued data vector.

The parameters in equation (3-1) are defined as

$$p(k) = \pi_k \tag{3-2}$$

$$p(\xi_k|k) = \mathcal{N}(\xi_j; \mu_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^D|\Sigma_k|}} e^{-\frac{1}{2}((\xi_j-\mu_k)^T \Sigma_k^{-1}(\xi_j-\mu_k))} \tag{3-3}$$

Each kth Gaussian component is described by the parameters $\{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$, representing respectively *prior* probabilities, *mean* vectors and *covariance* matrices.

The Expectation-Maximization (EM) algorithm is widely used to train GMM parameters. It takes the GMM parameters and iterates them until convergence of an optimization factor. EM has a simple local search technique that guarantees an increase of the likelihood. The algorithm is well documented, more details are provided in Chapter 4.

3.2.2 Gaussian Mixture Regression

The GMR model uses the Gaussian conditioning theorem and linear combination properties of Gaussian distributions to retrieve the values (Calinon, 2009).

GMR uses temporal values (ξ_t) as query points to estimate the corresponding spatial values ($\hat{\xi}_s$) through regression. Given a set of temporal and spatial values for a kth component of a GMM, the representation of mean and covariance matrix is

$$\mu_k = \{\mu_{t,k}, \mu_{s,k}\}, \qquad \Sigma_k = \begin{pmatrix} \Sigma_{t,k} & \Sigma_{ts,k} \\ \Sigma_{st,k} & \Sigma_{s,k} \end{pmatrix} \tag{3-4}$$

Conditional expectation ($\hat{\xi}_s$) and conditional covariance ($\hat{\Sigma}_s$) of the ξ_s given ξ_t are then calculated for a mixture of all GMM *k* components.

The GMR process is also well documented, so details are also placed in the appendix. Note that while the query points are described as temporal points, these inputs to the GMM and GMR can be any type of data. As is the case in our work, the learned system behaviors can be time-independent, and spatial coordinates, as an example, can be used as the query points.

The GMR model only needs the means and covariance matrices of the GMM to retrieve the signal. This helps to use memory more efficiently; otherwise, each time step would need its own mean and variance value.

3.3 Experiments Materials and Methods

3.3.1 Materials

The teleoperation system has two robots: a master robot (Quanser High Definition Haptic Device, or HD2) used directly by the therapist, and a slave robot (Yaskawa-Motoman SIA5F) handled by the patient. Even though both robots have upwards of seven DOF, the movements of the users and robots are constrained to only one DOF due to the nature of the cooperative task.

A potentiometer is used to measure the angle θ a bar attached to the Motoman makes with the horizontal axis. A mass is placed on the bar and allowed to slide along the length of the bar. Two identical springs attached to opposite sides of the bar pull the sliding mass towards their respective sides. Figure 3-3 shows the design of the bar.

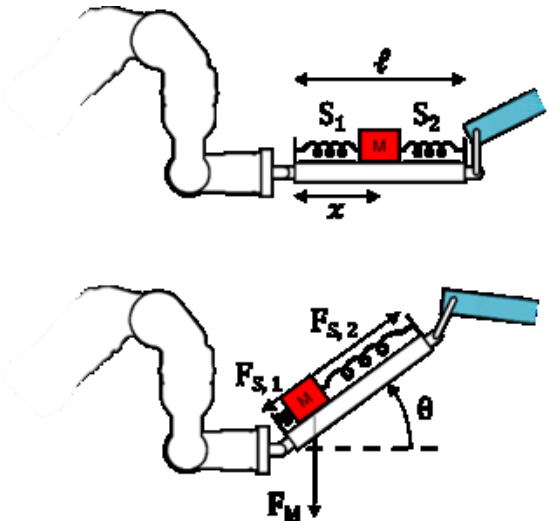

Figure 3-3 Design of the cooperative task. The slave robot holds one side of the bar, while the patient holds the bar from its other side. The bar's inclination can be altered in a 180° range.

3.3.2 Methods

3.3.2.1 Task

Hemiparesis after stroke can leave patients unable to lift their affected arm without significant difficulty or discomfort (Brami, 2003). As a result, performing ADLs can be

challenging for the patients. A common solution to these problems is to compensate the affected limb using the unaffected limb (Calinon, 2015). In other words, patients tend to use their unaffected limb to carry more weight during a given task, thereby neglecting their affected limb. This can result in a poor improvement of the affected limb and development of poor motor habits (Calinon, 2015). A commonly administered therapy activity for this kind of upper limb weakness involves having patients combine shoulder forward flexion, horizontal extension, abduction, and elbow extension to move the affected limb upwards and away from their trunk in a natural and synergistic manner. We utilize a modified version of this work; the task now requires the therapist and the patient to collaborate to lift a bar. The spring-mass system on top of the bar will allow the mass to slide towards one end of the bar in a manner directly proportional to θ, similar to if a box was being lifted with objects inside of it that slide back and forth freely. The therapist can thereby adjust the amount of force the patient must exert to lift the bar (i.e., the therapy intensity) by either lowering or raising his/her end, effectively resulting in an assistive/resistive therapy in a functional task context.

3.3.2.2 GMM and GMR Design

We use a GMM to learn the therapist's behavior during the task (trajectory of the movements) and GMR to reproduce them. The demonstration phase uses GMM to create K Gaussian distributions of dimensionality *D*. In this chapter, *K* has a value of 12 and *D* has a value of 3. *D* has as many dimensions as inputs to the GMM. These inputs are:

- Therapist position in the vertical axis (X_{Th})
- Patient position in the vertical axis (X_{Pa})
- Patient velocity in the vertical axis ($\dot{X}_{Pa}$)

These inputs are used in the GMM algorithm to learn the trajectory of the therapist's movements. We implement the GMM and GMR algorithms using the code presented in (Calinon, 2015).

The imitation phase uses GMR to retrieve the trajectory of the movements based on (Calinon, 2007). The GMR algorithm takes X_{Pa} and $\dot{X}_{Pa}$ as inputs, and based on the

GMM distributions, retrieves an appropriate value for X_{Th} as an output. Figure 3-4 shows the process of learning and reproducing the therapist's behavior with the given GMM inputs.

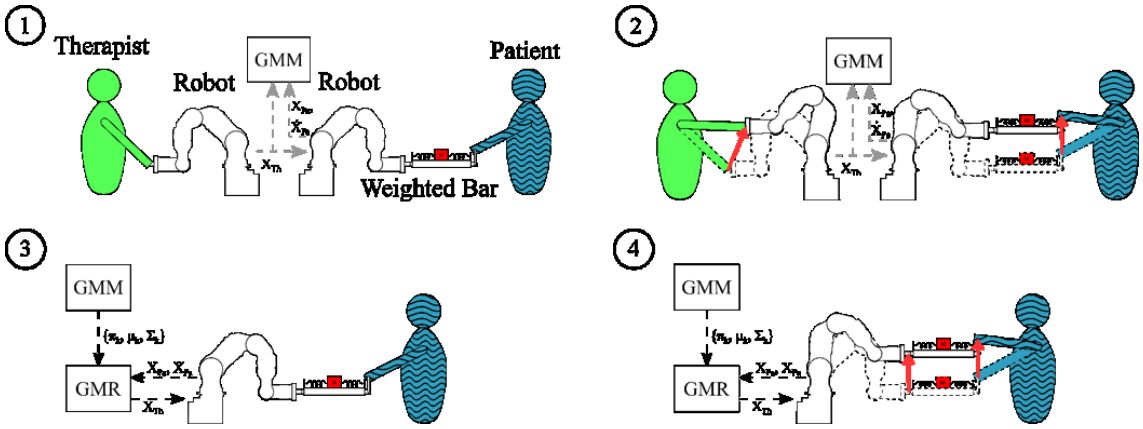

Figure 3-4 The two phases of LfD are shown in diagrams 1-2 and 3-4, respectively. In diagram 1, the therapist is present (making this the TIL phase). The patient will initiate movement as they lift the bar, to which the therapist will respond as shown in diagram 2. The data from both robots will be recorded and used to generate the GMM. Then in diagram 3, the patient is practicing in the absence of the therapist (the TOOL phase). The robot utilizes GMR to emulate the therapist and respond to the patient's movements, as shown in diagram 4.

3.3.2.3 Experiments

To show the accuracy and robustness of the system, we split the experiment into two phases. In the first phase, the system is trained to execute a single behavior, which could be to assist the patient, resist the patient or split the weight in a neutral way. These scenarios are also known as "plus 45", "negative 45" and "zero," corresponding to the angles at which the bar is held by the therapist to achieve assistance, resistance, or a neutral pose respectively. In this phase, the system is trained to keep a constant angle during the whole therapy/task/experiment. This implies that during this phase, only a single therapist behavior can be learned; if the system is trained to assist the patient, it will not be able to change that task unless the therapist records a different demonstration.

In the second phase, the system is trained with different scenarios. The idea is to create an adaptive system capable of assisting the patient, resisting the patient, or keeping a

neutral behavior with the patient, now based on the patient's performance. To do so, the GMM is trained with three different demonstrations of every scenario. Later, during the demonstration phase, the GMR takes the patient's behavior as input to reproduce the therapist behavior.

The system measures the patient's position and velocity to learn the patient's behavior. Based on the patient's velocity, the therapist can decide on how much assistance or resistance to apply. We selected four different general scenarios for the system to learn, described as follows. A positive and fast velocity means that the patient can easily perform the task and resistance can be applied to challenge the patient during the therapy. A positive and medium velocity means that the patient can perform the task and the therapist only has to keep/maintain a neutral behavior. A positive and slow patient's velocity means that the patient has some problems/difficulties in performing the task, so assistance is provided by the therapist. Finally, a negative patient's velocity means the patient is experiencing significant difficulty and is unable to perform the task. An even greater amount of assistance is then required to help the patient complete the task. These scenarios will also be referred to as "fast," "medium," "slow," and "back," respectively. Two block diagrams of the system are shown in Figure 3-5.

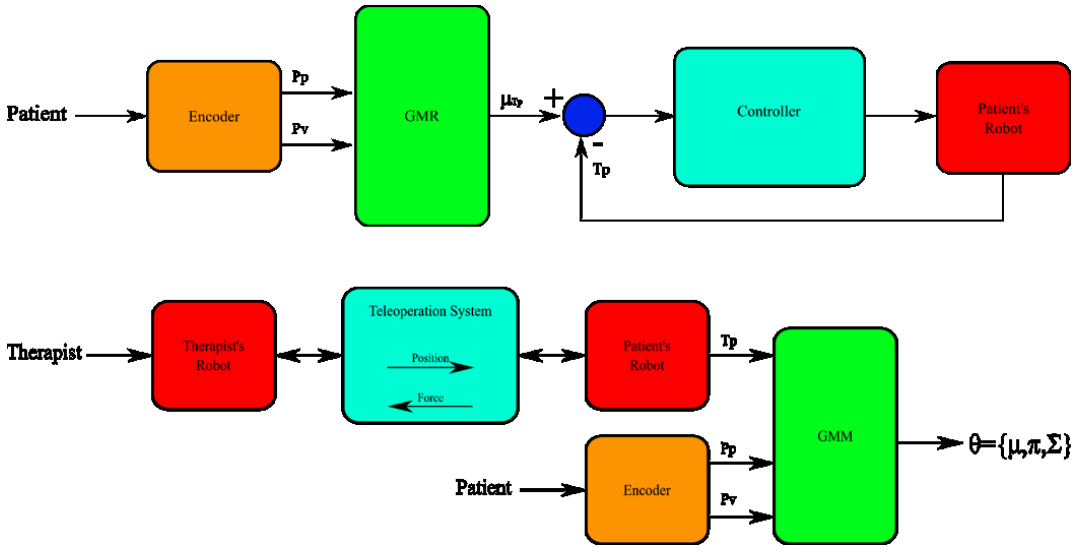

Figure 3-5 This figure summarizes the system's components needed to execute the task. The top figure shows a block diagram for the demonstration phase while the bottom figure shows a block diagram for the reproduction phase.

3.4 Results and Discussion

Two able-bodied participants played the roles of the therapist and the patient. During the experiments, the therapist's position, patient's velocity, and the patient's position data were recorded and then used in the GMM algorithm to train the robot. Three demonstrations were used to train the GMM. After the demonstration phase, the system's imitation performance was tested. The GMR model takes the patient position and patient velocity as inputs and returns the estimated therapist position as an output. This estimated therapist position was used to move the slave robot in the imitation phase. Therapist positions and patient positions are mainly used in the analysis of the results.

We present our results in two parts: first a qualitative examination of the system's imitation results for both phases, and second evaluation of the training data's efficacy.

3.4.1 GMR Output for Different Scenarios

GMR output results are shown in Figure 3-6 to Figure 3-14. In phase one, we show the results for assisting the patient, resisting the patient and splitting the weight in a neutral way for velocity-independent patient trajectories. For phase two, we show results for assisting, resisting or keeping a neutral behavior given velocity-specific patient trajectories.

The obtained plots show how the system can respond similarly to how a reference therapist would. Based on the input data, the GMR output estimated an accurate output through most of the different scenarios. Results for the first phase, shown in Figure 3-12 to Figure 3-14, show the system behaving close to the demonstrations, with clearly defined behaviors for assisting, resisting, or remaining neutral. In the second phase, the quality of the reproductions varies across the imitated behaviors. For patient trajectory data with higher velocities, GMR returns accurate trajectories with low variance, as in Figure 3-8. For slower velocities, however, velocity measurements are heavily affected by noise from hand tremor, muscle fatigue, etc.; the "slow" and "back" scenarios seen in Figure 3-6 and Figure 3-9 exhibit this problem. Results for these scenarios are less accurate, often switching between behaviors. A simulated data trajectory is also used in

Figure 3-10 to show the system's response to different scenarios given an ideal patient motion trajectory with minimal velocity fluctuation. In this situation the system produces very accurate results by the behaviors desired when training the GMM, indicating that the system could provide a perfect imitation for realistic inputs if properly adjusted. Knowing this level of accuracy is possible, this encourages further exploration of methods to adjust the training of the model to account for the variations above in patient data to achieve similar results for more realistic patient input. Finally, a real complex demonstration that combines multiple behaviors is used to show the robustness of the system, meant to resemble the interaction between a therapist and patient during a mock therapy exercise. The results show the system's behavior when presented with all the possible scenarios of patient motion, as well as including the transitions between scenarios. The system responds quite accurately throughout the task, but transitions are made too quickly to be safely implemented in clinical settings. Designing a different motion controller, for example, based on adaptive control principles, is an attractive possible solution.

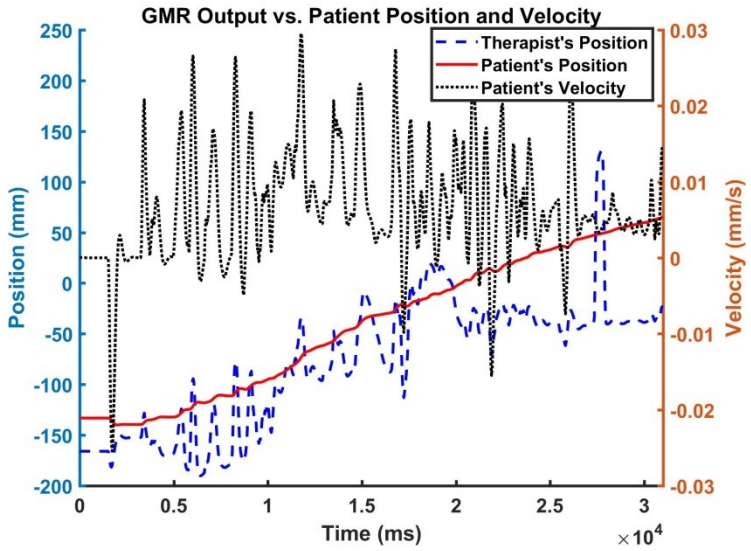

Figure 3-6 Obtained results for the slow scenario during the reproduction phase, the inputs are the patient's position (red solid line), patient's velocity (black dotted line), and the output is the therapist's expected position (blue dashed line).

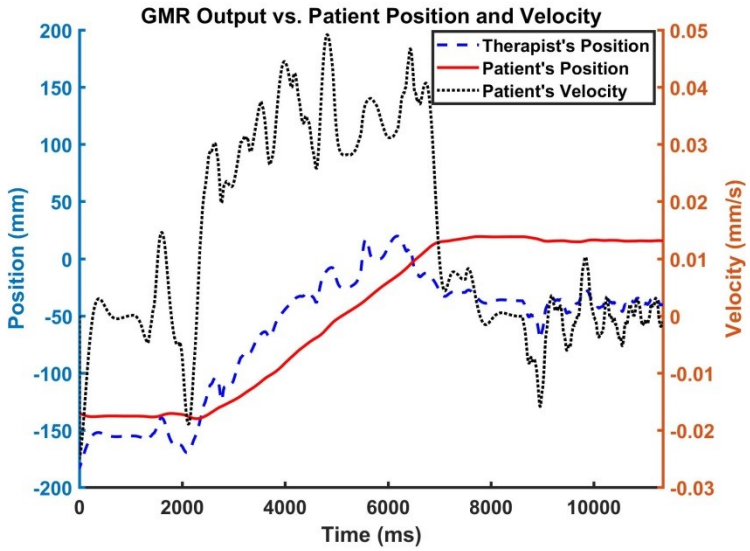

Figure 3-7 Obtained results for the medium scenario during the reproduction phase, the inputs are the patient's position (red solid line), patient's velocity (black dotted line), and the output is the therapist's expected position (blue dashed line).

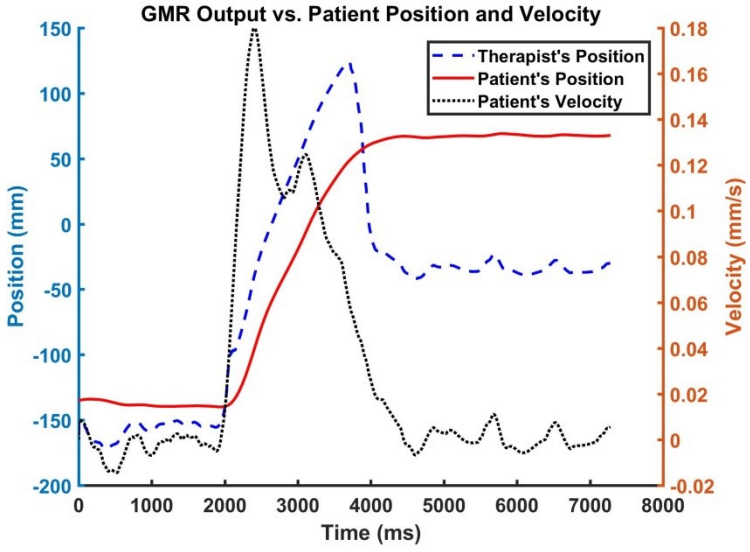

Figure 3-8 Obtained results for the fast scenario during the reproduction phase, the inputs are the patient's position (red solid line), patient's velocity (black dotted line), and the output is the therapist's expected position (blue dashed line).

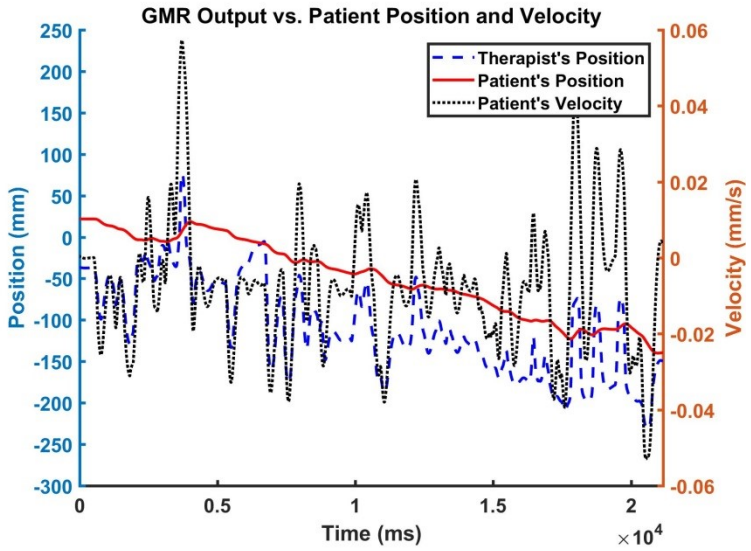

Figure 3-9 Obtained results for the back scenario during the reproduction phase, the inputs are the patient's position (red solid line), patient's velocity (black dotted line), and the output is the therapist's expected position (blue dashed line).

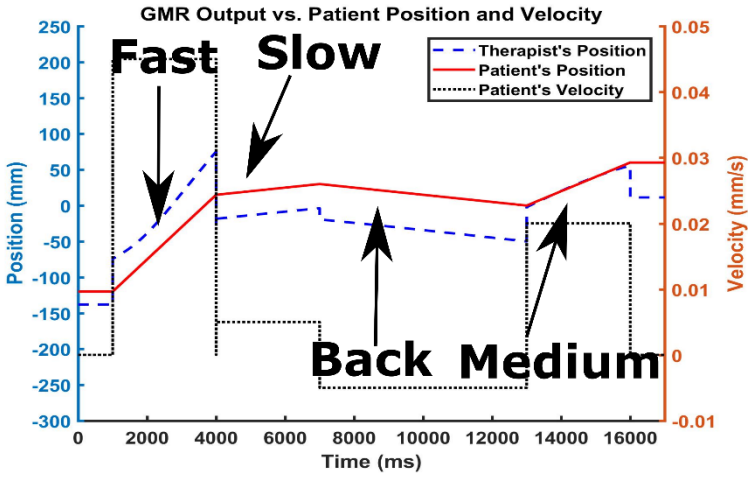

Figure 3-10 Obtained results for the simulated scenario during the reproduction phase, the inputs are the patient's position (red solid line), patient's velocity (black dotted line), and the output is the therapist's expected position (blue dashed line).

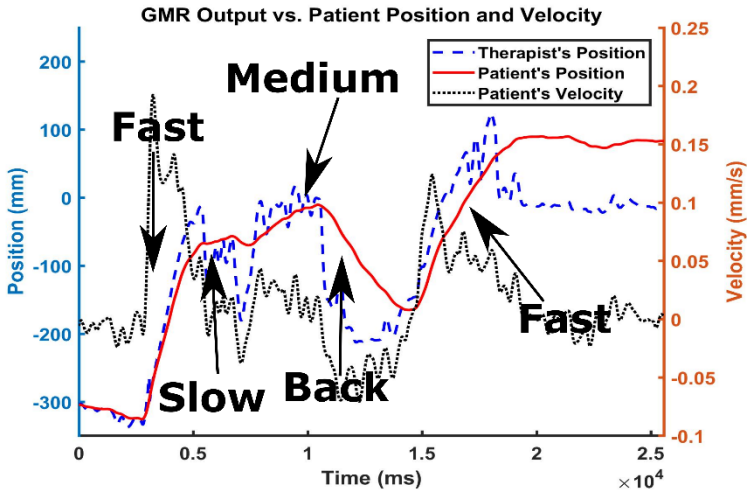

Figure 3-11 Obtained results for the multi-behavioral scenario during the reproduction phase, the inputs are the patient's position (red solid line), patient's velocity (black dotted line), and the output is the therapist's expected position (blue dashed line).

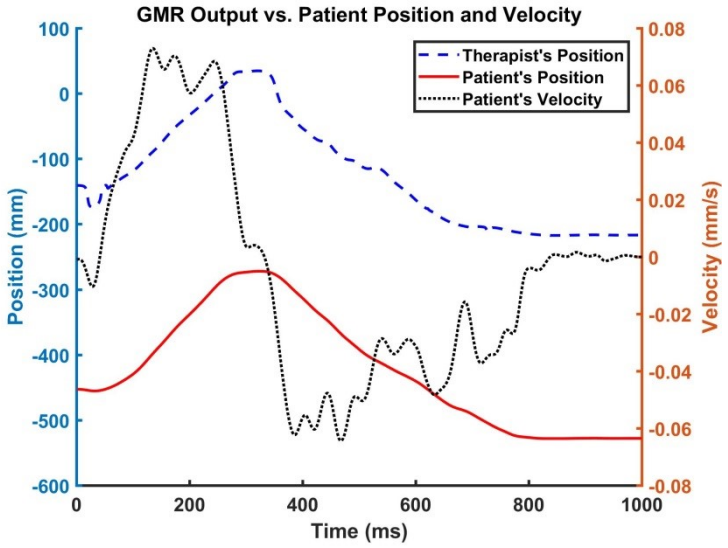

Figure 3-12 Obtained results for the plus 45 scenario during the reproduction phase, the inputs are the patient's position (red solid line), patient's velocity (black dotted line), and the output is the therapist's expected position (blue dashed line).

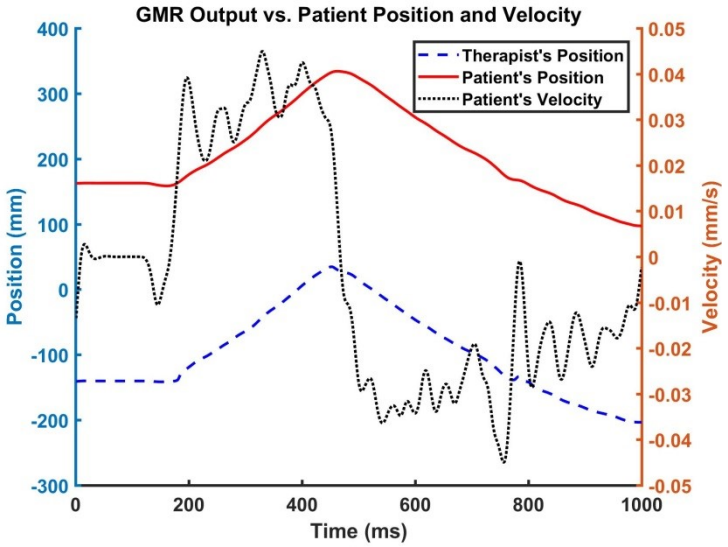

Figure 3-13 Obtained results for the negative 45 scenario during the reproduction phase, the inputs are the patient's position (red solid line), patient's velocity (black dotted line), and the output is the therapist's expected position (blue dashed line).

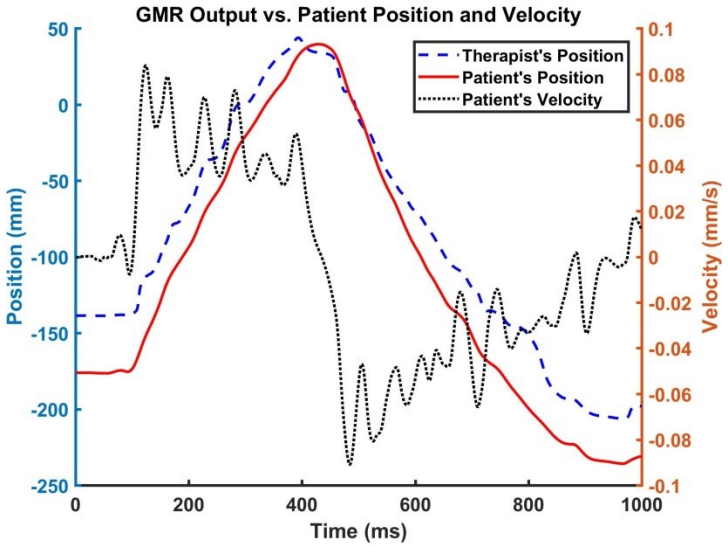

Figure 3-14 Obtained results for the zero scenario during the reproduction phase, the inputs are the patient's position (red solid line), patient's velocity (black dotted line), and the output is the therapist's expected position (blue dashed line).

3.4.2 Evaluation of Training Data Quality

Motivated by the previous results, we examine the efficacy of the dataset used to train the system. In the first experiment, a total of 12 demonstrations were recorded for training the GMM, with $N = 3$ demonstrations for each of the four behaviors. We now remove a single demonstration and use it instead as the input for the GMR process; this is performed for every demonstration dataset used for training. All demonstrations performed are assumed to be valid, i.e., the therapist's responses to the patient's actions are always intentional. By operating under this assumption, we can find demonstrations that are less useful if their trajectories are already included in the system. We quantify this as the error between the reference therapist trajectory, used to train the system, and the GMR output. Results are presented in Figure 3-15 to Figure 3-18. For most of the demonstrations, accuracy suffers from slower speed behaviors as mentioned previously. Standard deviation results indicate the reference trajectories are typically within GMR output for higher speeds Figure 3-16, while for lower speeds the trajectories differ from the demonstrated behavior Figure 3-15 and Figure 3-18.

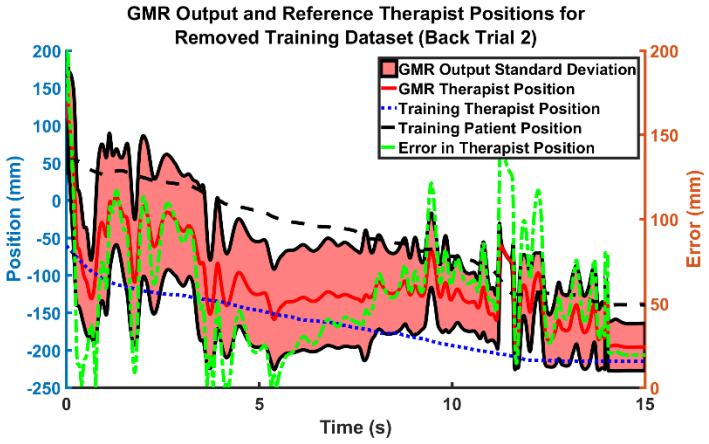

Figure 3-15 Analysis of the dataset quality for sample trials of back scenario in phase two.

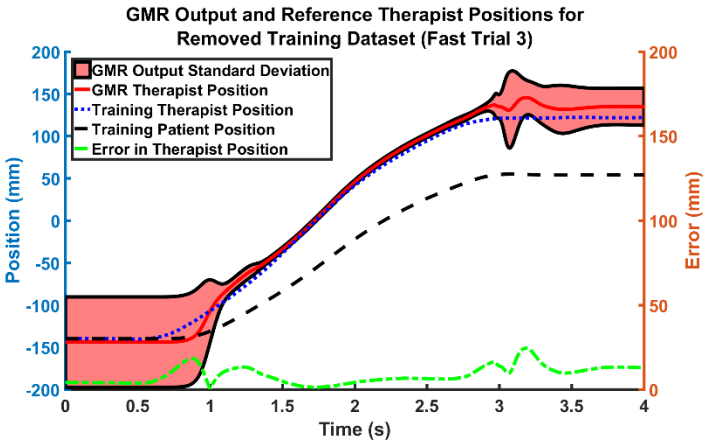

Figure 3-16 Analysis of the dataset quality for sample trials of medium scenario in phase two.

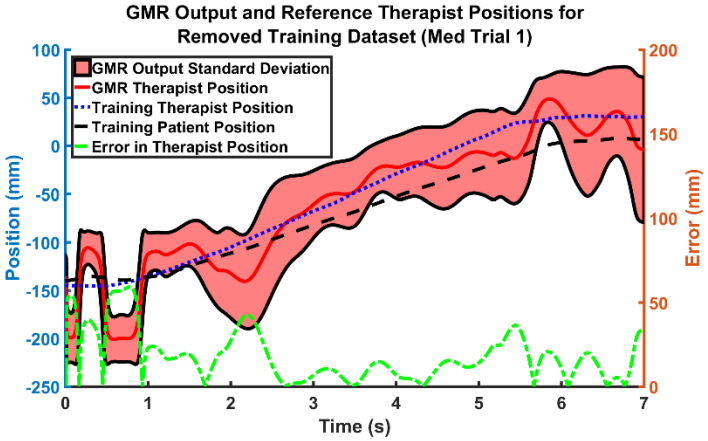

Figure 3-17 Analysis of the dataset quality for sample trials of medium scenario in phase two.

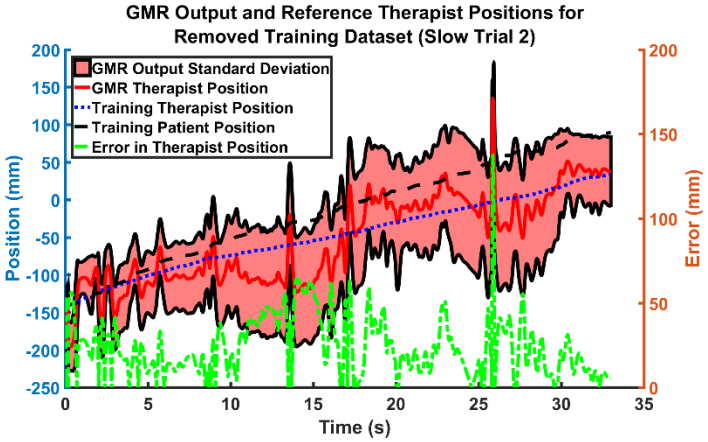

Figure 3-18 Analysis of the dataset quality for sample trials of slow scenario in phase two.

Table 3-1 shows the average error between the GMR output and the recorded therapist position for every removed demonstration. These results reinforce our earlier observation that responses to faster patient motions are captured better than slower movements with a limited number of demonstrations. By extension, we can infer that the system can better fit the Gaussian components to higher velocity data. Interestingly, results for overall average error of the slow and medium cases, in particular, are very similar. This may indicate that the GMM may not be able to distinguish between the velocities of the two cases well, resulting in equal sensitivities when demonstrations, in either case, are removed.

Table 3-1 Error between GMR output and recorded therapist position, averaged over the duration of the demonstration. Three trials are provided for each therapist behavior, which represents the removal of one of the three training demonstrations.

Trial Number	Average Error Between GMR Output and Recorded Therapist Position (mm)			
	Slow	Medium	Fast	Back
1	34.576	18.088	17.661	40.135
2	24.320	35.364	15.404	55.013
3	30.739	36.326	8.549	53.515
Overall Average	29.878	29.926	13.871	49.554

A resultant suggestion for works in the field of rehabilitation medicine looking to incorporate LfD principles would be to provide more demonstrations when aiming to imitate motions with large inherent variation, such as the slower movements seen in this work. Without resorting to more complicated regression methods with a greater focus on

ensuring stability, providing more demonstrations for trajectory spaces with high uncertainty is the simplest method of better defining task space behaviors. Otherwise, the problem could be reformatted as a non-linear dynamical system. Other methods, such as polynomial surface fitting, could conceivably provide a reasonably simple implementation of learning the task space but at the expense of ease of programmability.

3.5 Conclusion and summary

In this chapter, LfD techniques were applied to a cooperative therapy task performed through a telerobotic system. The demonstration and imitation phases of LfD were based upon GMM, and GMR approaches, respectively. The goal was to accurately replicate the therapist's actions during a cooperative object lifting task, in which assistance or resistance was provided by the therapist through holding the therapist's side of the object lower or higher, respectively. The velocity of the patient and positions of the therapist and the patient were recorded in the demonstration phase. Imitations of therapist movements were performed with the varying amount of success. Demonstrations provided by the therapist in response to faster patient movements were better learned, while slower patient movements had larger variations in velocity and produced less accurate imitations of the therapist's behavior. Examining the sensitivity of the system to the number of demonstrations provided for each scenario showed the differences between the GMR produced interactions, and those of a user representing a therapist were shown to be fairly small (between 8.54 and 17.66 mm) at higher patient velocities, but increased substantially (up to 55.51 mm) for more fine patient motions involving lower velocities due to the variance inherent to the motions.

4 LEARNING AND IMITATION OF SEMI-PERIODIC MOTIONS

4.1 Introduction

Machine learning has become one of the most powerful and popular tools in the last years (Calinon, 2009). It has been implemented in different fields such as civil engineering, mining, oil industry, hardware industry, finances, and computer sciences among others (Calinon, 2009). The robotics field is probably one of the most affected fields due to the fast growth and innovation introduced by machine learning (Calinon, 2009). Robots can be found almost in every field, from industrial automation to space exploration. In this chapter, we will be focusing on the medical application of robots especially in the rehabilitation and assistive technologies field. Due to the problems presented after a stroke episode, the combination of machine learning and medical robotics could bring a huge impact to the post-stroke rehabilitation as described in the following.

Stroke has become one of the most common causes of movement disorders worldwide (Federation, 2017) (Foundation, 2017). Symptoms associated with stroke often include loss of motor control, leading to difficulties in performing voluntary movements in upper and lower limbs (Association, 2018). As a consequence, people with disability are usually not able to perform activities of daily living (ADL) (Veerbeek, 2011).

To regain strength and mobility, the brain can rewire neurological pathways through therapy exercises that engage patients in repetitive tasks (Lum P. R., 2002). Traditionally, a therapist interacts with the patient throughout the therapy to guide the patient through different tasks. The task and the intensity, length, and complexity of the therapy are

chosen by the therapist depending on the patient's needs. A common target for post-stroke therapy is to regain the ability to perform ADLs. ADLs are widely used because the level of patient's success in such activities is a metric for the effectiveness of the rehabilitation process (Legg, 2006).

During post-stroke traditional therapy, and to complete a given task, the therapist usually holds the patient's affected limb. This technique is called hand-over-hand therapy and is widely used to guide the patient through the task and to get feedback about the patient's abilities (Hogan, 1992). Based on the obtained feedback, the therapist can decide either to help the patient to complete the task or to make the task more challenging; this is known as assistive or resistive therapy, respectively (Atashzar, 2012). When the patient is not able to complete the task by himself/herself, the assistance needed is provided by the therapist. Conversely, resistance can be provided by the therapist to give an extra challenge to the patient and build muscle strength. Every patient needs a different level of assistance or resistance.

Due to the increasing number of people with disability, the demand for therapeutic services has also increased. Therapists and therapy resources are limited and cannot answer all requests for therapeutic services (Tao, 2014). Also, hand-over-hand movement therapy lacks repeatability and objective measures of patient performance and progress and is also physically tasking for therapists (Najafi, 2017). To respond to this increasing burden on the healthcare system, robotics-assisted rehabilitation has emerged as an efficient and cost-effective solution. With robot-assisted therapy, the duration and number of training sessions can be increased while reducing the workload of therapists.

Rehabilitation robots are generally pre-programmed to execute a predefined task (Tavakoli, 2018). This approach does not take advantage of the robots' abilities to adapt to different situations and patients. Robots reprogramming by demonstration allows adjustment of the parameters of therapy on the fly. Since these adjustments are made by therapists based on the patient's needs and motion tolerances, the rehabilitation therapy is made more personalized and natural.

If the robot can learn the required task-specific assistance from the therapist, only a short interaction between the therapist and the patient is sufficient. Then, in the therapist's absence, the robot autonomously assists the patient similar to the therapist when performing the same task.

The initial short interaction between the therapist and the patient is beneficial as it leverages the therapist's experience and allows to combine the therapist's skills and experience with robotic therapy. As mentioned before, therapists have the knowledge and skill to determine the required assistance or resistance for a given patient in a given phase of recovery and are also able to modify or adapt the given task based on patient's needs. Because robots do not have this ability, a therapist has to be involved at least for a short duration at the beginning of rehabilitation therapy.

In this chapter, we propose to use LfD as a solution to reprogram rehabilitation robots based on observing a brief window of therapist-patient interaction. The proposed LfD algorithm allows the robot to be reprogramed as a therapist moves the robot while it is in a passive mode; this teaching method is known as kinesthetic teaching (Lee, 2012). This step is known as the learning or demonstration phase. Later, when the therapist is no longer with the patient, the robot reproduces the learned therapist's behavior; this step is known as the reproduction phase. In kinesthetic teaching, the therapists do not need to have any knowledge about robot programming.

The chapter is divided as follows: first, we explain the related works in Section 4.2. A description of the task is presented in Section 4.3. Section 4.4 describes the LfD algorithms to be used. Experiments are presented in Section 4.5. Section 4.6 presents the obtained results. Finally, conclusions and future work are presented in Section 4.7.

In this chapter, therapists and patients were *simulated*. Along this thesis, the word *therapist* refers to a person taking the role of the therapist. Notice that this person does not have therapeutic skills. The word *patient* refers to a non-disabled person that took the role of a person with disabilities. Different devices such as springs and transcutaneous electric nerve stimulator (TENS) were used to simulate the motor impairments.

4.2 Related Work

As mentioned before, ADLs are widely used in rehabilitation as a basis to design training strategies to regain strength and mobility (Guidali, 2011). Rehabilitation tasks can be divided into reaching tasks or functional tasks. Regardless of this categorization, the motions involved in a task can be either non-periodic or periodic. As described next, most of the past research has focused on learning and imitation of the non-periodic trajectories.

As mentioned in (Calinon, 2006), (Billard., 2009), (Konidaris, 2012), previously developed algorithms can be used for learning reaching motion tasks. Given that some of the most common ADLs such as brushing the teeth, involve periodic motions, it could be hard to learn and reproduce them using the current algorithms. Our goal is to extend the current LfD techniques to learn not only reaching motions but also repetitive motions, allowing therapists to use rehab robots in more diverse ADL tasks. This ability should bring benefits regarding the recovery time and the quality of the therapy.

Our research group has developed robotic rehabilitation schemes that try to improve therapies through the incorporation of LfD. In (Tao, 2014), the author proposed a paradigm called learn and replay (LAR) to realize direct bilateral telerehabilitation that encompasses two distinct phases to achieve time-sharing of a therapist. In the first phase, the system learns the therapist's impedance. Later, in the second phase, the system uses the learned impedance to reproduce the therapist behavior in his/her absence. In Chapter 3, a telerobotic cooperative rehabilitation system based on GMM and GMR is used for learning and imitation phases, respectively. In the learning phase, the therapist interacts with the patient in a cooperative task. Later, in the demonstration phase, the system learns the therapist's position-based behavior and replicates it in his/her absence when the patient is alone interacting with the robotic system.

In (Maaref, 2016), the authors developed a robot-assisted rehabilitation system for co-operative therapy combining LfD and Assist-as-Needed. In the demonstration phase, the system learns the therapist's impedance using GMM. Later, GMR is used to build a model of the therapist's behavior. Based on the difference between the patient's performance

and the learned therapist's behavior, the method determines whether to assist the patient completing the task or not.

One of the most important limitations of the previously presented work is the fact that they can only be used to learn reaching tasks. However, as mentioned before, ADL are not limited to reaching motions. In this chapter, as mentioned before, a special interest in periodic motion tasks such as brushing the teeth, combing the hair, cleaning a surface with a cloth, cutting vegetables, amongst others. The lack of an ability to learn repetitive motion tasks constraint the capabilities of the LfD-based robotic systems in the rehabilitation field. While it is theoretically possible to segment a semi-periodic motion trajectory into small pieces, where each piece represents a reaching motion, this learning approach has limitations. Besides the memory requirements to store the segmented data points, the most important limitation is the fact that it does not leverage or learn the periodicity information.

Thus, an improvement is needed for a robot to learn periodic tasks. The contributions of this chapter are to use, implement and combine LfD algorithms to be able to learn repetitive motion tasks using Fast Fourier Transform (FFT) and Fourier Series (FS) as well as Gaussian Mixture Model (GMM) and Gaussian Mixture Regression (GMR). While the GMM/GMR based algorithms were created to learn how to reach a target in the spatial domain, our system tries to learn how to reach a frequency and amplitude in the frequency domain.

4.3 Task Description

While the proposed framework is task-independent in the sense that the system can learn and reproduce any given repetitive behavior, two representative tasks are considered here. For the first task, we have a 1-DOF (degree of freedom) motion while for the second task we have a 2-DoF motion. In the first task, the goal is to move the robot's end effector in a repetitive way between two points by following a desired frequency as in Figure 4-1 (Left). Notice that in this scenario, the robot must learn to perform the repetitive task following the demonstrated frequency and with the demonstrated amplitude; while in a purely motion

task between two points, periodicity following a desired frequency is not possible due to the lack of information used to train the system. In the second task, the goal is to move the robot's end effector along a 2D path in a repetitive way as shown in Figure 4-1(Right). The robot must learn to perform the repetitive task following the demonstrated frequency and on the demonstrated path.

The next section proposes two different learning methods. The first method relies on FS while the second method combines FS, GMM, and GMR. These two methods and the two tasks mentioned above create four possible combinations (or scenarios), three of which are implemented in this chapter. The first scenario combines the first method and the first task. The second scenario combines the first method and the second task. The third scenario combines the second task and the second method.

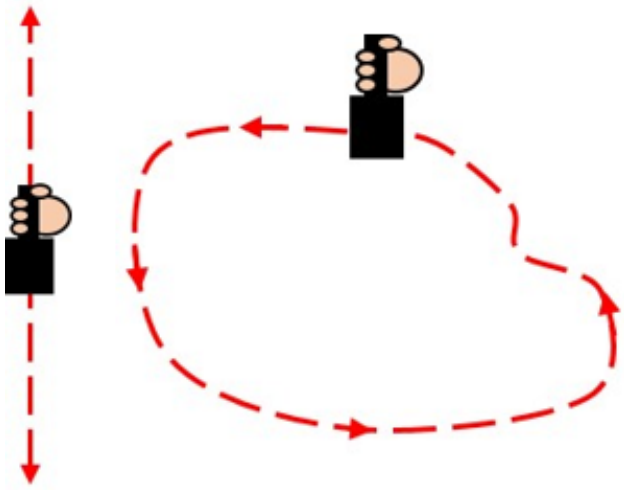

Figure 4-1 The left figure shows a diagram for task 1, while right figure shows a diagram for task 2. In both tasks, the patient has to follow a desired trajectory while moving the robot's end effector.

The task has to be divided into two phases. In the learning phase, the therapist moves the robot to execute the task for a short duration. The robotic system uses position sensors to record the therapist's trajectory. Based on the recorded data, the system builds a model of the demonstrated behavior. Later, in the reproduction phase, when the therapist is no longer interacting with the robot, the robotic system reproduces the demonstrated trajectory. Due

to the presence of a patient in the reproduction phase, the robotic system will measure the patient's behavior, compare it with the demonstrated behavior, and only compensate for the difference between the behavior demonstrated by the therapist and that shown by the patient. Figure 4-2 shows a diagram of the task.

Details related to this controller are presented in Section 4.4.

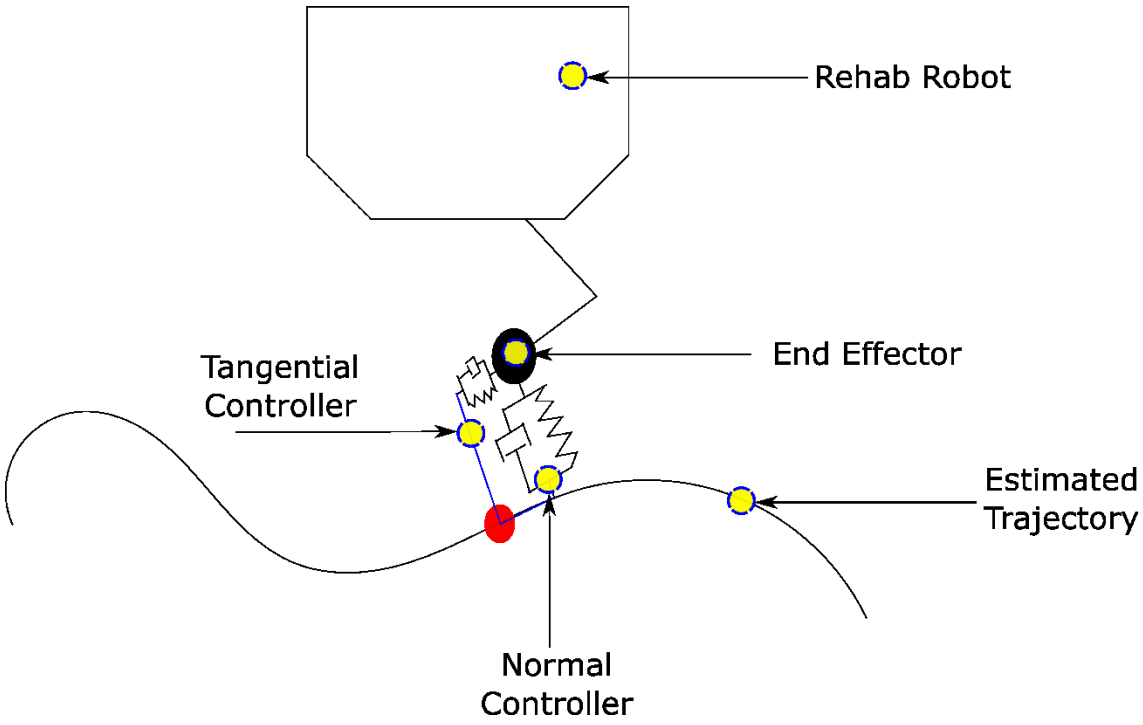

Figure 4-2 First, during the demonstration phase, the therapist performs the task for a short time. Then, using robot learning from demonstration, the task is modeled as an average trajectory and variations in trajectory. The controller assists the patient by two varying impedance models (spring-damper) to follow the demonstrated trajectory and remain in the demonstrated range of variability.

4.4 Proposed Framework

One solution to the high demand for therapeutic services is to learn the therapist's behavior and replicate it for the patient later when the therapist is unavailable. Given the nature of the proposed task, Fourier Series (FS) is considered as an easy and accurate way to model the therapist's behavior. Using FS, the therapist's semi-periodic motion can be

modeled as a sum of sinusoidal and cosines signals, where each signal has a particular frequency and amplitude.

In the first and second scenarios, the main idea is to determine the frequencies and amplitudes of the FS terms. Based on the obtained results, the system models and learns the therapist's behavior in the demonstration phase. Later, during the reproduction phase, the system compares the patient's performance against the therapist's demonstrated behavior, and by using a position controller, the robotic system reduces the difference between the patient and therapist's performance. The third scenario involves the combination of GMM, GMR, and FS. As in the previous scenarios, FS is used to model the therapist's behavior. The novelty, compared to the previously implemented methods emerges in the combination of FS, GMM and GMR. In this section, we explain the demonstration and reproduction phases with detail.

4.4.1 First Scenario

4.4.1.1 Demonstration Phase

Using an FS (Kido, 2015) and based on the simple repetitive task described in Section 4.3, the therapist's motion is modeled as

$$f(t) = a_0 + \sum_{n=1}^{N} a_n Cos(nwt) + b_n Sin(nwt) \tag{4-1}$$

where

$$a_0 = \frac{1}{M} Re(F_0) \tag{4-2}$$

$$a_n = \frac{2}{M} Re(F_n) \tag{4-3}$$

$$b_n = \frac{2}{M} Im(F_n) \tag{4-4}$$

where a_0 is a constant term, *w* is the fundamental frequency of the signal, *N* is the number of harmonics in the series, *t* is the time, *M* the number of data points in the signal $f(t)$, *F* is the n^{th} Discrete Fourier Transform (DFT) of $f(t)$, and *T* is the period of the signal. Notice that $\Delta = \{a_0, a_n, b_n\}$ are unknown parameters, this parameters are also known as Fourier Series Coefficients (FSC). In order to find these parameters, **(4-1)** is used in combination with **(4-2)**, **(4-3)**, and **(4-4)**.Once we estimate the Δ parameters, the FS given by **(4-1)** is used as a generalization of the therapist's behavior. This newly learned behavior is used in the reproduction phase to help the patient to complete the given task.

4.4.1.2 Reproduction Phase

During the reproduction phase, the system measures the patient's performance and compares it versus the behavior previously demonstrated by the therapist and modeled in **(4-1)**. The difference between the two behaviors is used as the input to the robot position controller. The position controller takes the error and keeps the patient's behavior as close as possible to the therapist's demonstrated behavior.

As shown in Figure 4-2, the controller assists the patient by two varying impedance models (spring-damper) to follow the average demonstrated trajectory and remain in the range of variability of the demonstrated trajectories. One spring-damper system assists the patient along the tangent to the trajectory, while the second spring-damper system assists the patient along the normal to the trajectory. The controller along the normal gives the patient the freedom to move within a small range of motion close to the estimated position, while the controller along the tangent assist the patient in following the right position and frequency.

To implement these controllers, tangential and normal lines to the estimated trajectory have to be found for every given point. The tangent can be found through the slope *m* between two consecutive points as

$$m = \frac{y_1 - y_2}{x_1 - x_2} \quad (4\text{-}5)$$

where (x_1, y_1) is the previous estimated position and (x_2, y_2) is the current estimated position. Once we have the slope of the tangent, we have to compute the angle between the fixed frame and the dynamic frame is computed as follows:

$$\theta = \tan^{-1}(m) \quad (4\text{-}6)$$

Finally, the current robot's end effector position has to be transformed from the fixed to the dynamic frame. To do that, we use the following equations:

$$x_r = x_o \cos\theta + y_o \sin\theta \quad (4\text{-}7)$$

$$y_r = -x_o \sin\theta + y_o \cos\theta \quad (4\text{-}8)$$

where (x_o, y_o) is the robot's end effector position in the fixed frame, while the (x_r, y_r) is the robot's end effector in the dynamic frame.

Once the robot's position is represented in the dynamic frame, we can use a simple PD controller along each axis of the new frame. In this way, a spring-damper system can control the robot along the tangent and normal lines. The previous transformations can be executed inside the controller block of the block diagrams of Figure 4-4 and Figure 4-6.

Notice that this frame changes for every new estimated point. Therefore, this computation runs through the whole therapy time.

Figure 4-3 illustrates a diagram of the fixed and the dynamic frames.

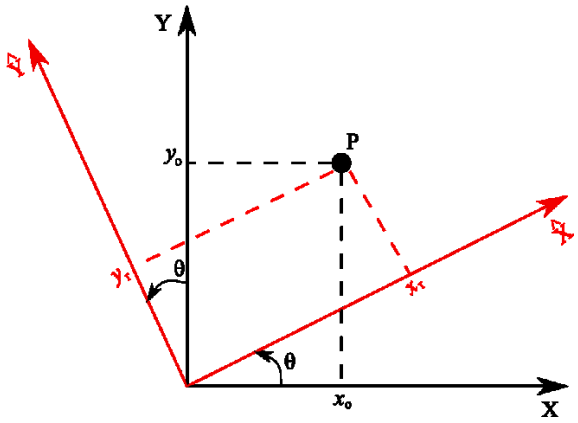

Figure 4-3 Robot's end-effector representation in both fix (black) frame (X,Y) and dynamic (red) frame $(\hat{X},\hat{Y})$.

4.4.2 Second Scenario

4.4.2.1 Demonstration Phase

As mentioned in Section 4.3, the second task involves a 2D periodic motion. The 2D motion can be decomposed into two 1D motions. The idea is to use the same approach as in the first scenario for X-axis and Y-axis independently.

4.4.2.2 Reproduction Phase

The reproduction phase is similar to the one used for Scenario 1. Because Scenario 2 has 2D, the reproduction has to be implemented for each axis. The system controls X-axis and Y-axis independently.

Notice that any D-dimensional periodic movement can be easily learned, generalized and reproduced using this algorithm. The fact that the 1D task can be scaled to any dimension shows the power of this algorithm. Figure 4-4 shows a block diagram for the reproduction phase used in scenarios 1 and 2. In this figure, x,y represent the current robot's end effector position, $\tilde{x},\tilde{y}$ represent the estimated position during the reproduction phase, E represents the error between the current position and the estimated position, and F represents the force needed to move the robot to the estimated position.

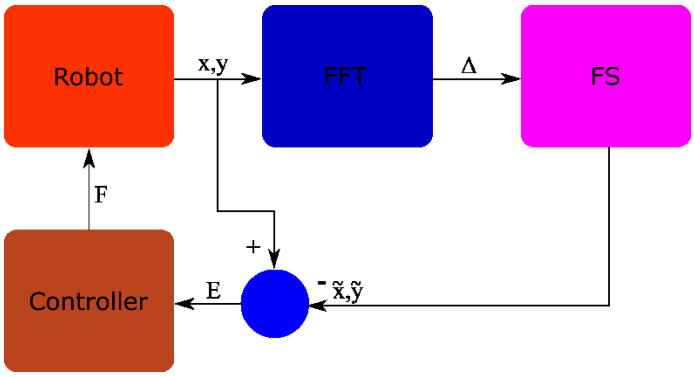

Figure 4-4 Block diagram of the system used for Scenarios 1 and 2.

4.4.3 Third Scenario

GMM and GMR are widely used in LfD due to their robustness and well-defined implementation. The way they work can be explained as follows: The GMM algorithm takes N demonstrations of p data points. The algorithm clusters the data points in K different Gaussians, where each Gaussian has its own mean and covariance matrix. A mixture of Gaussians is later used to combine the previously computed Gaussians. This is used by the GMR to reconstruct the output. The GMR algorithm uses the previously computed GMM data and, based on the inputs; it retrieves an estimation of the expected output (Calinon, 2007). We implement the GMM and GMR algorithms using the code presented in (Calinon, 2015).

4.4.3.1 Demonstration Phase

During the demonstration phase, the GMM algorithm is used to build a model based on the therapist's behavior.

The GMM is a weighted sum of K component Gaussian densities, and it is given by the equation,

$$p(\xi_j) = \sum_{k=1}^{K} p(k)p(\xi_j|k) \qquad (4\text{-}9)$$

where $p(k)$ is priors, $p(\xi_j|k)$ is the conditional density function and ξ_j is a D-dimensional continuous-valued dataset vector defined by $\{\xi_j\}_{j=1}^N = \{\xi_{p,j}, \xi_{\Delta,j}\}_{j=1}^N$. The dataset consists of *N* datapoints of a 2D component in polar coordinates ($\xi_{p,j}$), and a 3D component with the Δ values ($\xi_{\Delta,j}$). The variables in **(4-9)** are defined as

$$p(k) = \pi_k \tag{4-10}$$

$$p(\xi_j|k) = \mathcal{N}(\xi_j; \mu_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^D|\Sigma_k|}} e^{-\frac{1}{2}((\xi_j-\mu_k)^T \Sigma_k^{-1}(\xi_j-\mu_k))} \tag{4-11}$$

Thus, each Gaussian component is described by the parameters $\Omega_i = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K$, representing prior probabilities, mean vectors and covariance matrices, respectively.

To get the maximum-likelihood estimation of the mixture parameters, the Expectation-Maximization (EM) algorithm has been widely used in the literature (Calinon, 2012). By starting from a rough estimation of the Ω parameters by k-means segmentation, it takes the GMM parameters and iterates them until convergence is reached. To guarantee an increase of the likelihood, this algorithm uses a simple local search technique. The equations for the EM algorithm are given by the following.

E-step:

$$p_{k,j}^{(t+1)} = \frac{\pi_k^{(t)} \mathcal{N}(\xi_j; {\mu_k}^{(t)}, {\Sigma_k}^{(t)})}{\sum_{i=1}^K \pi_i^{(t)} \mathcal{N}(\xi_j; {\mu_i}^{(t)}, {\Sigma_i}^{(t)})} \tag{4-12}$$

$$E_k^{(t+1)} = \sum_{j=1}^N p_{k,j}^{(t+1)} \tag{4-13}$$

M-step:

$$\pi_k^{(t+1)} = \frac{E_k^{(t+1)}}{N} \tag{4-14}$$

$$\mu_k^{(t+1)} = \frac{\sum_{j=1}^{N} p_{k,j}^{(t+1)} \xi_j}{E_k^{(t+1)}} \tag{4-15}$$

$$\Sigma_k^{(t+1)} = \frac{\sum_{j=1}^{N} p_{k,j}^{(t+1)} (\xi_j - \mu_k{}^{(t+1)})(\xi_j - \mu_k{}^{(t+1)})^T}{E_k^{(t+1)}} \tag{4-16}$$

The iteration stops when $\frac{\mathcal{L}^{(t+1)}}{\mathcal{L}^{(t)}} < C_1$, with the log-likelihood $\mathcal{L}$ defined as

$$\mathcal{L}_\Theta = \frac{1}{N} \sum_{j=1}^{N} \log(p(\xi_j)) \tag{4-17}$$

where C_1 is a selected threshold for convergence and t represents the current value while t+1 represents the estimated or new value. Figure 4-5 shows a block diagram of the EM algorithm.

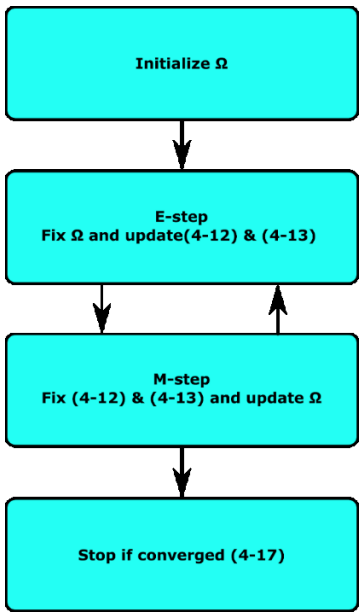

Figure 4-5 Block Diagram of EM algorithm

In this phase, a single 2D periodic motion demonstration was recorded. Using **(4-2)**, **(4-3)**, and **(4-4)**, FSC was computed from each demonstration. Given the model and

characteristics of the demonstrated behavior, polar coordinates (r,θ) for every single time step are computed. These computed values and the FSC are the inputs to the GMM. The top diagram in Figure 4-6 shows a visual explanation of the demonstration phase. The obtained GMM parameters will be used by the GMR in the reproduction phase.

4.4.3.2 Reproduction Phase

In the reproduction phase, the previously computed Ω parameters and the GMR algorithm are used to retrieved and estimate Δ that will be used in **(4-1)**. The GMR algorithm uses the Gaussian conditioning theorem and linear combination properties of Gaussian distributions to retrieve the values (Calinon, 2009).

The GMR algorithm uses temporal values (ξ_p) as query points to estimate the corresponding values ($\hat{\xi}_s$) through regression. Given a set of values for the kth component of a GMM, the representation of mean and covathe riance matrix is

$$\mu_k = \{\mu_{p,k}, \mu_{\Delta,k}\}, \qquad \Sigma_k = \begin{pmatrix} \Sigma_{p,k} & \Sigma_{p\Delta,k} \\ \Sigma_{\Delta p,k} & \Sigma_{\Delta,k} \end{pmatrix} \tag{4-18}$$

The conditional expression and estimated conditional covariance of $\xi_{\Delta,k}$ given ξ_p are calculated for each GMM component *k* as

$$\hat{\xi}_{\Delta,k} = \mu_{\Delta,k} + \Sigma_{\Delta p,k}(\Sigma_{p,k})^{-1}(\xi_p - \mu_{p,k}) \tag{4-19}$$

$$\hat{\Sigma}_{\Delta,k} = \Sigma_{\Delta,k} - \Sigma_{\Delta p,k}\left(\Sigma_{p,k}\right)^{-1}\left(\Sigma_{p\Delta,k}\right) \tag{4-20}$$

equations **(4-12)** and **(4-13)** are then used to get the probability that the kth component is responsible for ξ_t as

$$p(\xi_\Delta|\xi_t) = \sum_{k=1}^{K} \beta_k \mathcal{N}(\xi_{s,k}; \hat{\xi}_{s,k}, \hat{\Sigma}_{s,k}) \tag{4-21}$$

$$\beta_k = \frac{p(p|k)}{\sum_{i=1}^{K} p(\xi_p|i)}$$

condition expectation ($\hat{\xi}_\Delta$) and conditional covariance ($\hat{\Sigma}_\Delta$) of the ξ_Δ given ξ_p are calculated using equations **(4-13)**, **(4-14)** and **(4-15)** for a mixture of all *K* Gaussian components:

$$\hat{\xi}_\Delta = \sum_{k=1}^{K} \beta_k \, \hat{\xi}_{\Delta,k} \quad , \quad \hat{\Sigma}_\Delta = \beta_k^2 \hat{\Sigma}_{\Delta,k} \tag{4-22}$$

The GMR model only needs the means and covariance matrices of the GMM to retrieve the signal. This helps to use memory more efficiently. To compute these values, GMR needs two inputs. The first input is the GMM parameters, while the second one is a periodic position vector in polar coordinates. Given a periodic motion, the GMR can compute the estimated FSC. Figure 4-6 shows a block diagram of the proposed algorithm. In this figure, *x,y* represent the current robot's end effector position, $\tilde{x}, \tilde{y}$ represent the FS estimated position, $\hat{x}, \hat{y}$ represent the GMR estimated position, *E* represents the error between the FS estimated position and the estimated position computed by the GMR, $\hat{\mu}$ is the mean value obtained by the GMR (estimated FSC), and *F* represents the force needed to move the robot to the computed estimated position.

In the next section, an experiment demonstrates the power and capabilities of these new algorithms.

4.5 Experiments

To perform the experiment, we used a planar haptic-enabled rehabilitation robot (Quanser, Inc. Markham, Canada) (Lu, 2012).

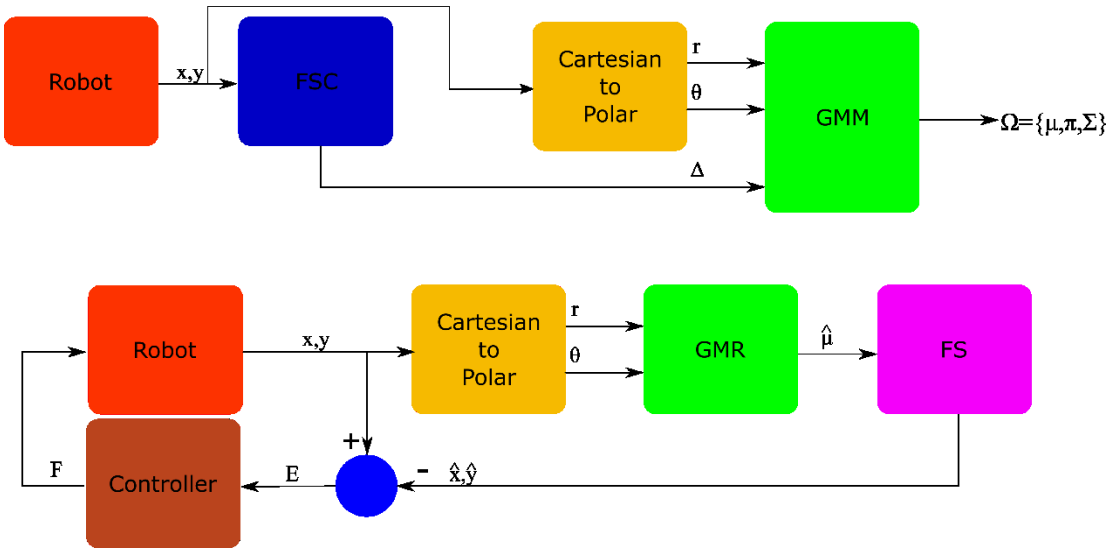

Figure 4-6 Block diagrams of the system used for task three. The top figure shows a block diagram to describe the learning phase. The bottom figure shows a block diagram that describes the reproduction phase.

There are two different ways to implement the learning and imitation of therapy. Figure 4-7 shows a parallel operation mode while Figure 4-8 shows a sequential operation execution. In the sequential execution, first, the therapist interacts with the robot to train the system and later in the absence of the therapist and during the reproduction phase, the patient interacts with the robot. The parallel execution is similar to the sequential execution with the difference that during the training phase, the therapist and the patient work at the same time with the robot. Even though the proposed learning and imitation framework can execute both modes, we will focus on the sequential mode in the rest of this chapter.

So far, we have tested the system with a non-disabled user. The feasibility and efficacy of the proposed framework are evaluated by conducting the experiment simulating an adult with cerebral palsy symptoms by using transcutaneous electrical nerve stimulation and a spring array. Figure 4-9 and Figure 4-10 show the setup for the demonstration and reproduction phases, respectively. Future implementation and studies with real patients are discussed in Chapter 6.

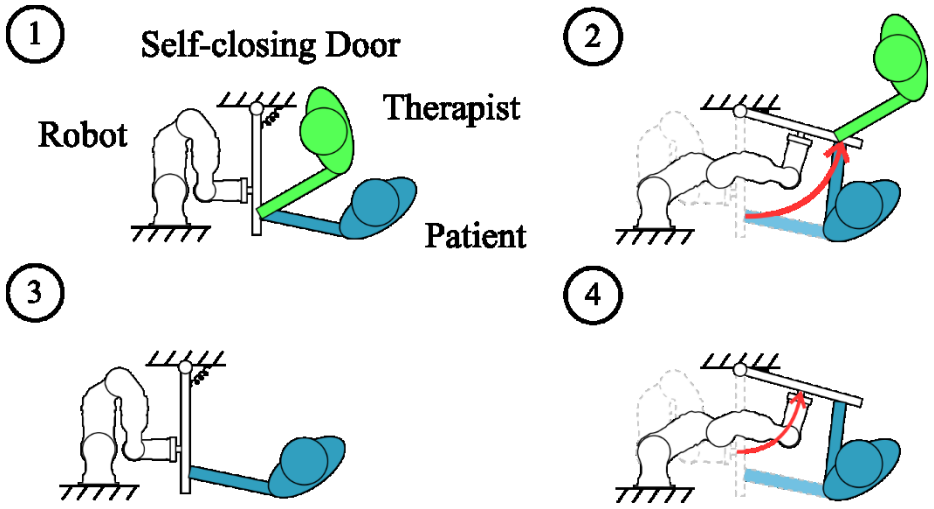

Figure 4-7 Parallel training mode. During the training phase (stage 1 & 2) the therapist and the patient interact with the robot to teach it about the task performance. Later, during the reproduction phase (stage 3 & 4), the therapist is no longer involved in the therapy. The robot helps the patient to perform the task as taught by the therapist.

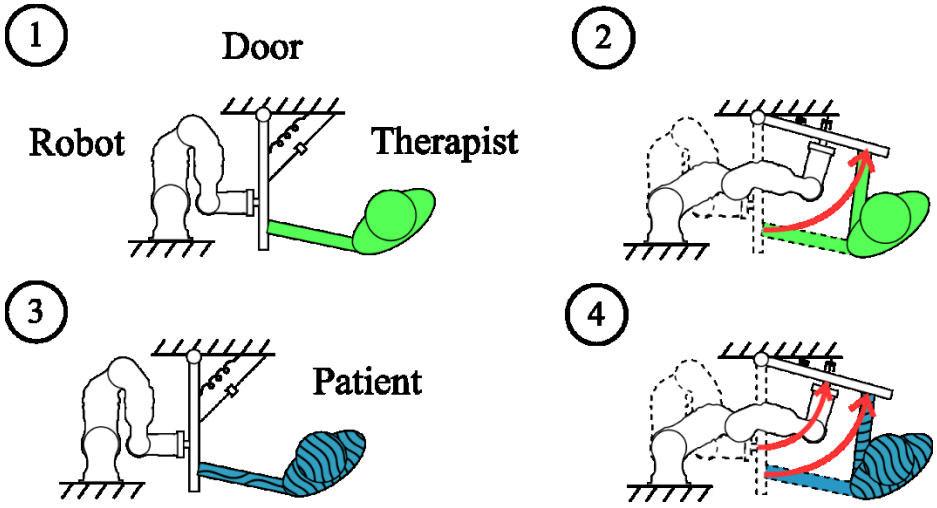

Figure 4-8 Sequential training mode. During the training phase (stage 1 & 2) the therapist teaches the robot about the task performance. Later, during the reproduction phase (stage 3 & 4), the therapist is no longer involved in the therapy. The robot helps the patient to complete the task as taught by the therapist.

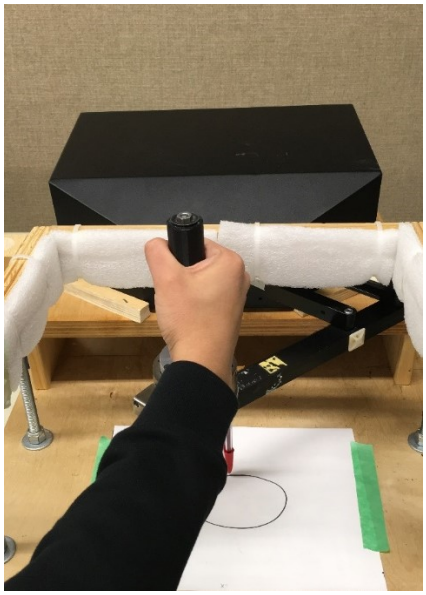

Figure 4-9 Figure showing the demonstration phase. The therapist interacts with the robot to train the system.

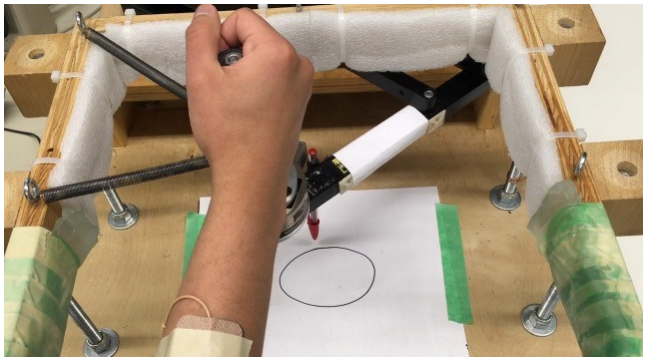

Figure 4-10 Figure showing the reproduction phase. The patient interacts with the robot to execute the given task, in the therapist's absence, the robot assists the patient to complete and perform the task just as the therapist did.

4.6 Results

In this section, we present the obtained results of the experiments. The way they are presented is as follows: First, the patient's task performance assisted by the robot is compared against the therapist's demonstration of the task through their single-side

amplitude spectra and the robot's end-effector trajectories. As a numerical analysis, the correlation, as well as the root-mean-square error (RMSE), are computed to show the similarity between the therapist's performance and the patient's performance (in terms of the robot end-effector trajectory).

4.6.1 First Scenario

In the first scenario, we analyze and compare the therapist's behavior against the patient performance in the reproduction phase with and without robotic assistance. Figure 4-11 shows the robot's end effector trajectory along the X-axis, the orange and dashed line represents the therapist's demonstration, the blue and solid line represents the patient's reproduction with robotic assistance, while the black and solid line represents the patient's performance without assisted by the robot. As can be seen, the system created a general model of the signal using the FS algorithms previously described. Based on the visual analysis, the assisted reproduction looks similar to the therapist's demonstration; on the other hand, the non-assisted reproduction does not match the therapist's demonstration at all.

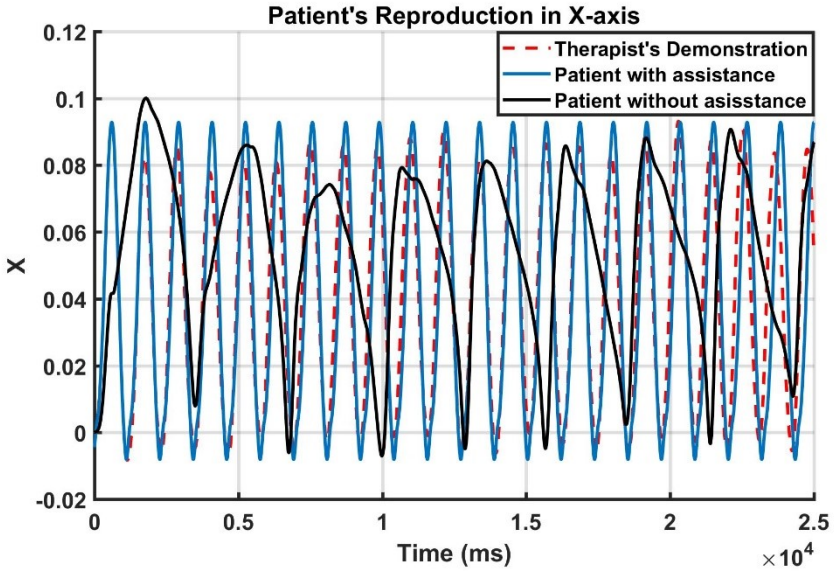

Figure 4-11 Therapist's demonstration and patient's reproduction with and without robotic assistance in X-axis.

The next analysis involves the cross-correlation between the patient's reproduction with and without robotic assistance and the therapist's demonstration. As can be seen in Figure 4-12, both signals present a similarity, but due to the results shown by the assisted reproduction, we can conclude that it is more similar to the therapist's demonstration than the non-assisted reproduction. This is easy to demonstrate, the patient's reproduction without assistance result does not show as much correlation as the assisted reproduction.

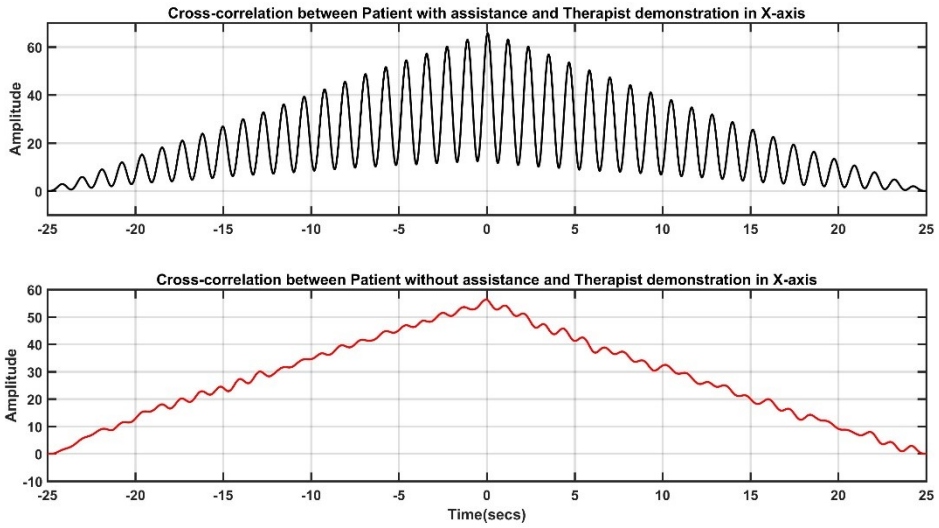

Figure 4-12 Cross-correlation between the patient's reproduction with and without robotic assistance and the therapist's demonstration in X-axis.

The last visual analysis runs in the frequency domain. Figure 4-13 shows the power spectrum of the patient's reproduction with and without robotic assistance respectively. In this figure, the power spectrum of the therapist's demonstration is shown to compare the results. As can be seen, the assisted reproduction shows better and more accurate result, while the non-assisted reproduction does not share similarities in the frequency domain.

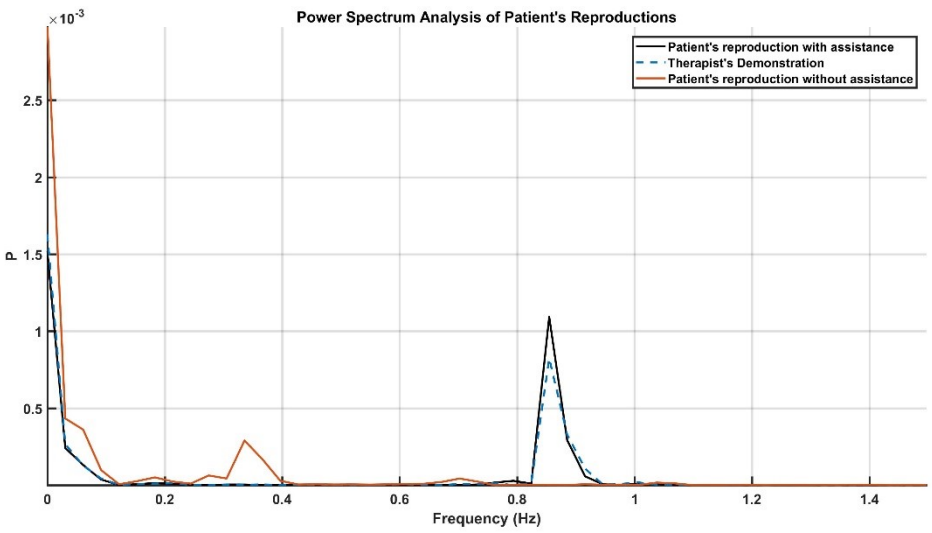

Figure 4-13 Power Spectrum analysis of patient's reproduction with and without robotic assistance and therapist's demonstration.

Finally, Table 4-1 shows the obtained results of the correlation coefficients (CC), mean square error (MSE), and Euclidian distance (ED), analysis.

Table 4-1 Numerical analysis of 1D periodic motion

Method	Patient's Reproduction Analysis	
	Assisted-X	Non-assisted-X
CC	0.9126	0.0577
MSE	0.0002	0.0756
ED	2.2567	6.7547

The obtained results do not differ from the previous visual analysis. They help us to confirm that the patient's reproduction with robotic assistance is much better than the non-assisted reproduction. If the reader has interest on how to analyze these numerical results, Appendix C presents a brief introduction to the methodology used.

4.6.2 Second Scenario

The first analysis for the second scenario shows the patient's reproduction with and without assistance and the therapist's demonstration in X-axis and Y-axis in Figure 4-14 and Figure 4-15 respectively. As can be seen in both figures, the assisted reproduction follows almost perfectly the therapist's demonstration for almost 15 seconds. Due to the methodology used to learn the therapist's demonstration, the system uses fix FSC that do not allow adaptability to different inputs. In other words, the system works in open loop. Notice that the non-assisted patient's reproduction does not match the demonstrated behavior at all.

Figure 4-16 shows the 2D patient's reproduction with and without assistance and the therapist's demonstration. The visual results confirm that the assisted reproduction is better than the non-assisted reproduction.

Notice that as in the previous scenario, the system uses fix values and works in an open loop, therefore improvement is needed to make it more real and natural.

Cross-Correlation analysis of the patient's reproduction with and without robotic assistance and the therapist's demonstration in X-axis and Y-axis is presented in Figure 4-17 and Figure 4-18 respectively. Based on the visual analysis, we can say that in general, the assisted reproduction presents more similarities with the therapist's demonstration than the non-assisted reproduction. An interesting result emerges in the non-assisted reproduction in Y-axis; it shows a similar result as the assisted reproduction. Based on this, we can conclude that the patient's reproduction without assistance got a better result following the demonstration along the Y-axis but not along the X-axis. For this reason, the assisted reproduction presents a better result.

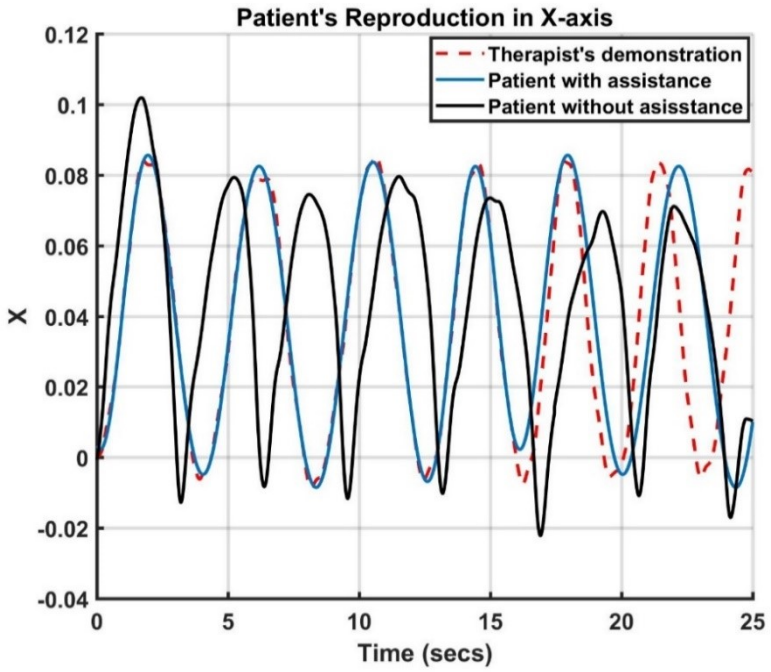

Figure 4-14 Therapist's demonstration and patient's reproduction with and without robotic assistance in X-axis

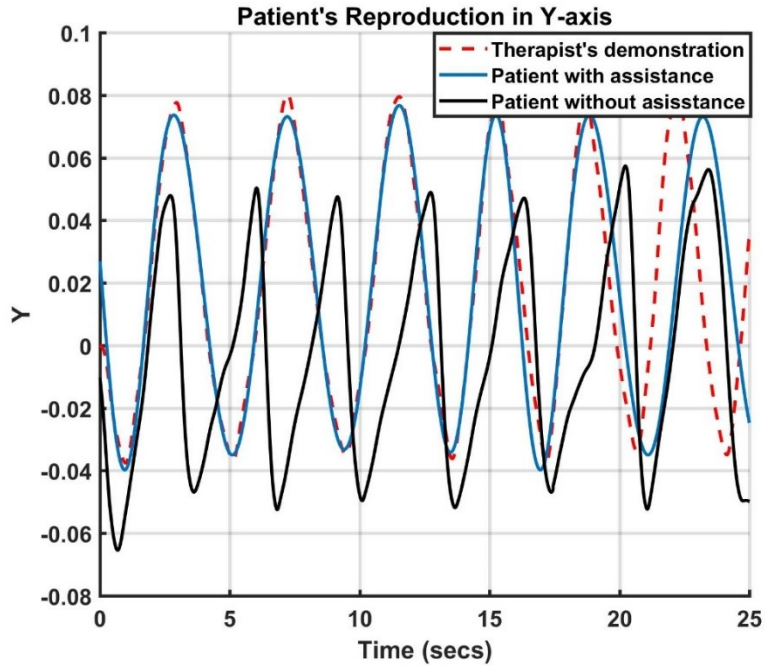

Figure 4-15 Therapist's demonstration and patient's reproduction with and without robotic assistance in Y-axis

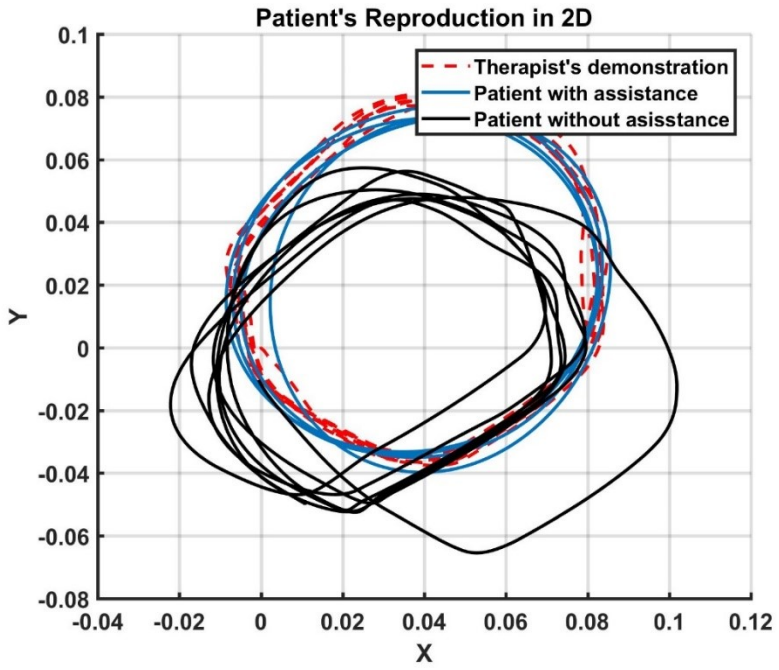

Figure 4-16 Therapist's demonstration and patient's reproduction with and without robotic assistance in 2D.

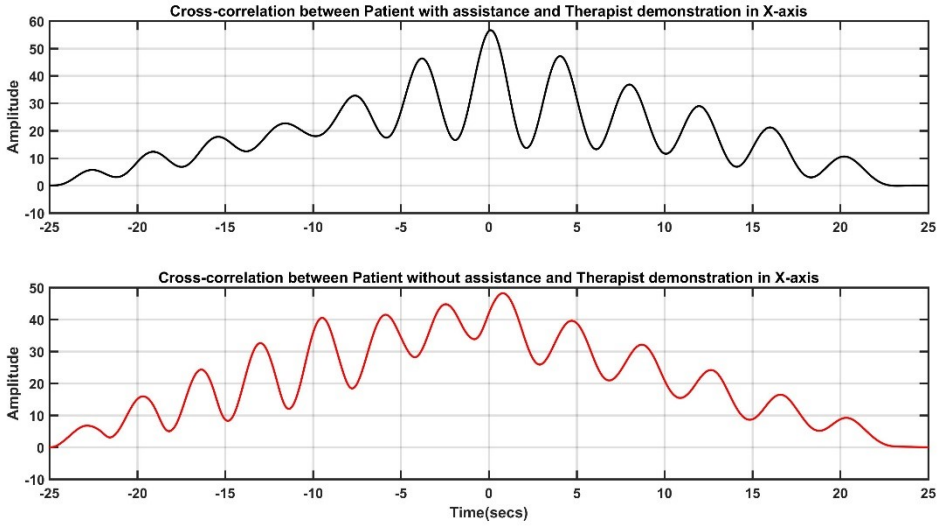

Figure 4-17 Cross-correlation between the patient's reproduction with and without robotic assistance and the therapist's demonstration in X-axis.

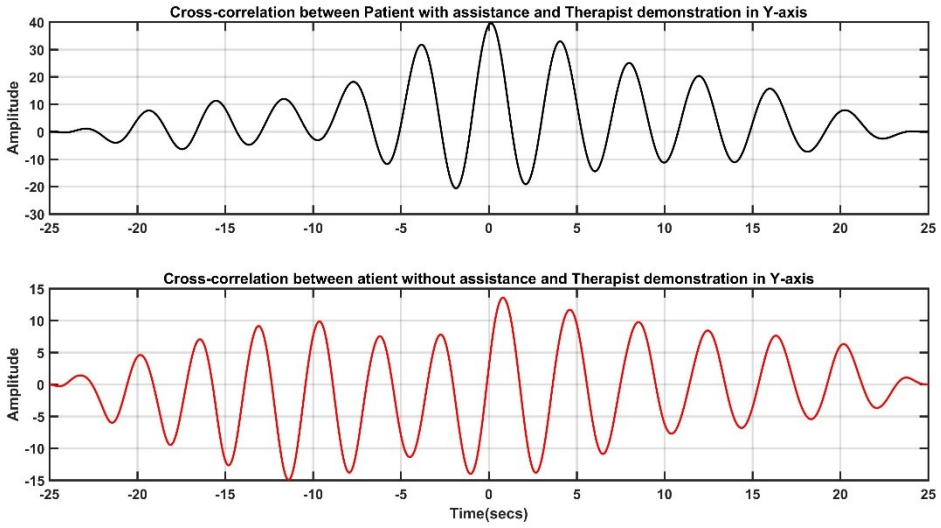

Figure 4-18 Cross-correlation between the patient's reproduction with and without robotic assistance and the therapist's demonstration in Y-axis.

The next step in the analysis uses the power spectrum of the patient's reproduction with and without robotic assistance and the therapist's demonstration. Figure 4-19 shows the patient's reproduction with and without assistance in X-axis. While Figure 4-20 shows the patient's reproduction with and without assistance in Y-axis. One more time, and as in the previous scenario, the power spectrum of the therapist's demonstration is shown to compare the results.

The obtained results show that the assisted and non-assisted reproduction for X-axis and Y-axis share similarities with the therapist's demonstration. Both reproductions are close to the main frequency presented during the demonstration. As a general result, we can say that the assisted reproduction is slightly better than the non-assisted reproduction, but a final statement cannot be taken based on this analysis.

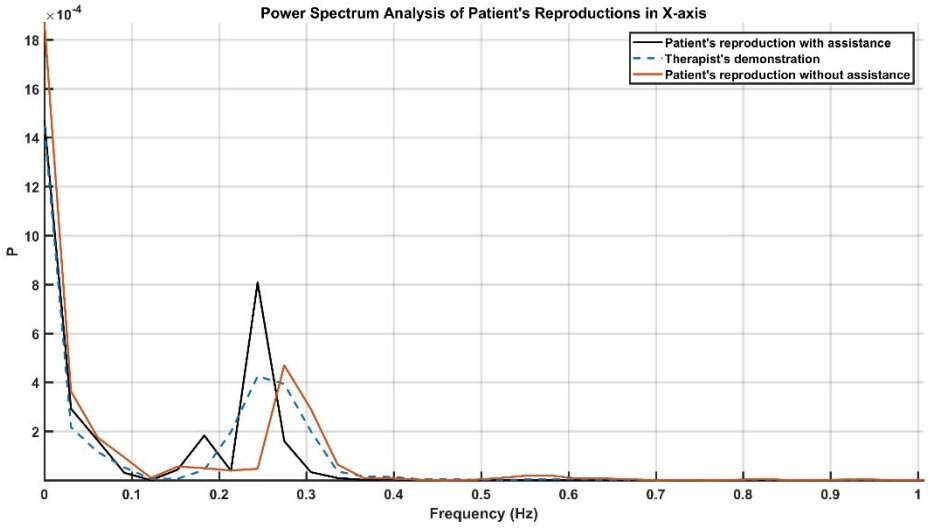

Figure 4-19 Power Spectrum analysis of patient's reproduction with and without robotic assistance and therapist's demonstration in X-axis.

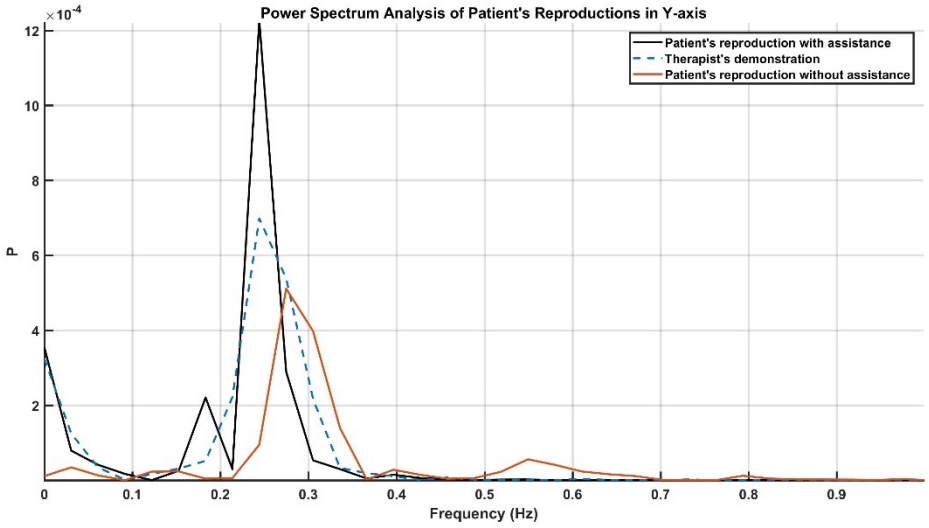

Figure 4-20 Power Spectrum analysis of patient's reproduction with and without robotic assistance and therapist's demonstration in Y-axis.

One more time, and as in the previous scenario, the obtained results of the correlation coefficients (CC), mean square error (MSE), and Euclidian distance (ED) are presented in Table 4-2.

As can be seen, some results (like MSE) present a similar conclusion for the assisted and non-assisted reproduction, while other results (like CC) give a clearer idea about the best reproduction. By taking all the numerical results, we can conclude that the patient's reproduction with robotic assistance is better than the non-assisted reproduction.

Table 4-2 Numerical analysis of 2D periodic motion

Method	Patient's Reproduction Analysis			
	Assisted-X	Assisted-Y	Non-assisted-X	Non-assisted-Y
CC	0.7877	0.8229	0.0306	0.1259
MSE	0.0004	0.0005	0.0019	0.0028
ED	3.2729	3.6197	6.8751	8.2964

4.6.3 Third Scenario

Finally, the obtained results of the third scenario are shown and discussed using a similar methodology as the one presented for the analysis of the previous scenarios. First, we compare the robot's end effector trajectory of the therapist's demonstration, against the patient's reproduction with and without assistance of the robot. Figure 4-21 and Figure 4-22 compare the demonstration and reproductions along the X-axis and Y-axis. As can be seen in both graphs, the therapist's demonstration and the patient's reproduction assisted by the robot have some mismatch for a couple of seconds, also, notice that the

frequency of the patient's reproduction assisted by the robot has some changes during the reproduction. As mentioned before, the controller allows the patient to have freedom along the normal; for this reason, the patient's reproduction assisted by the robot has a small mismatch. Notice that despite the mismatch between the therapist's demonstration and the patient's reproduction assisted by the robot during the first seconds, the system adjusted the parameters and assisted the patient to reproduce the demonstrated behavior. These adjustments are only possible due to the machine learning algorithms.

Figure 4-23 shows the robot's end-effector motion in the 2D space. The results show that the original therapist's demonstration and the patient's reproduction assisted by the robot are very similar. On the other hand, the patient's reproduction without assistance does not match the demonstrated behavior, and it can be taken as a failed attempt. Based on these results, we can conclude that FS + GMM-GMR algorithms can learn and reproduce the therapist's behavior accurately.

Figure 4-24 and Figure 4-25 show the cross-correlation results of the patient's reproduction with and without robotic assistance and the therapist's demonstration. As can be seen, the results show that the assisted reproduction has similarities with the therapist's demonstration in X-axis and Y-axis. The cross-correlation results are totally expected and corroborate the previous obtained results. The plots show that there is a complete correlation between the assisted reproduction and the therapist demonstration. On the other hand, the non-assisted reproduction presents some similarities in X-axis and no similarities in the Y-axis.

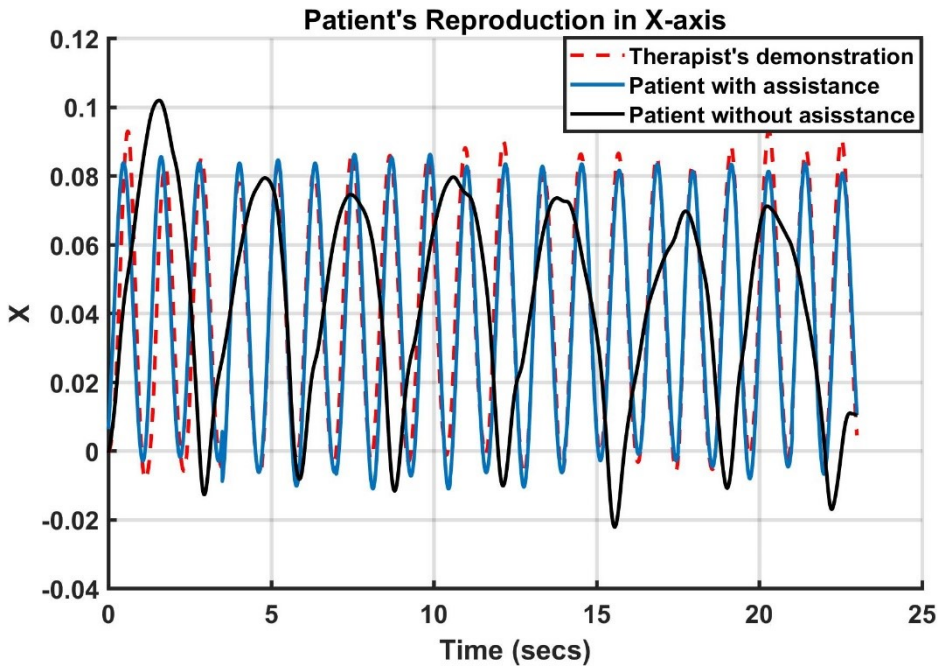

Figure 4-21 Therapist's demonstration and patient's reproduction with and without robotic assistance in X-axis.

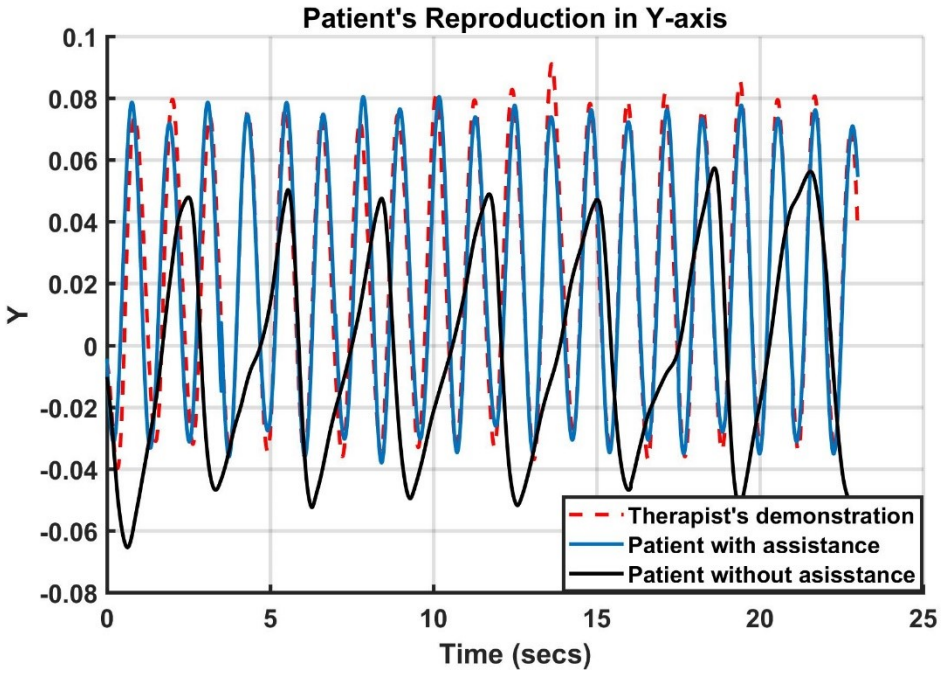

Figure 4-22 Therapist's demonstration and patient's reproduction with and without robotic assistance in Y-axis.

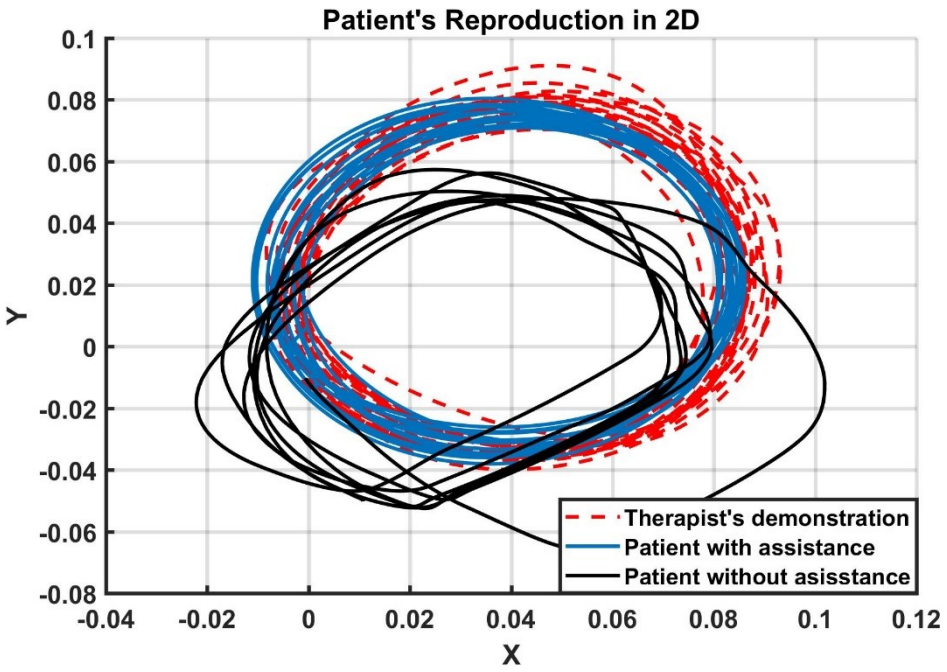

Figure 4-23 Therapist's demonstration and patient's reproduction with and without robotic assistance in 2D.

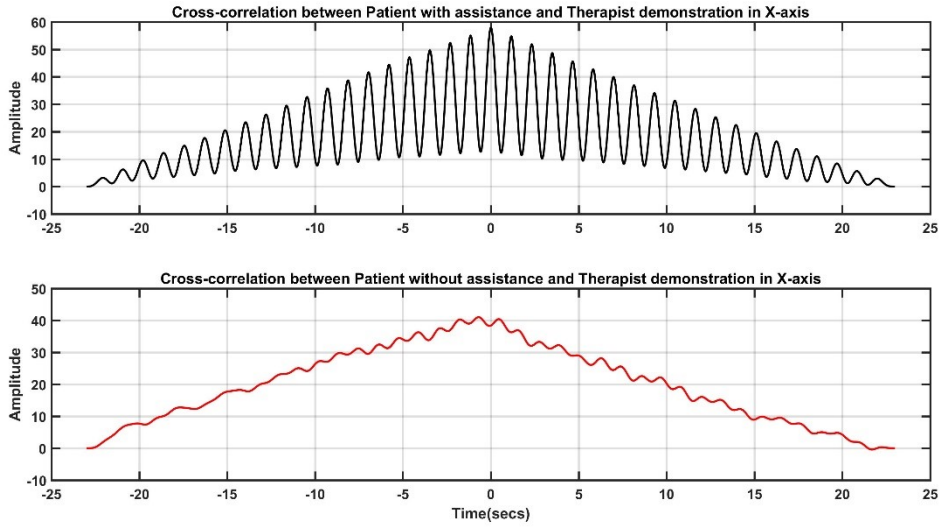

Figure 4-24 Cross-correlation between the patient's reproduction with and without robotic assistance and the therapist's demonstration in X-axis.

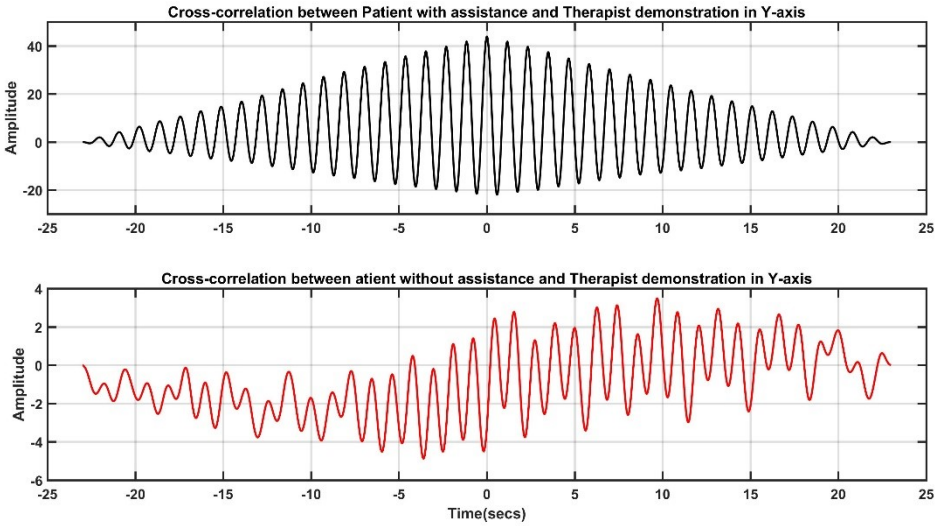

Figure 4-25 Cross-correlation between the patient's reproduction with and without robotic assistance and the therapist's demonstration in Y-axis.

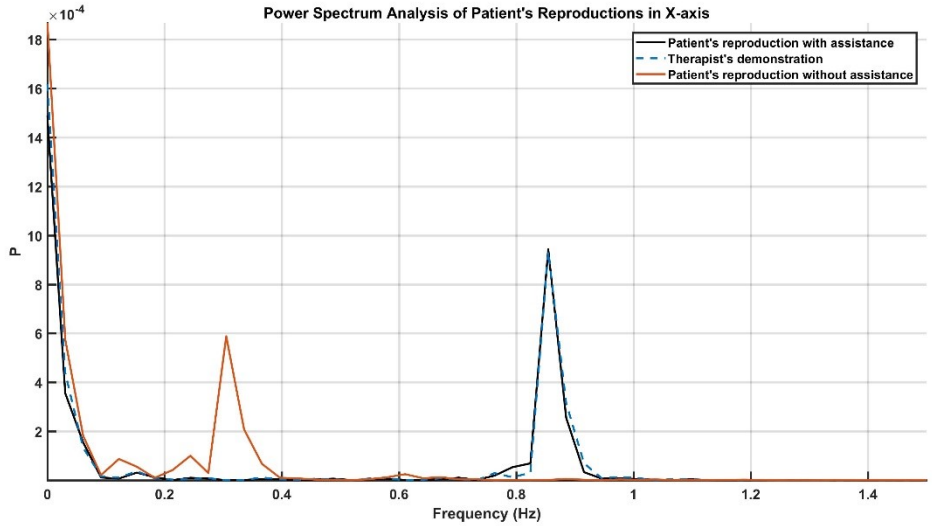

Figure 4-26 Power Spectrum analysis of patient's reproduction with and without robotic assistance and therapist's demonstration in X-axis.

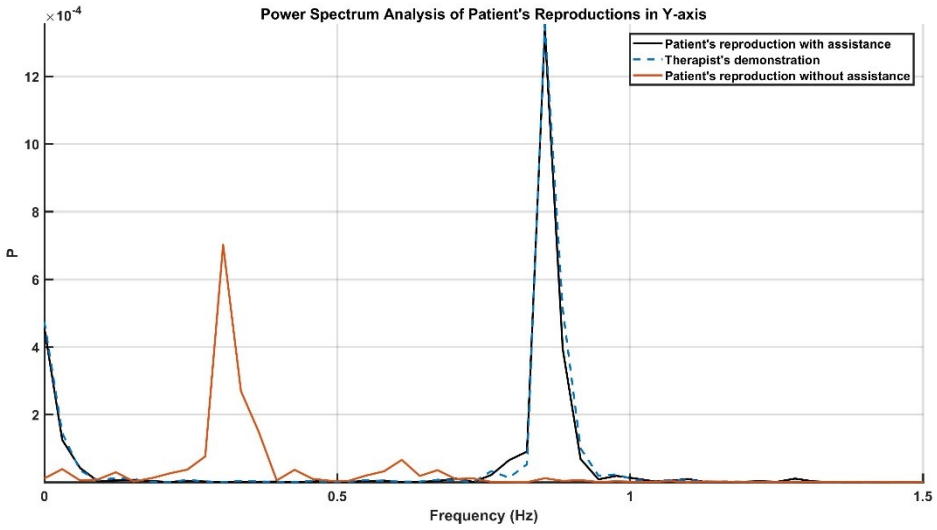

Figure 4-27 Power Spectrum analysis of patient's reproduction with and without robotic assistance and therapist's demonstration in Y-axis.

Finally, Figure 4-26 shows the patient's reproduction with and without assistance in X-axis, while Figure 4-27 shows the patient's reproduction with and without assistance in Y-axis.

The power spectrum of the therapist's demonstration is shown to compare the results. By looking at the obtained results, we can easily conclude that the patient's reproduction without assistance does not match the demonstrated frequency at all. For this reason, and based on this analysis, the assisted reproduction is better than the non-assisted reproduction. The assisted reproduction shows that it has almost the same frequency and amplitude as the demonstrated behavior by the therapist. Based on these results, we can conclude that, as in the previous analysis, the assisted reproduction presents better results than the non-assisted reproduction.

Correlation coefficients (CC), mean square error (MSE), and Euclidian distance (ED) are used to show a numerical analysis of the reproductions. These results can be found in Table 4-3.

Table 4-3 Numerical analysis of 2D periodic motion with GMM and GMR algorithms

Method	Patient's Reproduction Analysis			
	Assisted-X	Assisted-Y	Non-assisted-X	Non-assisted-Y
CC	0.9526	0.9541	-0.0743	-0.0539
MSE	0.0001	0.0001	0.0021	0.0035
ED	1.5112	1.7967	6.9389	8.9547

Using the numerical results presented in Table 4-3, we can easily conclude that the assisted reproduction is better than the non-assisted reproduction.

Notice that the numerical results presented in scenario 3 are better than the results obtained in scenario 2. The reason is that the second scenario built a general and fixed model of the demonstration and it operates in an open loop. For this reason, the second scenario shows worse results. The methodology used in scenario two cannot modify its frequency nor amplitude based on the patient's behavior. In other words, the second scenario does not allow any freedom to the patient while the third scenario can help the patient to adjust his/her frequency and amplitude in real time to match the demonstration. This adaptability concept helps the patient to have control and freedom over the performance of the task.

4.7 Conclusion and summary

In this chapter, an LfD technique was applied to learn and imitate a semi-periodic trajectory following a task performed through a robotic rehabilitation system. The demonstration and imitation phases of LfD were based upon GMM, and GMR approaches, respectively. The goal was to replicate the therapist's behavior accurately. In scenarios 1 and 2, we demonstrated the effectiveness of the modeled system through an

FS. The results showed an acceptable reproduction of the therapist's behavior but, due to the systems nature, the results show the necessity of improvement. In scenario 3, polar coordinates of the robot's end-effector, as well as the FSC along each axis in the 2D space were recorded in the demonstration phase. A significant improvement arose due to the LfD. Thanks to the algorithms, dynamic frequencies and amplitudes of the FS are possible to be reproduced. As seen in the results, better and more accurate reproduction is possible. In future works, we would like to extend this algorithm to more complex tasks using different LfD algorithms and a 3D space.

5 LEARNING AND ROBOTIC IMITATION OF THERAPIST'S MOTION AND FORCE

5.1 Introduction

Strokes have become one of the most common causes of death and disability worldwide (Balasubramanian, 2010). In recent years, and due to the stroke incidents, the number of people with disabilities has increased and reached new records. Usually, after a stroke episode, the affected person suffers a loss of mobility in all or parts of the body; due to this disability, the person is not able to perform basic daily living activities.

The most common way to help the people with disabilities is through therapy exercises that engage patients in repetitive tasks (Lum, 2002). These therapy exercises are usually given in a methodology called traditional therapy, where the therapist interacts with the patient during a therapy session. The therapist's role in the therapy is to set the exercises based on the patient's needs, to interact and help the patient, and to assess the patient's performance (Hammond, 2004).

Due to the increase of stroke episodes, and the limited health care resources, the demand for therapy services has become a growing problem around the world. One of the most common problems is the lack of therapists' availability. During a therapy session, a therapist has to be present during the session; given that there is a limited number of therapists, the recovery time and number of therapy sessions get affected.

Due to their ability to execute programmed tasks without fatigue and with high accuracy, an interesting solution to solve these problems emerged from the robotics field. The

combination of robotics and human rehabilitation started after the second half of the 20th century (Krebs, 2004), however, it was not until 1989 when the Massachusetts Institute of Technology (MIT) developed the first non-industrial rehabilitation robot (Hogan, 1992). After the introduction of this robot into the rehabilitation field, the development of new robotic systems and techniques were implemented (Krebs, 2004).

Currently, robotic rehabilitation is divided into two different categories, the first one is known as assistive therapy; it uses haptic devices to assist the patient in completing a given task. The second category is known as resistive therapy; the haptic device used in this category opposes the patient's actions by applying resistive forces. Resistive therapy is commonly used to build muscle strength.

Every patient has a different level of disability and different needs, therefore, a personalized therapy focused on every patient is needed. Unfortunately, most of the developed rehabilitation robotic systems were designed for executing predefined and preprogrammed tasks. To take advantage of the robots' ability of adaptability and reprogramming, mathematical, robotics and programming knowledge are needed. However, most of the therapists do not have the required knowledge to modify the preprogrammed rehabilitation robots. This limits the use of these systems in actual clinical settings.

In this chapter, we propose the use of a technology called Learning from Demonstration (LfD) to help the therapists to reprogram the robots without any knowledge other than their therapeutic background. The proposed LfD algorithm is known as Stable Estimator of Dynamical Systems (SEDS). This algorithm is used to learn the therapist's behavior through kinesthetic teaching during the demonstration phase (Lee, 2012). Later, when the therapist is no longer present to interact with the patient, the robotic system reproduces and imitates the therapist's behavior during a step known as the reproduction phase.

This chapter is organized as follows. Section 5.2 presents the related works. A description of the tasks is presented in Section 5.3. In Section 5.4, we explain the LfD algorithm used

in this chapter. Section 5.5 presents two experiments developed to test the system. Finally, results, conclusions, and future work are covered in Sections 5.6 and 5.7, respectively.

In this chapter, therapists and patients were *simulated*. Along this thesis, the word *therapist* refers to a person taking the role of the therapist. Notice that this person does not have therapeutic skills. The word *patient* refers to a non-disabled person that took the role of a person with disabilities. Different devices such as springs and transcutaneous electric nerve stimulator (TENS) were used to simulate the motor impairments.

5.2 Related Work

Our research group has been working and implementing LfD technologies that leverage the learning and imitation features offered by LfD to save the therapist's time. Some examples can be found in the following works: In (Tao, 2014), a haptic teleoperation system was developed by the authors; the idea is to use this system for home-based therapy. This system employed impedance-based learning of the therapist's behavior. The learned behavior is later used in the reproduction phase to imitate the therapist's behavior. In Chapter 3, we developed a telerehabilitation cooperative task. By using Gaussian Mixture Model (GMM) and Gaussian Mixture Regression (GMR), we were able to enable learning and imitation of the therapist's behavior.

In (Maaref, 2016) a combination of robotic rehabilitation system, LfD, and Assist-as-Needed (AAN) was implemented. The system learns the therapist's impedance using GMM. Later, a model of the therapist's behavior is built using GMR. The obtained model is used in combination with AAN. The AAN computes the error between the patient's current performance and the behavior obtained from the GMR and, based on this error; it determines whether to assist the patient or not.

In this research, we use a similar approach as in the presented implementations. Some improvements are presented by using the well-developed SEDS algorithm. This algorithm uses GMM to build a model in the form of a nonlinear dynamic system capturing the demonstrated behavior. SEDS uses conditions to ensure global asymptotic stability (GSA) of the system. The reproduction phase is run through a combination of

the classical GMR and the SEDS algorithm. While GMR by its own cannot ensure GAS, SEDS does so.

It is clear that the main advantage of SEDS is the fact that it ensures GAS of the system, even for a small number of demonstrations, the system is capable of converging to the desired value. A more reliable and robust system is implemented by using this algorithm.

5.3 Task Description

In this chapter, we propose two different tasks to show the LfD capabilities in the robotic rehabilitation field. The first task is position control, while the second task is a force control task. The proposed system can perform the task without any previous knowledge of the task nor task environment. As mentioned before, due to SEDS' features, a small number of demonstrations are needed to learn the therapist's behavior.

Both tasks are divided into two different phases. The first phase is known as the demonstration phase, where the therapist executes the task a couple of times. During this phase, the system builds a model of the demonstrated behavior. The second phase is known as the reproduction phase. In this phase, the therapist is no longer involved in the therapy; the robotic system takes the therapist's role and reproduces the therapist behavior by using the SEDS algorithm.

There are two ways to implement the therapy. Figure 4-7 shows the parallel mode while Figure 4-8 shows the sequential mode. In the sequential mode, first, the therapist interacts with the robot to train the system, later, in the absence of the therapist and during the reproduction phase, the patient interacts with the robot. The parallel mode is similar to the sequential mode; the difference emerges during the training phase; the therapist and the patient work at the same time with the robot. Even though the system can execute both training modes, we decided to use the sequential mode for convenience and to show that the algorithms used in this thesis can be trained using either parallel or sequential mode.

5.3.1 Task 1

For the first task, a point-to-point reaching task was implemented, where from a random initial point, the patient must reach a random target point following the desired trajectory. This task involves 2-DOF in a planar space where the goal is to target shoulder-and-elbow therapy.

During the demonstration phase, the therapist is asked to execute the task from two different initial points to a fixed target. The goal of the demonstration phase is to show the system how to approach any given target. Afterward, the system learns the therapist's reaching behavior and builds a model. Later, during the reproduction phase, the system can execute the task from any random initial point to any random target. Due to the SEDS nature, the system generalizes the demonstrated motion and can estimate trajectories even for non-demonstrated values.

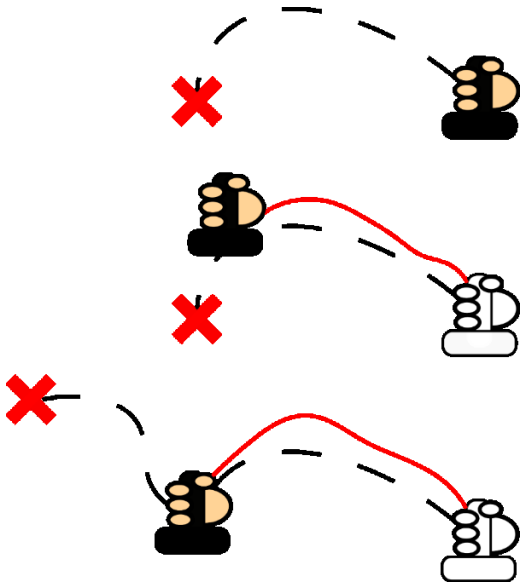

Figure 5-1 Task 1 graphical description.

Notice that the system can follow desired trajectories by creating a new target after reaching the original target, in other words, the patient would be able to move from an initial point to a target, once the patient reaches the target, that position is taken as the

new initial point and a new target is set. In this way, complex trajectories can be segmented into smaller pieces. In this task, the patient is asked to move the robot's end-effector from a fixed and predetermined initial point to different targets by following the demonstrated trajectory. Details about this methodology will be discussed in Section 5.5. A schematic model of this idea is showed in Figure 5-1.

5.3.2 Task 2

The second task involves a force tracking motion in a 2-DOF planar space used for shoulder-and-elbow therapy. The idea is to attach the robot's end-effector to an array of springs that combined, keeps the robot's end-effector in a passive equilibrium point. During the demonstration phase, the therapist is asked to move the robot from an initial point to the passive equilibrium point following a desired and semi-constant velocity; this can be translated in a non-linear applied force along X and Y axis. Due to the nature of the springs' array, at the beginning of the demonstrations, some springs are compressed while others are extended. The system captures the interaction forces between the patient and the spring's array, the summation of forces along each axis is computed to create two virtual springs with dynamic stiffness as shown in Figure 5-2. As in the previous task, SEDS generalizes the recorded values and creates a model of the therapist's behavior. During the reproduction phase, the patient is asked to move the robot's end-effector from a random and non-demonstrated initial point to the passive equilibrium point of the system following the desired force demonstrated by the therapist. A diagram of this task can be found in Figure 5-3.

5.4 Learning From Demonstration Algorithms

In this section, an introduction to the SEDS and GMM algorithms is given. A deeper explanation of these algorithms can be consulted in (Calinon, 2009), (Billard, 2011).

The first algorithm to be covered in this section is GMM; it is a probability density function commonly used to cluster data; it has been widely implemented in the LfD field to encode spatial and temporal components of continuous trajectories and behaviors.

GMM relies on a weighted sum of Gaussian component densities, where each Gaussian component is represented by two values, a mean and a covariance matrix.

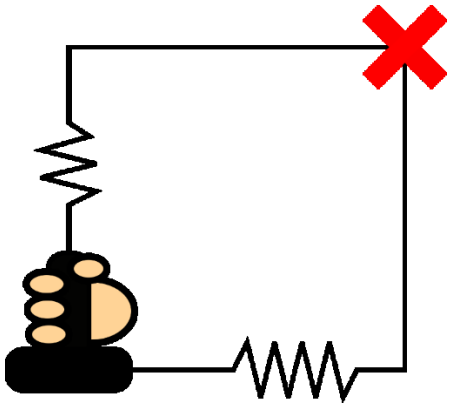

Figure 5-2 Task 2 virtual springs along X-axis and Y-axis.

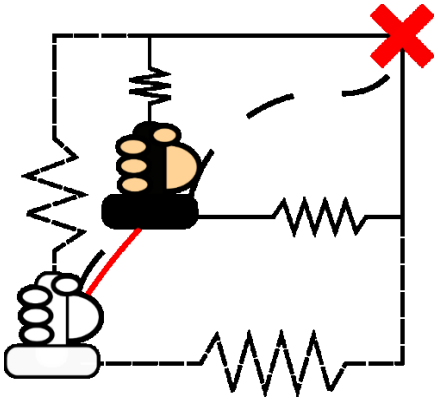

Figure 5-3 Task 2 graphical description.

Due to its nature, the SEDS algorithm requires a nonlinear autonomous dynamical system model, for this reason, the behavior to be learned has to follow this characteristic. In this chapter, depending on the desired task, we take the patient's position, which is the same as the robot's end-effector position, or the human/robot interaction force as the input of the dynamical system, and the velocity or the first derivative of the force as the output. In both tasks, the system is defined as

$$\dot{\xi} = f(\xi) \tag{5-1}$$

where *f* is defined as a nonlinear function with a single equilibrium point and continuous for all time *t*. Depending on the task, ξ is the position or force of the robot's end-effector in Cartesian space. Notice that given any initial condition, the motion or force evolves according to the dynamics of $\dot{\xi}$.

The estimated model is given by

$$\dot{\xi} = \hat{f}\,(\xi) \tag{5-2}$$

Where *f* describes the actual dynamics of motion or force in the task demonstrated by the therapist, $\hat{f}$ is a function that estimate it by the set of parameters $\{\pi^K, \mu^K, \Sigma^K\} = \theta^K = \boldsymbol{\theta}$. Where, μ is the mean of the Gaussian model, Σ is the covariance matrix, π is the prior, and *K* is the index of the Gaussian model. The optimal values of these parameters are computed based on the set of demonstrations.

μ^K and Σ^K represent each Gaussian distribution *K* and defined by

$$\mu^K = \begin{pmatrix} \mu_{\xi}^K \\ \mu_{\dot{\xi}}^K \end{pmatrix} \quad \Sigma^K = \begin{pmatrix} \Sigma_{\xi}^K & \Sigma_{\xi\dot{\xi}}^K \\ \Sigma_{\dot{\xi}\xi}^K & \Sigma_{\dot{\xi}}^K \end{pmatrix} \tag{5-3}$$

Each point $\left\{\xi^{t,n}, \dot{\xi}^{t,n}\right\}_{t=0,n=1}^{T^n,N}$ where $(.)^{t,n}$ is the *t*-th data point of the *n*-th demonstration in the *N* demonstrations of the recorded data is linked to a probability density function given by

$$P\left(\xi^{t,n}, \dot{\xi}^{t,n}; \boldsymbol{\theta}\right) = \sum_{k=1}^{K} P(k) P\left(\xi^{t,n}, \dot{\xi}^{t,n} | k\right) \begin{cases} \forall_n \in 1..N \\ t \in 0..T^n \end{cases} \tag{5-4}$$

where $P(k)$ is the prior of the Gaussian distribution k , *T* is the total number of training data points, and $P\left(\xi^{t,n}, \dot{\xi}^{t,n} | k\right)$ is the conditional probability density function given by

$$P\left(\xi^{t,n}, \dot{\xi}^{t,n}|k\right) = \mathcal{N}\left(\xi^{t,n}, \dot{\xi}^{t,n}; \mu^K, \Sigma^K\right)$$
$$= \frac{1}{\sqrt{(2\pi)^{2d}|\Sigma^k|}} e^{-\frac{1}{2}\left([\xi^{t,n}, \dot{\xi}^{t,n}]-\mu^K\right)^T (\Sigma^k)^{-1} \left([\xi^{t,n}, \dot{\xi}^{t,n}]-\mu^K\right)} \quad (5\text{-}5)$$

After taking the posterior mean of $P(\dot{\xi}|\xi)$, the estimated function is given by

$$\dot{\xi} = \sum_{k=1}^{K} \frac{P(k)P(\xi|k)}{\sum_{i=1}^{K} P(i)P(\xi|i)} \left(\mu_{\dot{\xi}}^k + \Sigma_{\dot{\xi}\xi}^k \left(\Sigma_{\xi}^k\right)^{-1} \left(\xi - \mu_{\xi}^k\right)\right) \quad (5\text{-}6)$$

using the simplification:

$$\begin{cases} A^k = \Sigma_{\dot{\xi}\xi}^k \left(\Sigma_{\xi}^k\right)^{-1} \\ b^k = \mu_{\dot{\xi}}^k - A^k \mu_{\xi}^k \\ h^k(\xi) = \dfrac{P(k)P(\xi|k)}{\sum_{i=1}^{K} P(i)P(\xi|i)} \end{cases} \quad (5\text{-}7)$$

And substituting (5-7) into (5-6) we obtain:

$$\dot{\xi} = \hat{f}\,(\xi) = \sum_{k=1}^{K} h^k(\xi)(A^k\xi + b^k) \quad (5\text{-}8)$$

Note that **(5-8)** is a sum of linear dynamical systems that results in a nonlinear function. In this equation, $A^k\xi + b^k$ corresponds to a line with slope A^k and passes through the center of the Gaussians μ^K. The nonlinearity of this equation is given by $h^k(\xi)$; this is a weighting term that gives the influence of each Gaussian in the estimated function.

Given the previous conditions, **(5-8)** is asymptotically stable. To ensure GAS, the system must meet two conditions:

$$\begin{cases} b^k = -A^k\xi^* \\ A^k + (A^k)^T \prec 0 \end{cases} \quad \forall k = 1, \dots, K \quad (5\text{-}9)$$

Here, $\prec 0$ denotes the negative definiteness of a matrix. The proof is shown (Billard, 2011).

Given the conditions and equations to ensure GAS of the function, the next step is to find the parameters of **(5-8)**. SEDS is used to find the optimal values. Mean Square Error (MSE) is an optimization objective function that provides a solution to this problem. This function is combined with SEDS to measure the accuracy of the estimations based on the recorded data. The obtained minimization are the optimal parameters:

$$\min_{\theta} \mathcal{J}(\theta) = \frac{1}{2\mathcal{T}} \sum_{n=1}^{N} \sum_{t=0}^{T^n} \left\| \hat{\dot{\xi}}^{t,n} - \dot{\xi}^{t,n} \right\|^2 \tag{5-10}$$

subject to the following constraints:

$$\begin{cases} b^k = -A^k \xi^* \\ A^k + (A^k)^T \prec 0 \\ \Sigma^k \succ 0 \\ 0 < \pi^K \leq 1 \\ \Sigma_{k=1}^{K} \pi^K = 1 \end{cases} \qquad \forall k = 1, \dots, K \tag{5-11}$$

Note that the first two constraints in **(5-11)** are the previously defined conditions for stability presented in **(5-9)**. MSE can be taken as a non-linear programming problem (Bradley, 1992), and can be solved using Successive Quadratic Programming (SQP) (Wright, 2006).

Once the parameters are computed, the system is ready to be implemented. Figure 5-4 shows the block diagram of the system. Notice that this configuration does not depend on the task nor the environment.

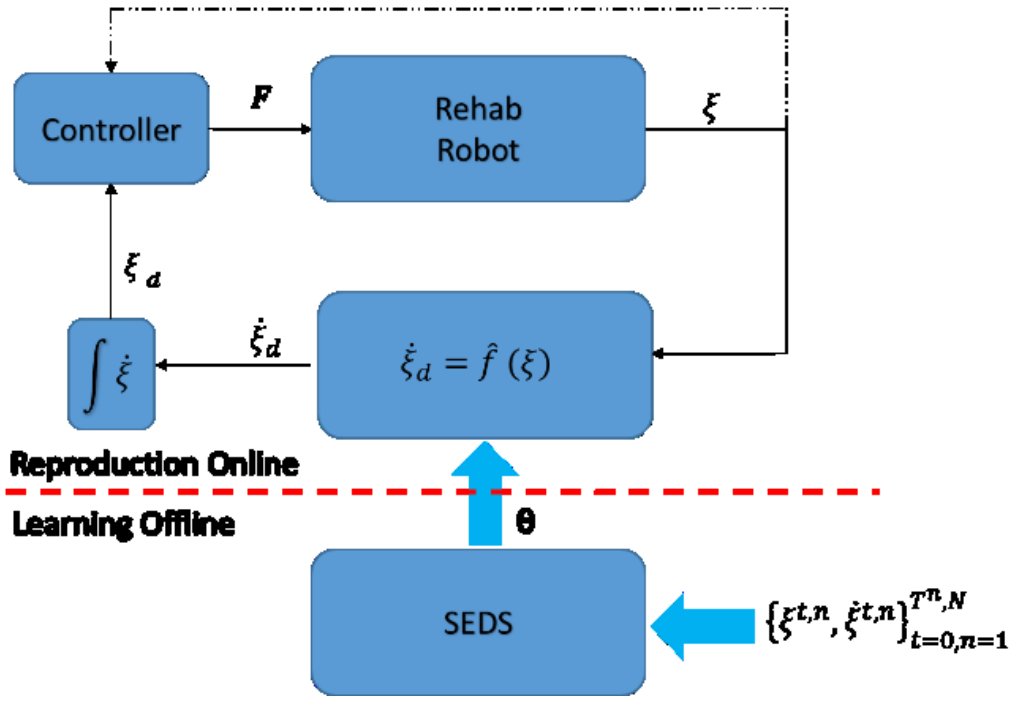

Figure 5-4 Block diagram showing the learning and reproduction phase.

5.5 Experiments

In this section, we used a Rehabilitation Robot (Quanser, Inc. Markham, Canada) (Lu, 2012) and a Gamma force sensor (ATI-IA, NC, USA); Figure 5-5 shows the rehab robot and setup used in this chapter.

5.5.1 Task 1

During the demonstration phase, the therapist executed ten different demonstrations from two different initial points. A model of the demonstrated trajectory was built. As shown in the block diagram in Figure 5-4, a controller and position feedback are needed in this task. A PD position controller is used to ensure the robot follows the desired trajectory in the reproduction phase. In the reproduction phase, the patient is asked to reproduce the task from a non-demonstrated initial point with and without robotic assistance.

Figure 5-5 Rehab Robot and setup used in these experiments.

To measure the accuracy of the system, a sub-task was implemented. This sub-task makes use of one non-demonstrated initial point and one target. In this way, a comparison between the therapist's demonstration and the patient's reproduction with and without robotic assistance can be analyzed more easily. Section 5.6 shows and discusses the obtained results.

5.5.2 Task 2

As mentioned in Section 5.3, the task involves an array of springs like the one shown in Figure 5-6. In this task, the controller shown in the block diagram of Figure 5-4 is equal to one, and the feedback is not needed. In the demonstration phase, the therapist executed five different demonstrations from a given initial point in the 2D space; due to the array and the initial position, the initial forces have similar values. The robot/human interaction force, its first derivative, and time were recorded and used by the learning algorithm. After the demonstrations, the system builds a nonlinear dynamical model capturing the data. Then, during the reproduction phase, the patient is asked to reproduce the task from

a non-demonstrated initial point with and without robotic assistance. Section 5.6 shows and discusses the obtained results.

Figure 5-6 Task 2 setup with the springs' array.

5.6 Results

In this section, we show the obtained results during the reproduction phase. The goal is to compare the patient's behavior with and without robotic assistance against the therapist's demonstration. Different methods such as correlation coefficients (CC), mean square error (MSE), Euclidian norm (ED), mean absolute percentage error (MAPE), and cross-correlation (XC) are used to measure and analyze the results.

5.6.1 Task 1

5.6.1.1 Sub-Task

As mentioned before, a sub-task is used to measure the patient's accuracy in Task 1. Figure 5-7 shows the patient's reproduction from a non-demonstrated initial point with (blue and solid line) and without (black and solid line) robotic assistance and the

therapist's demonstration (red and solid line). As can be seen, SEDS generalized the demonstrated behavior and can assist the patient to reach the desired target following a similar trajectory as the demonstrated by the therapist. On the other hand, the patient without assistance is not able to reproduce the task as demonstrated. The patient reached the target by following a straight line. Based on this visual result, we can conclude that the patient with robotic assistance performed a better reproduction than a patient without reproduction.

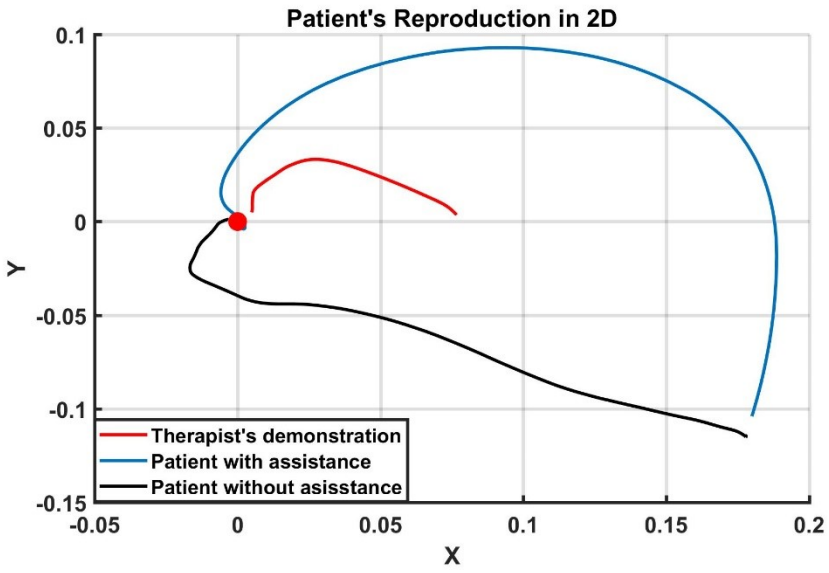

Figure 5-7 Sub-task reproductions.

Following the analysis of the patient's behavior, the cross-correlation between the patient's reproduction with and without assistance and the therapist's demonstration along X-axis and Y-axis are shown in Figure 5-8 and Figure 5-9. As can be seen and according to the cross-correlation results, both reproductions are very similar. This result does not show that the reproductions are totally correlated, they only show a similar level of correlation. This is expected, both reproductions started from a different initial point and does not show a similar path as the demonstrated at all. As a general conclusion, patient's reproduction with robotic assistance is slightly better than the reproduction without assistance.

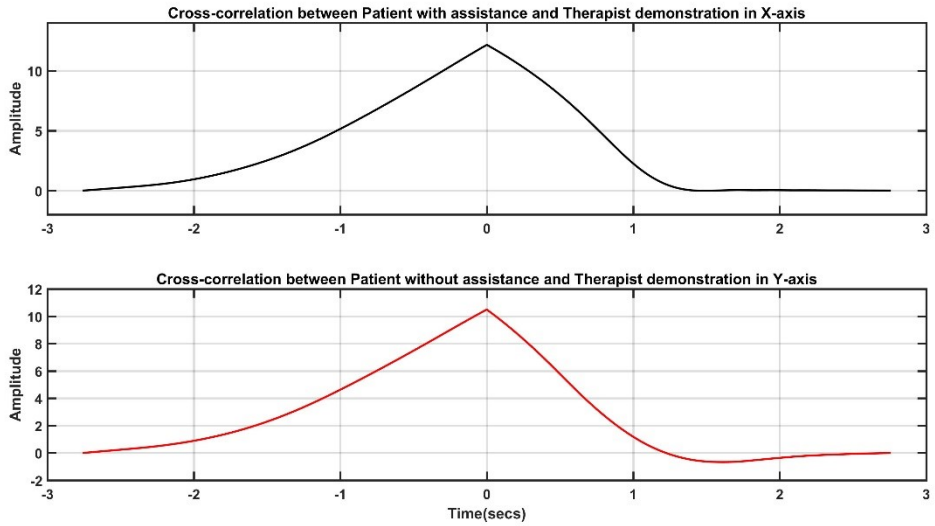

Figure 5-8 Cross-correlation plots of patient's reproduction with assistance (Top) and patient's reproduction without assistance (Bottom) along the X-axis.

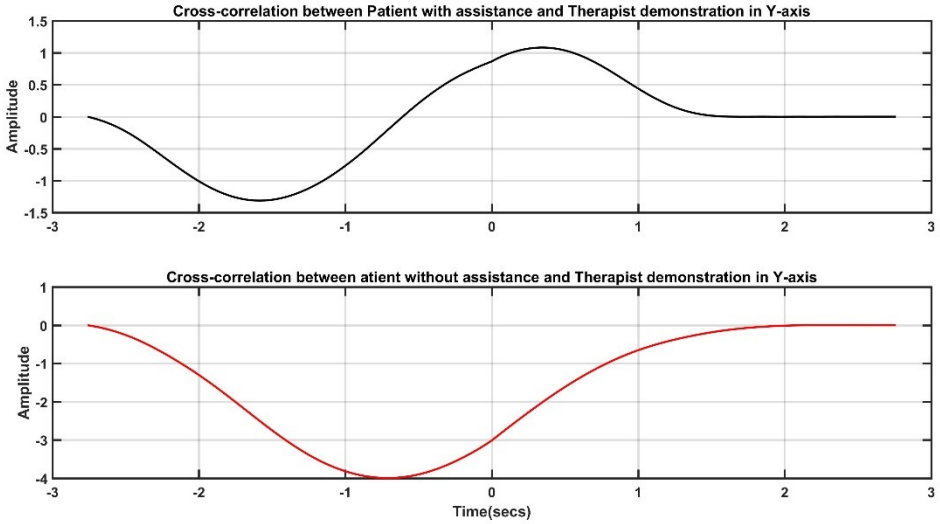

Figure 5-9 Cross-correlation plots of patient's reproduction with assistance (Top) and patient's reproduction without assistance (Bottom) along the Y-axis.

For the next analysis, we analyze the data in the frequency domain by comparing the power spectrum of the patient's reproduction with and without robotic assistance against

the therapist's demonstration. Figure 5-10 shows the patient's reproduction with and without assistance in X-axis. While Figure 5-11 shows the patient's reproduction with and without assistance in Y-axis.

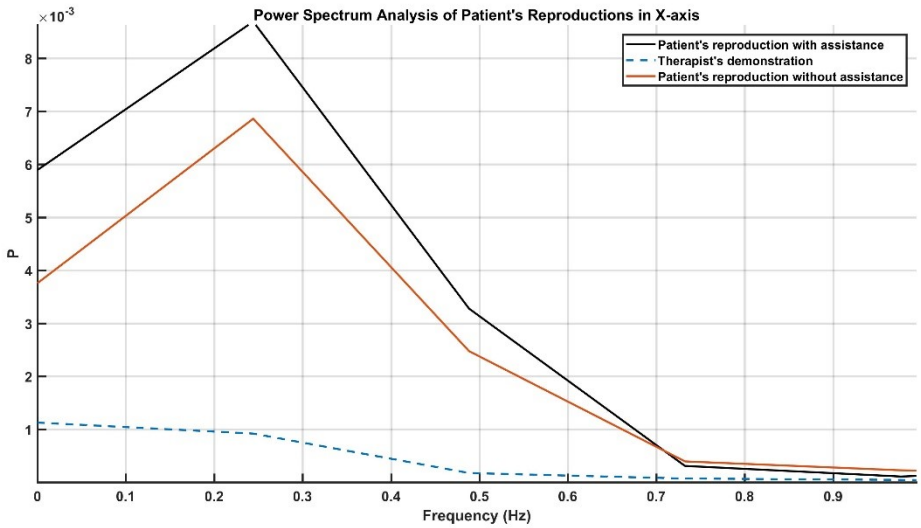

Figure 5-10 Power Spectrum analysis of patient's reproduction with and without robotic assistance and therapist's demonstration in X-axis.

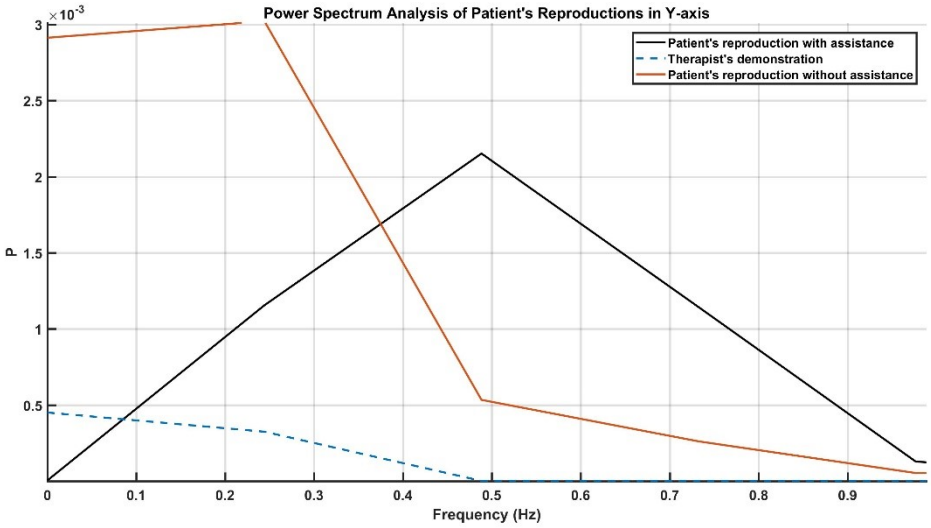

Figure 5-11 Power Spectrum analysis of patient's reproduction with and without robotic assistance and therapist's demonstration in Y-axis.

According to the results, the patient's reproduction without robotic assistance has more similarities with the therapist's demonstration. Given that the patient's reproduction with robotic assistance also shares similarities with the therapist's demonstration, we cannot conclude anything based on this results.

Finally, and to show numerical results, Table 5-1 shows the obtained results of the correlation coefficients (CC), mean square error (MSE), and Euclidian distance (ED) analysis.

Table 5-1 Numerical analysis of position control task using SEDS

Method	Patient's Reproduction Analysis			
	Assisted-X	Assisted-Y	Non-assisted-X	Non-assisted-Y
CC	0.9331	0.7272	0.9632	0.1458
MSE	0.0058	0.0032	0.0038	0.0075
ED	3.9963	2.9502	3.2438	4.5511

The obtained results in this table show that the reproduction along the X-axis is similar in the patient's reproduction with and without robotic behavior. The difference emerges in the Y-axis; the results show that the patient's reproduction with robotic assistance obtained better results than the reproduction without assistance. In general, the patient's reproduction with robotic assistance obtained better results; Y-axis was the key component.

Based on the presented results, we can conclude that SEDS can imitate and generalize a demonstrated behavior and it could be used in future experiments to help patients to reproduce a given task. It can generalize the demonstrated behavior and reproduce it even for non-demonstrated values with accuracy.

5.6.1.2 Complex task

The next step in Task 1 is to implement the complex trajectory by generalizing the learned behavior as mentioned in Section 5.3.

A random and non-demonstrated initial point was chosen. The goal is to reach three different targets (0,0), (0.06,0.06), and (-0.06,-0.06) following the demonstrated therapist's behavior. Figure 5-12 shows the obtained results for the patient's reproduction with (orange and solid line) and without (blue and solid line) robotic assistance.

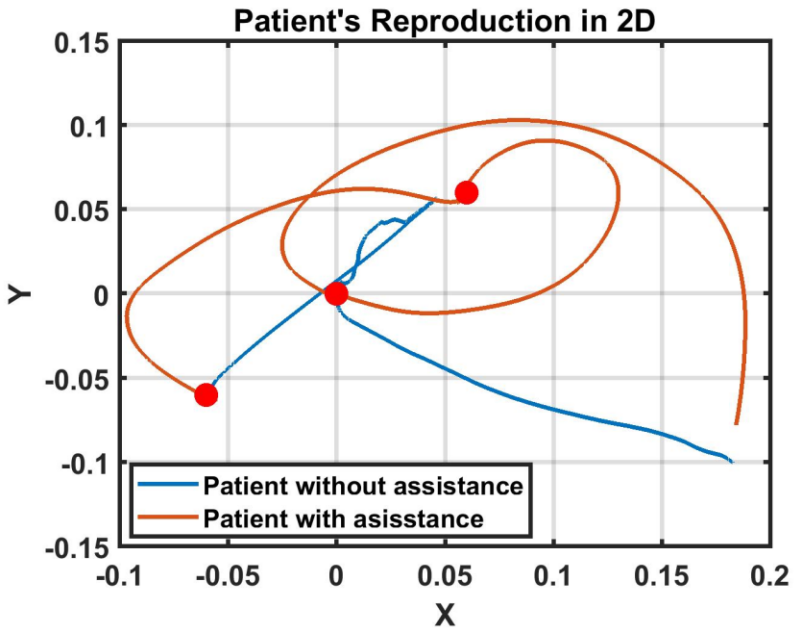

Figure 5-12 Complex task patient's reproduction.

As can be seen, given that the reproduction with assistance reached the desired targets and followed the demonstrated trajectory, the patient's reproduction with robotic assistance obtained better results than the reproduction without assistance. Based on this, we can conclude that SEDS is robust and accurate for reaching motion tasks. It can generalize any task and, due to its nature, it could be used to implement complex trajectories following desired behaviors by segmenting the complex trajectory into smaller trajectories.

5.6.2 Task 2

To analyze Task 2, similar methods to the ones presented in Task 1 analysis are used. Figure 5-13 and Figure 5-14 shows the forces of the patient's reproduction with robotic assistance (blue and solid line), patient's reproduction without assistance (orange and solid line), and therapist's demonstration (yellow and solid line) along the X-axis and Y-axis respectively. In both cases and as mentioned in Section 5.5, the reproductions started from a non-demonstrated initial point.

As can be seen in both results, and given that both reproductions started from the same non-demonstrated initial point, the patient's reproduction without assistance has a significant overshoot at the beginning of the reproduction. On the other hand, the patient's reproduction with robotic assistance presents a better result. The obtained forces are smaller; this means that the reproduction of the task was smoother and closer to the demonstration presented by the therapist. Notice that the assisted reproduction presents a smaller overshoot. The assisted reproduction helps to complete the task and brings the robot's end effector close to the passive equilibrium point; nevertheless, it does not reach the exact equilibrium point, this is due to the implemented controller. This problem can be fixed by using a different gain in the controller. Despite this mismatch, we can conclude that the system works properly and it assists the patient during the reproduction phase.

The next step in the analysis section shows the cross-correlation between the patient's reproduction with and without robotic assistance and the therapist's demonstration. Figure 5-15 shows the results along X-axis, while Figure 5-16 shows the results along Y-axis.

As can be seen in both figures, the assisted reproduction presents better results than the non-assisted. The peaks showed in the result obtained by the assisted reproduction help to determine the correlation and similarity with the therapist's demonstration. Based on this, we can conclude that this shows the accuracy and robustness of SEDS.

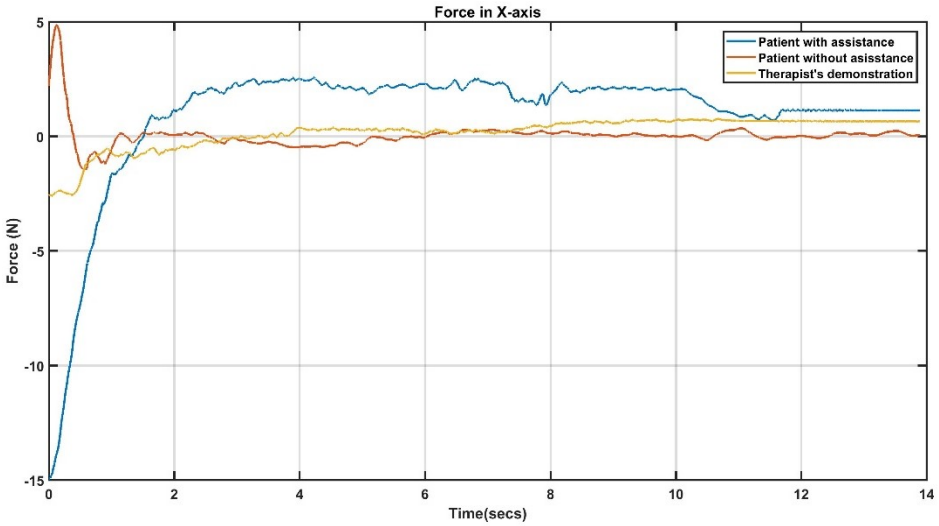

Figure 5-13 Measured forces along X-axis.

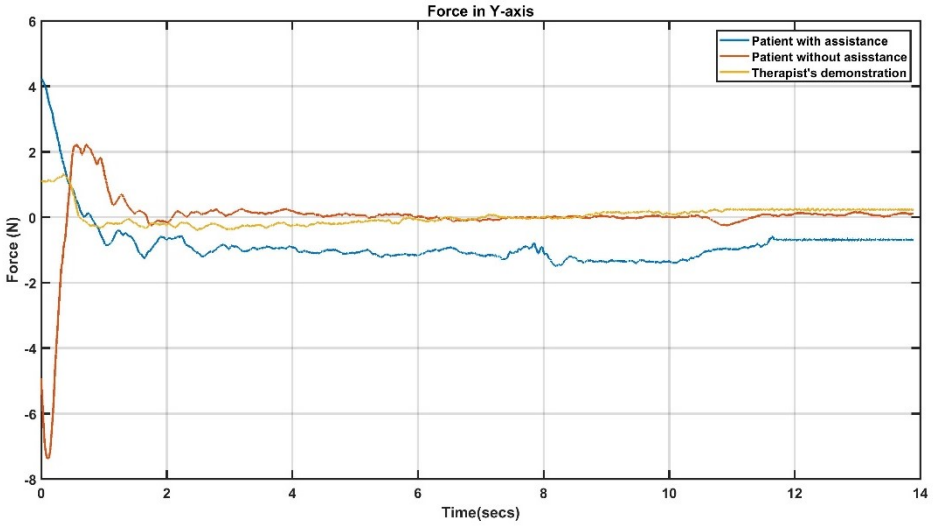

Figure 5-14 Measured forces along Y-axis.

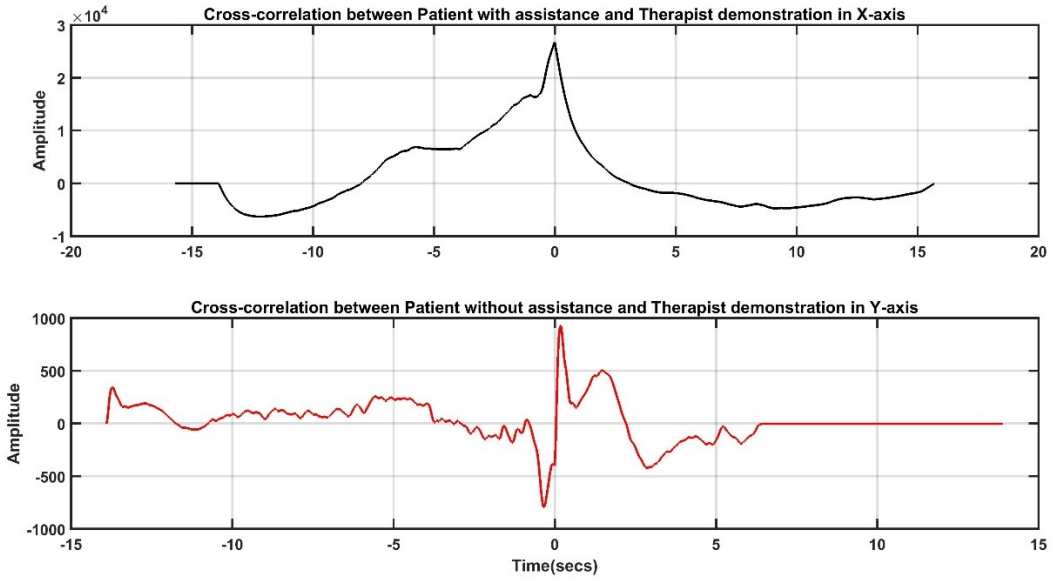

Figure 5-15 Cross-correlation plots of patient's reproduction with assistance (Top) and patient's reproduction without assistance (Bottom) along the X-axis.

For the next analysis, we decided to use the frequency domain. Figure 5-17 shows the patient's reproduction with and without assistance in X-axis. While Figure 5-18 shows the patient's reproduction with and without assistance in Y-axis.

The results show that the patient's reproduction with robotic assistance share more similarities with the therapist's demonstration, while the patient's reproduction without robotic assistance does not show any similarity. Thanks to these results, one more time we can support the accuracy and effectiveness of SEDS.

Finally, numerical results are shown in Table 5-2. As in Task 1 analysis, it shows the obtained results of the correlation coefficients (CC), mean square error (MSE), and Euclidian norm (ED) analysis. The obtained results in Table 5-2 show interesting results; CC shows the close relationship and better results between the assisted reproduction and the therapist's demonstration. MSE shows a significant error in the assisted reproduction along the X-axis. ED shows better results for the non-assisted reproduction. There is an explanation for these results. As mentioned before, using Figure 5-13 and Figure 5-14 we can see that the patient's reproduction with robotic assistance does not stay at the equilibrium point, for this reason, we got "unexpected" results during this last analysis.

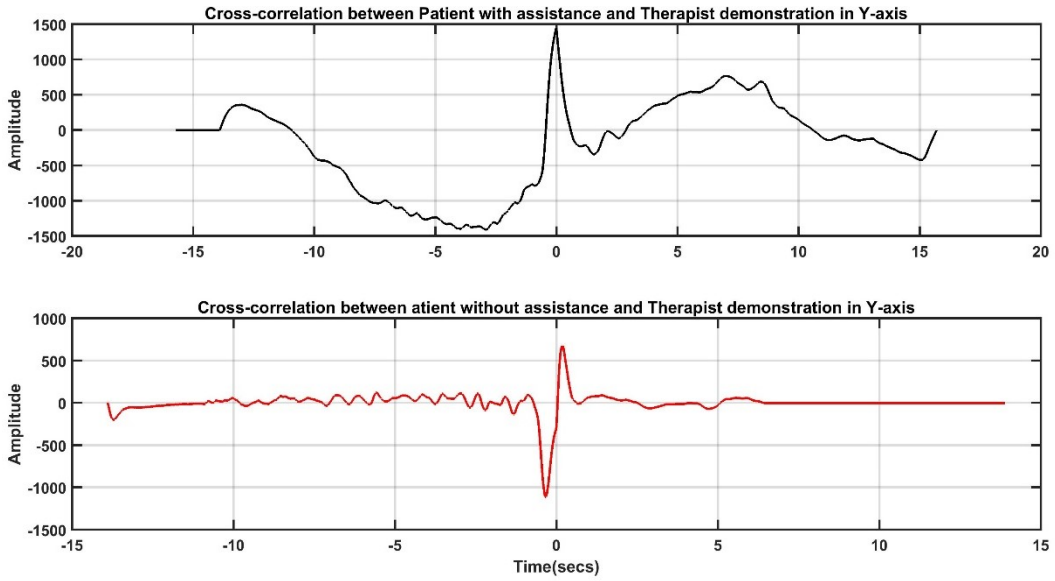

Figure 5-16 Cross-correlation plots of patient's reproduction with assistance (Top) and patient's reproduction without assistance (Bottom) along the Y-axis.

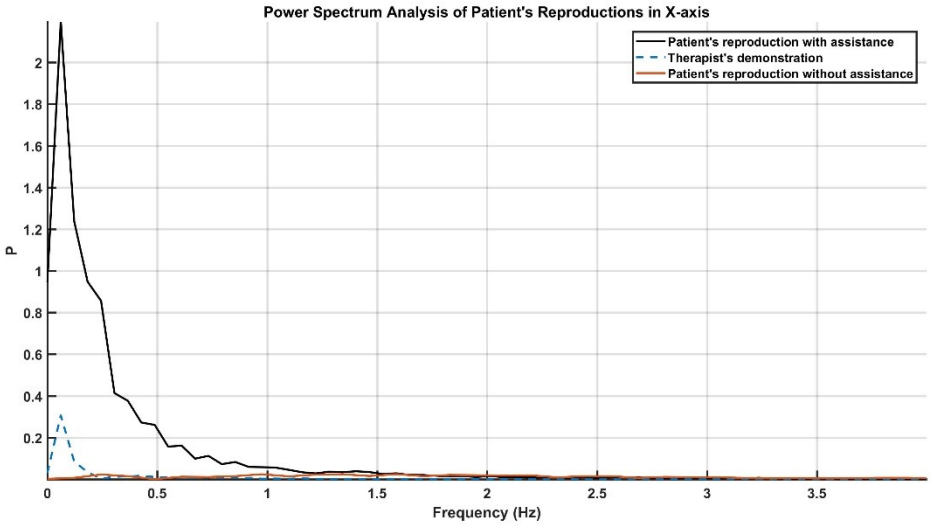

Figure 5-17 Power Spectrum analysis of patient's reproduction with and without robotic assistance and therapist's demonstration in X-axis.

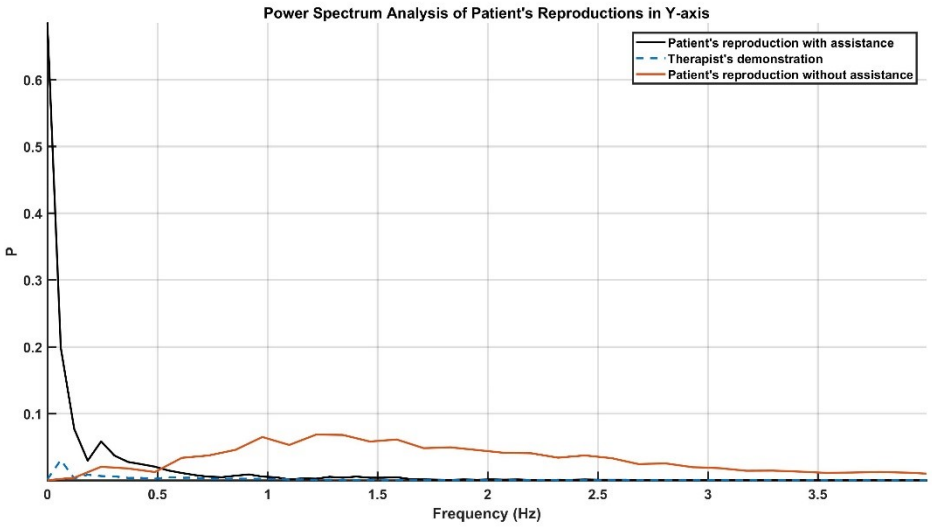

Figure 5-18 Power Spectrum analysis of patient's reproduction with and without robotic assistance and therapist's demonstration in Y-axis.

Table 5-2 Numerical analysis of force control task using SEDS

Method	Patient's Reproduction Analysis			
	Assisted-X	Assisted-Y	Non-assisted-X	Non-assisted-Y
CC	0.8132	0.6647	-0.3335	-0. 5534
MSE	5.7936	1.1044	1.1659	1.3252
ED	283.7095	123.8685	127.2720	135.6879

Given the previous results, we can conclude that SEDS presented an outstanding outcome. It generalized the demonstrated behavior even for non-demonstrated initial points. In general, results show way better reproduction when using robotic assistance.

5.7 Conclusion and summary

The demand for therapy services is increasing due to stroke. This is becoming a serious problem due to the limited resources of the healthcare system, especially because of the limited number of therapists. Robots' characteristics make them an excellent tool to carry out the physically demanding and repetitive tasks of therapy.

In this chapter, we demonstrated that the presence of therapists is not necessary throughout the entire therapy session; robots can be used to learn and continue their role and help the patients while therapists can share their time with other patients. Even though LfD is not a new tool, it has a huge potential in robotic rehabilitation to develop more intelligent and more reliable devices. SEDS proves that it is a solid, robust and reliable algorithm to solve the demanding necessity of medical robotics improvements. In the future, we will implement an LfD system combined with AAN feature (Maaref, 2016). This improvement will allow the system to assist the patient only when it is necessary. In other words, if the patient is performing the task, then the system will not interfere. If the patient cannot complete the task, the system will assist the patient.

6 CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

This thesis presented a step forward in the rehabilitation robotics context. By using Learning from Demonstration (LfD) algorithms, we showed that there is a paradigm in the rehabilitation field where the robots can learn the therapist's behavior and reproduce it even for fairly complex tasks. Even though there are some publications related to this topic, this is a relatively new technology that has to be studied and developed to bring more and better results to the healthcare system and the patients with a post-stroke disability. Due to its novelty, the potential of this new paradigm in the rehabilitation field has not been exploited fully yet.

In Chapter 3, we proposed a cooperative task between a patient and a therapist using a telerehabilitation system. In this task, the patient and the therapist interacted through a telerobotic system to perform a cooperative task, namely to lift a bar. In this experiment, the main goal was to show the robustness of LfD algorithms especially Gaussian Mixture Models (GMM) and Gaussian Mixture Regression (GMR) and the fact that therapist-patient cooperation can be facilitated by robotic systems. Due to its adaptability and simple implementation, we took these algorithms as the most basic combination of rehabilitation robots and LfD. Therefore, this is considered as the starting point of this research.

In Chapter 4, a semi-periodic motion tracking task was considered to show that LfD algorithms can learn and reproduce more complex tasks compared to simple point-to-point reaching tasks. In this case, we learned the frequency, amplitude, and periodicity of

a given semi-periodic motion. Using GMM and GMR algorithms as well as Fourier Series (FS), the system was able to learn a periodic motion.

Chapter 5 presented a more complex and powerful LfD algorithm called Stable Estimator of Dynamical Systems (SEDS). This chapter demonstrates the robustness of this algorithm by performing two different experiments. The first experiment involves a 2 degrees of freedom (DoF) reaching motion task, where the patient has to move the robot's end effector from a dynamic initial point to a fixed target (the origin) following the desired trajectory. The second experiment involves a 2-DoF force task, where the patient has to move the robot's end-effector from an initial point to a target following the desired force. This experiment was implemented to show how this algorithm can be used for complex Activities of Daily Living (ADL) that involve force and position. In both cases, the therapist demonstrated the task during the demonstration phase, and later, during the therapist's absence, the robot took the therapist's role and interacted with the patient as showed during the demonstration phase. It is clear that due to the SEDS ability to converge to the desired value, even for non-demonstrated data points, it makes the robot programming easier, safer, and more reliable.

6.2 Future Work

6.2.1 Implement more complex ADLs with sophisticated rehabilitation robots

Due to the importance of ADL in rehabilitation therapy and given the complexity of these tasks, new multi-DoF motions have to be implemented to allow for a better therapy. In Chapter 3, we presented a cooperative task in 1D space while in Chapter 5 we presented tasks in 2D space. It is clear that to perform most of the ADL, the rehabilitation systems must be able to move in a 3D space without any limitations. To do this, rehabilitation robotic systems such as the Kinova arm have to be used.

6.2.2 SEDS applied to periodic motions

As mentioned in Chapter 4, there are different ways to reproduce a semi-periodic motion using LfD. GMM and GMR as well as FS, were used in this thesis to show a simple and reliable way to learn and reproduce these motions. Nevertheless, another LfD algorithm

such as SEDS could be used for this purpose. As mentioned in Chapter 5, SEDS would ensure global asymptotic stability of the system as well as a safer and more reliable system.

6.2.3 LfD algorithms applied to patient's and therapist's assessment

Until now, most of the research in smart rehab robots has been focused on rehabilitation following a standard procedure, where the therapist trains the system and later the system interacts with the patient in the therapist's absence. Nevertheless, these systems could also be used in a different procedure where the robot learns the patient's behavior, and later, it interacts with the therapist to give him/her an idea about the patient's motor abilities. Finally, these systems could be used for therapists' training purposes. The therapists in training could interact with the robotic systems while it learns and records their behaviors. Later, an expert therapist can evaluate their performance by asking the robot to reproduce the learned behaviors.

6.2.4 Reinforcement learning applied to the current systems

Until now, machine learning algorithms have been used to train the robots and reproduce the therapist's behavior. Simple, well known, and popular algorithms have been implemented due to their reliability and simplicity. A next step in machine learning algorithms could be reinforcement learning. This algorithm is a step forward from the previous algorithms. It uses the previous experience to improve the outcomes of the system. This algorithm finds the best result based on trial and error. By combining these algorithms with the rehabilitation systems presented in this thesis, better, faster, and more complex results in therapy session could be obtained. As a result, patients could get a better quality of life and a faster recovery time.

6.2.5 Perform experiments with people with disabilities

This is probably the most important step for continuing the work in this thesis. The work presented in this thesis was tested with non-disabled persons wearing a transcutaneous electric nerve stimulator (TENS) and with springs attached to their hand (the robot's end-effector) to simulate the inability to move in a straight line. By testing these systems with

real patients, a better and more complete idea about the capabilities and limitations of the proposed works could be obtained. If we can prove that the proposed systems are as good as traditional therapy, that is a positive step.

REFERENCES

Association, N. S. (2018, March 23). *Explaining a Stroke.* Retrieved from Stroke: http://www.stroke.org/sites/default/files/resources/ExplainingStrokeBrochure.pdf

Atashzar, S. F. (2012). Networked teleoperation with non-passive environment: Application to tele-rehabilitation. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 5125-5130). Vilamoura.

Atashzar, S. F. (2015). A new passivity-based control technique for safe patient-robot interaction in haptics-enabled rehabilitation systems. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (pp. 4556-4561). Hamburg.

Atkeson, C. G. (1997). Robot learning from demonstration. *ICML* , 12-20.

Balasubramanian, S. Z. (2010). Cooperative and active assistance based interactive therapy. *Complex Medical Engineering (CME), 2010 IEEE/ICME International Conference*, (pp. 311–315). Gold Coast.

Billard, S. M. Z. (2011). Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Transactions on Robotics*, 943–957.

Billard., E. G. (2009). Learning nonlinear multi-variate motion dynamics for real-time position and orientation control of robotic manipulators. *2009 9th IEEE-RAS International Conference on Humanoid Robots*, (pp. 472-477). Paris.

Bogart, J. (2017, December 6). *Stroke Report 2016*. Retrieved from Canadian Stroke Best Practice: http://www.strokebestpractices.ca/news-feature/stroke-report-2016-just-released/

Bradley, S. P. (1992). Applied mathematical programming. *Reading, Mass.: Addison-Wesley*.

Calinon, S. (2009). *Robot programming by demonstration: a probabilistic approach.* Lausanne, Switzerland: EPFL Press.

Calinon, S. (2015). A tutorial on task-parameterized movement learning and retrieval. *Intelligent Service Robotics*, 1–29.

Calinon, S. B. (2007). Incremental learning of gestures by imitation in a humanoid robot. *Proceeding of the ACM/IEEE international conference on Human-Robot interaction - HRI 07*, (pp. 255-262). Arlington, Virginia, USA.

Calinon, S. B. (2007). What is the teacher's role in robot programming by demonstration?: Toward benchmarks for improved learning. *Interaction Studies*, 441-464.

Calinon, S. D. (2010). Learning and Reproduction of Gestures by Imitation. *IEEE Robotics & Automation Magazine*, 44–54.

Calinon, S. G. (2006). On learning the statistical representation of a task and generalizing it to various contexts. *Proceedings 2006 IEEE International Conference on Robotics and Automation*, (pp. 2978-2983). Orlando.

Calinon, S. L. (2012). Statistical dynamical systems for skills acquisition in humanoids. *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, (pp. 323-329). Osaka.

Canadians, 2. r. (2017, December 6). *Getting to the heart of the matter*. Retrieved from Heart and Stroke : http://www.heartandstroke.ca/-/media/pdf-

files/canada/2017-heart-month/heartandstroke-reportonhealth-2015.ashx?la=en&hash=497A83F1FE8388479DC5D7DB27322C191B866D57

Carignan CR, K. H. (2006). Telerehabilitation robotics: Bright lights, big future? . *The Journal of Rehabilitation Research and Development*, 695-710.

Colombo, G. J. (2000). Treadmill training of paraplegic patients using a robotic orthosis. *Journal of rehabilitation research and development*, 693-700.

Colombo, R. P. (2005). Robotic techniques for upper limb evaluation and rehabilitation of stroke patients. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 311-324.

Deutsch, J. E. (2001). Post-stroke rehabilitation with the Rutgers Ankle System: a case study. . *Presence: Teleoperators & Virtual Environments*, 416-430.

Devautl, T. F. (2015). Learning from Demonstration for Distributed, Encapsulated Evolution of Autonomous Outdoor Robots. *Proceedings of the Companion Publication of the 2015 on Genetic and Evolutionary Computation Conference on Genetic and Evolutionary Computation*, (pp. 1381-1382). Madrid.

E. C. Lu, R. W. (2011). Development of a robotic device for upper limb stroke rehabilitation: A user-centered design approach. *Journal of Behavioral Robotics*, 176-184.

Evrard, P. G. (2009). Teaching physical collaborative tasks: object-lifting case study with a humanoid. *2009 9th IEEE-RAS International Conference on Humanoid Robots*, (pp. 399-404). Paris.

Federation, W. H. (2017, December 6). *Stroke*. Retrieved from Stroke: http://www.world-heart-federation.org/cardiovascular-health/stroke/

Foundation, H. &. (2017, February 14). *Statistics - heart and stroke foundation of Canada,*. Retrieved from heartandstroke.ca: http://www.heartandstroke.com/site/c.ikIQLcMWJtE/b.3483991/k.34A8/Statistics.htm

Gribovskaya, E. K.-Z. (2010). Learning Non-linear Multivariate Dynamics of Motion in Robotic Manipulators. *The International Journal of Robotics Research*, 80–117.
Guidali, M. D. (2011). A robotic system to train activities of daily living in a virtual environment. *Medical & Biological Engineering & Computing*, 1213–1223.

Hagis, C. (2003). *History of Robots.* Citerseer.

Hogan, N. K. (1992). MIT-MANUS: a workstation for manual therapy and training. I. *[1992] Proceedings IEEE International Workshop on Robot and Human Communication*, (pp. 161-165). Tokyo.

Hammond, A. (2004) What is the role of the occupational therapist?. *Best practice & research Clinical rheumatology*, 491-505.

Kido, K. (2015). *Digital Fourier Analysis: Fundamentals.* New York, NY: Springer New York.

Konidaris, G. K. (2012). Robot learning from demonstration by constructing skill trees. *The International Journal of Robotics Research*, 360–375.

Krebs, H. F. (2003). Rehabilitation Robotics: Performance-Based Progressive Robot-Assisted Therapy. *Autonomous Robots*.

Krebs, H. I. (2004). Rehabilitation robotics: pilot trial of a spatial extension for MIT-Manus. *Journal of NeuroEngineering and Rehabilitation*.

Krebs, H. I. (2006). Therapeutic Robotics: A Technology Push. *Proceedings of the IEEE*, 1727-1738.

Lauretti, C. C. (2017). Learning by Demonstration for Planning Activities of Daily Living in Rehabilitation and Assistive Robotics. *IEEE Robotics and Automation Letters*, 1375–1382.

Lee, S. H. (2012). Learning basis skills by autonomous segmentation of humanoid motion trajectories. *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, (pp. 112-119). Osaka.

Legg, L. D. (2006). Occupational therapy for patients with problems in activities of daily living after stroke. *Cochrane Database of Systematic Reviews*, 922.

Loos, H. R. (2008). Rehabilitation and Health Care Robotics. In K. O. Siciliano B., *Springer Handbook of Robotics* (pp. 1223–1251). Berlin: Springer.

Lu, E. C. (2012). Development of a robotic device for upper limb stroke rehabilitation: A user-centered design approach. *Paladyn, Journal of Behavioral Robotics*, 176-184.

Lum, P. R. (2002). Robotic Devices for Movement Therapy After Stroke: Current Status and Challenges to Clinical Acceptance. *Topics in Stroke Rehabilitation*, 40–53.

Lum, P. S. (2002). Robot-assisted movement training compared with conventional therapy techniques for the rehabilitation of upper-limb motor function after stroke. *Archives of physical medicine and rehabilitation*, 952-959.

Lum, P. S. (2004). Evidence for improved muscle activation patterns after retraining of reaching movements with the MIME robotic system in subjects with post-stroke hemiparesis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 186-194.

Lydakis, A. M. (2017). A learning-based agent for home neurorehabilitation. *2017 International Conference on Rehabilitation Robotics (ICORR)*, (pp. 1233-1238). London.

Maaref, M. R. (2016). A Bicycle Cranking Model for Assist-as-Needed Robotic Rehabilitation Therapy Using Learning From Demonstration. *IEEE Robotics and Automation Letters*, 653–660.

Maaref, M. R. (2016). A gaussian mixture framework for co-operative rehabilitation therapy in assistive impedance-based tasks. *IEEE Journal of Selected Topics in Signal Processing*, 904–913.

Mcclure, J. S. (2012). Adherence to Canadian Best Practice Recommendations for Stroke Care: Vascular Cognitive Impairment Screening and Assessment Practices in an Ontario Inpatient Stroke Rehabilitation Facility. *Topics in Stroke Rehabilitation*, 141-148.

Movahedazarhouligh, S. V. (2015). Feasibility of telerehabilitation implementation as a novel experience in rehabilitation academic centers and affiliated clinics in Tehran: assessment of rehabilitation professionals' attitudes. . *International journal of telemedicine and applications*, 7.

Najafi, M. A. (2017). Robotic learning from demonstration of therapist's time-varying assistance to a patient in trajectory-following tasks. *2017 International Conference on Rehabilitation Robotics (ICORR)*, (pp. 888-894). London.

Peternel, L. P. (2013). Teaching robots to cooperate with humans in dynamic manipulation tasks based on multi-modal human-in-the-loop approach. *Autonomous Robots*, 123–136.

Popescu, V. G. (2000). A virtual-reality-based telerehabilitation system with force feedback. *IEEE Transactions on Information Technology in Biomedicine*, 45-51.

Reinkensmeyer, D. J.-C. (2000). Understanding and treating arm movement impairment after chronic brain injury: progress with the ARM guide. *Journal of rehabilitation research and development*, 653-662.

Reynolds, D. (2017, December 6). *Gaussian Mixture Models.* Retrieved from MIT Lincoln Laboratory: https://www.ll.mit.edu/mission/cybersec/publications/publication-files/full_papers/0802_Reynolds_Biometrics-GMM.pdf.

Roby-Brami, A. F. (2003). Motor compensation and recovery for reaching in stroke patients. *Acta Neurologica Scandinavica*, 369–381.

Sainburg, R. G. (2013, December 6). Bilateral Synergy: A Framework for Post-Stroke Rehabilitation. *Journal of neurology & translational neuroscience.*, 1025. Retrieved from http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3984050/

Schaal, S. I. (2003). Computational approaches to motor learning by imitation. *Philosophical Transactions of the Royal Society*, 537–547.

Skillicorn, D. B. W. (2018). The Design Space of Social Robots. *arXiv preprint arXiv:1801.04857*.

Tao, R. (2014). *Haptic Teleoperation Based Rehabilitation Systems for Task-Oriented Therapy MSc thesis.* Edmonton, AB: University of Alberta.

Tavakoli, J. F. (2018). Kinesthetic Teaching of a Therapist's Behavior to a Rehabilitation Robot. *International Symposium of Medical Robotics.* Atlanta.

Veerbeek, J. M. (2011). Early Prediction of Outcome of Activities of Daily Living After Stroke: A Systematic Review. *Stroke*, 1482–1488.

Voelker, R. (2005). Rehabilitation Medicine Welcomes a Robotic Revolution. *JAMA*, 1191-1195.

Volpe, B. K. (1999). Robot training enhanced motor outcome in patients with stroke maintained over 3 years. *Neurology*, 1874.

Williams, D. J. (2001). A robot for wrist rehabilitation. *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE*, 1336-1339.

Worsnopp, T. T. (2007). An actuated finger exoskeleton for hand rehabilitation following stroke. *In Rehabilitation Robotics, 2007. ICORR 2007. IEEE 10th International Conference*, 896-901.

Wright, J. N. (2006). 18 Sequential quadratic programming. *Numerical Optimization* , 529–562.

Zariffa, J. K. M. (2012). Relationship Between Clinical Assessments of Function and Measurements From an Upper-Limb Robotic Rehabilitation Device in Cervical Spinal Cord Injury. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 341-350.

APPENDIX A: REHAB ROBOT INVERSE AND FORWARD KINEMATICS

In this thesis, a 2-DOF Rehabilitation Robot (Quanser, Inc. Markham, Canada) was used to perform most of the experiments. The robot use QUARC, a Quanser's real-time control software running in Matlab and Simulink. The software provides access to low-level sensors and actuators. In this appendix, we provide the robot's kinematics and identified dynamics.

The robot has two motors that move the robot's end effector through a capstan disc as shown in **Figure A0-1**. The forward kinematics relate the robot's joint angles to its end effector position in Cartesian space. Using **Figure A0-2**, it can be shown that the forward kinematics relations are

$$x \doteq h_x(\theta_1, \theta_2) = d_1 \cos(\theta_1) + d_2 \sin(\theta_2) \tag{A-1}$$

$$y \doteq h_y(\theta_1, \theta_2) = d_1 \sin(\theta_1) + d_2 \cos(\theta_2), \tag{A-2}$$

where d_1 = 10'' and d_2 = 10.5''.

Due to the robot's design, the physical limitations on its joint angles are:

$$-55° \leq \theta_1 \leq 90° \tag{A-3}$$

$$0° \leq \theta_2 \leq 145° \tag{A-4}$$

$$\theta_1 - (\theta_2 - 90°) \geq 35° \tag{A-5}$$

The first and second limitations (A-3) (A-4) prevent the links from colliding with the physical stops at the extremities of the capstan disc. The last limitation (A-5) prevents a collision between the links.

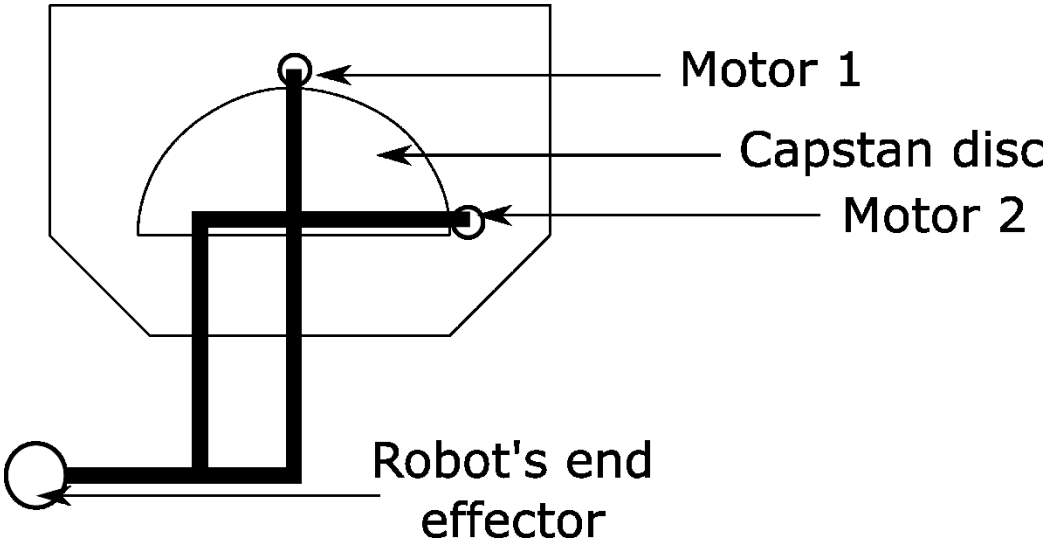

Figure A0-1 Rehab robot components.

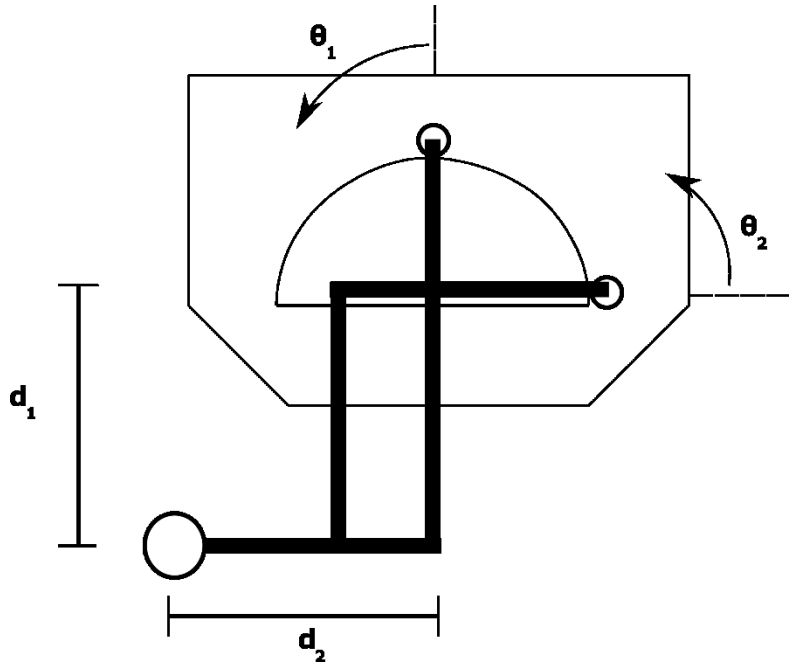

Figure A0-2 Rehab robot links and joint angles.

Using the forward kinematics, we can compute the robot's Jacobian as follows:

$$J \doteq \begin{bmatrix} \frac{\partial h_x}{\partial \theta_1} & \frac{\partial h_x}{\partial \theta_2} \\ \frac{\partial h_y}{\partial \theta_1} & \frac{\partial h_y}{\partial \theta_2} \end{bmatrix} = \begin{bmatrix} -d_1 \sin(\theta_1) + d_2 \cos(\theta_2) \\ d_1 \cos(\theta_1) + d_2 \sin(\theta_2) \end{bmatrix} \tag{A-6}$$

Using Figure A0-3, we can get the inverse kinematics with a simple geometric analysis. Applying the law of cosines to ΔABC, $\overline{AB} = d_1$ and $\overline{BC} = d_2$ we get

$$\alpha = \text{acos}\left(\frac{d_1^2 + d_2^2 - \overline{AB}^2}{2 d_1 d_2}\right) \tag{A-7}$$

$$\beta = \text{acos}\left(\frac{\overline{AC}^2 + d_1^2 - d_2^2}{2 d_1 \overline{AC}}\right) \tag{A-8}$$

Notice that

$$\gamma = atan2(y, x) \tag{A-9}$$

$$\overline{AC} = \sqrt{x^2 + y^2} \tag{A-10}$$

where the $atan2(.)$ term denotes the fourth quadrant arctangent. Given that $\theta_1 = \beta + \gamma$ and $\theta_2 = \theta_1 + \alpha - 90°$, the inverse kinematics relations are given by:

$$\theta_1 = \text{acos}\left(\frac{x^2 + y^2 + d_1^2 - d_2^2}{2 d_1 \sqrt{x^2 + y^2}}\right) + atan2(y, x) \tag{A-11}$$

$$\theta_2 = \theta_1 + \text{acos}\left(\frac{d_1^2 + d_2^2 - x^2 - y^2}{2d_1 d_2}\right) - 90° \quad \text{(A-12)}$$

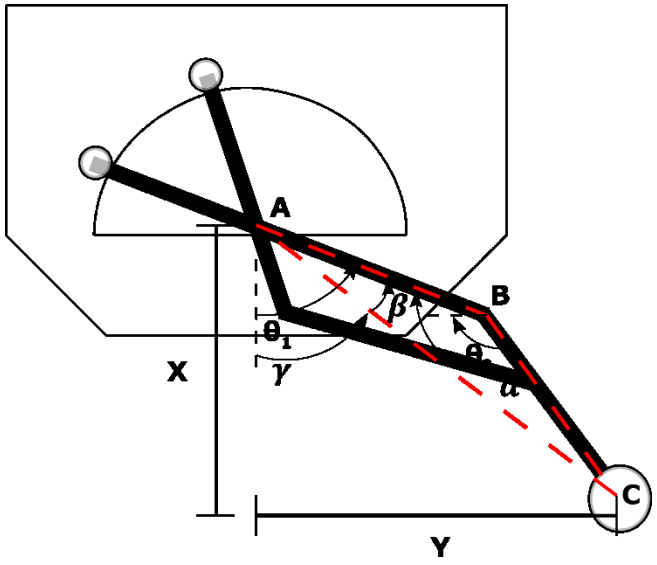

Figure A0-3 Model used to solve inverse kinematics.

APPENDIX B: MATLAB CODE

In this thesis, MATLAB was used as the main tool to program and implement the experiments presented in Chapters 3, 4 and 5. Appendix B will present the most important MATLAB codes and functions used, as well as a brief explanation on how they have been used.

B.1 Code for Chapters 3 and 4

The code used for Chapters 3 and 4 shared the same methodology and functions. The difference between the codes is the number and type of data values used to train the system. The used methodology follows the following steps:

1. Record data from the experiments of the demonstration phase.
2. Load the recorded dataset.
3. Build a matrix using the recorded data (following the GMM function requirements) with the values to be used for the GMM function.
4. Initialize the options to be used by the GMM and GMR functions.
5. Call the GMM function.
6. Use the obtained GMM results in the Simulink model.
7. Call the GMR function in real time from the Simulink model.

To give an example about the code used for these experiments, the most important sections of the code used in Chapter 4 is presented. Notice that the code used for Chapter 4 is a more complex version of the code used for Chapter 3.

```
clear all
clc


%% Initialize Values and variables such as time-step, number of data points to be used
to train the system, window size, etc.

Initialize_Values

%% Load the recorded data
Load_data

%% Compute polar coordinates
```

```
if (r_T_recorded ~= 1)
    for i = 1:demos
        variables = eval(['data',num2str(i)]);
        r         = sqrt(variables(2,:).^2+variables(3,:).^2);  %Compute r value
        theta     = atan2(variables(3,:),variables(2,:));       %Compute theta value
        theta_dot = horzcat((theta(1)-theta(2))/timeStep,diff(theta)/timeStep);
%Compute first derivative
        data1 = vertcat(r,theta,theta_dot,variables);           %Save the computed
values in a matrix
    end
end


%% Analyze Data to find the FSC

fft_Window_x = data1(5,1:windowSize);      %Create a window of the original signal
along X-axis
fft_Window_y = data1(6,1:windowSize);      %Create a window of the original signal
along Y-axis

T =  windowSize/1000; % period (secs/cycle)
N = windowSize-1 ; % number of discrete data
ffreq = 2*pi/T ; % fundamental frequency

fx=fft_Window_x;
fhatx = fx ;
fhatx(1) = (fx(1)+fx(N+1))/2 ;
fhatx(N+1) = [] ;

fy=fft_Window_y;
fhaty = fy ;
fhaty(1) = (fy(1)+fy(N+1))/2 ;
fhaty(N+1) = [] ;

unos = ones(1,7);

tWindow = t(1:10000);
FSCx = zeros(4,7,demoLen-windowSize);
FSCy = zeros(4,7,demoLen-windowSize);
As_Array = zeros(7,demoLen-windowSize);
Bs_Array = zeros(7,demoLen-windowSize);
fsc_x = [];     %Initialize an array to store the FSC of X
fsc_y = [];     %Initialize an array to store the FSC of Y

% The following for-loop is used to get the FSC of every single window
for i=0:(demoLen-windowSize-1)
    F = fft(fhatx,N) ;      %Get the FFT of the windowed data
    F=F(1:(N+1)/2) ;        %Use only one half of the data
    k=0:(N/2-1) ;           %Vector used in combination with the frequency to store
the resolution of the signal
    omega=k*ffreq ;         % in units of rads/sec
    A = 2*real(F)/N ;       %Vector to store the a_n values
    A(1)= A(1)/2 ;          %a_0 value
    B = -2*imag(F)/N ;      %Vector used to store the b_n values

    FSCx(:,:,i+1) = [A(1)*unos; A(2:8); B(2:8); omega(2:8)]; %Matrix used to store the
FSC values


    %%%%%%This section does the same things for Y-axis%%%%%
    F = fft(fhaty,N) ;
    F=F(1:(N+1)/2) ;
    k=0:(N/2-1) ;

    omega=k*ffreq ; % in units of rads/sec
    % extracting the coefficients
    % ---------------------------
    A = 2*real(F)/N ;
    A(1)= A(1)/2 ;
```

```
    B = -2*imag(F)/N ;

    FSCy(:,:,i+1) = [A(1)*unos; A(2:8); B(2:8); omega(2:8)];

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    fft_Window_x = data1(5,1+i:windowSize+i);     %Move the window of X-axis
    fft_Window_y = data1(6,1+i:windowSize+i);     %Move the window of Y-axis
    tWindow = t(1+i:windowSize+i);                %Move the window of time
    fx=fft_Window_x;                              %Update the new window
    fhatx = fx ;                                  %Update the new variable
    fhatx(1) = (fx(1)+fx(N+1))/2 ;
    fhatx(N+1) = [] ;
    fy=fft_Window_y;
    fhaty = fy ;
    fhaty(1) = (fy(1)+fy(N+1))/2 ;
    fhaty(N+1) = [] ;
end

%% Build a matrix (following the GMM function requirements) with the values to be used
for the GMM function

temArray = [];
temArray = t(1:demoLen-windowSize);
temArray = vertcat(temArray,data1(1,1:demoLen-windowSize));    %r
temArray = vertcat(temArray,data1(2,1:demoLen-windowSize));    %Theta
temArray = vertcat(temArray,reshape(FSCx(1,1,:),1,demoLen-windowSize));    %A0
temArray = vertcat(temArray,reshape(FSCx(2,1,:),1,demoLen-windowSize));    %A1
temArray = vertcat(temArray,reshape(FSCx(2,2,:),1,demoLen-windowSize));    %A2
temArray = vertcat(temArray,reshape(FSCx(2,3,:),1,demoLen-windowSize));    %A3
temArray = vertcat(temArray,reshape(FSCx(2,4,:),1,demoLen-windowSize));    %A4
temArray = vertcat(temArray,reshape(FSCx(2,5,:),1,demoLen-windowSize));    %A5
temArray = vertcat(temArray,reshape(FSCx(2,6,:),1,demoLen-windowSize));    %A6
temArray = vertcat(temArray,reshape(FSCx(2,7,:),1,demoLen-windowSize));    %A7
temArray = vertcat(temArray,reshape(FSCx(3,1,:),1,demoLen-windowSize));    %B1
temArray = vertcat(temArray,reshape(FSCx(3,2,:),1,demoLen-windowSize));    %B2
temArray = vertcat(temArray,reshape(FSCx(3,3,:),1,demoLen-windowSize));    %B3
temArray = vertcat(temArray,reshape(FSCx(3,4,:),1,demoLen-windowSize));    %B4
temArray = vertcat(temArray,reshape(FSCx(3,5,:),1,demoLen-windowSize));    %B5
temArray = vertcat(temArray,reshape(FSCx(3,6,:),1,demoLen-windowSize));    %B6
temArray = vertcat(temArray,reshape(FSCx(3,7,:),1,demoLen-windowSize));    %B7
Data_x   = temArray;



temArray = [];
temArray = t(1:demoLen-windowSize);
temArray = vertcat(temArray,data1(1,1:demoLen-windowSize));    %r
temArray = vertcat(temArray,data1(2,1:demoLen-windowSize));    %Theta
temArray = vertcat(temArray,reshape(FSCy(1,1,:),1,demoLen-windowSize));    %A0
temArray = vertcat(temArray,reshape(FSCy(2,1,:),1,demoLen-windowSize));    %A1
temArray = vertcat(temArray,reshape(FSCy(2,2,:),1,demoLen-windowSize));    %A2
temArray = vertcat(temArray,reshape(FSCy(2,3,:),1,demoLen-windowSize));    %A3
temArray = vertcat(temArray,reshape(FSCy(2,4,:),1,demoLen-windowSize));    %A4
temArray = vertcat(temArray,reshape(FSCy(2,5,:),1,demoLen-windowSize));    %A5
temArray = vertcat(temArray,reshape(FSCy(2,6,:),1,demoLen-windowSize));    %A6
temArray = vertcat(temArray,reshape(FSCy(2,7,:),1,demoLen-windowSize));    %A7
temArray = vertcat(temArray,reshape(FSCy(3,1,:),1,demoLen-windowSize));    %B1
temArray = vertcat(temArray,reshape(FSCy(3,2,:),1,demoLen-windowSize));    %B2
temArray = vertcat(temArray,reshape(FSCy(3,3,:),1,demoLen-windowSize));    %B3
temArray = vertcat(temArray,reshape(FSCy(3,4,:),1,demoLen-windowSize));    %B4
temArray = vertcat(temArray,reshape(FSCy(3,5,:),1,demoLen-windowSize));    %B5
temArray = vertcat(temArray,reshape(FSCy(3,6,:),1,demoLen-windowSize));    %B6
temArray = vertcat(temArray,reshape(FSCy(3,7,:),1,demoLen-windowSize));    %B7
Data_y   = temArray;



val=6;  %Number of Gaussians
```

```
numbOfTrain =2; %Number of values used to train the GMM

%Call GMM Function
GMM_GMR
```

In the previous code, the functions named "Initialize_Values" and "Load_data" presented at the beginning are used to initialize the basic variables such as time-step or window-size and to load the recorded data respectively.

The GMM_GMR function is a function that calls some GMM-GMR functions developed and presented in (Calinon S. , Robot programming by demonstration: a probabilistic approach., 2009). This function trains the system and computes the GMM parameters that will be later used by the GMR function. The functions called in the GMM_GMR functions are as follow

```
auxData = Data_x(1:numbOfTrain,:);
Data_xa0  = vertcat(auxData,Data_x(initVariable,:));
model_xa0 = init_GMM_kmeans(Data_xa0(:,1:trainingEND), model_xa0);
model_xa0 = EM_GMM(Data_xa0(:,1:trainingEND), model_xa0);
[DataOut_xa0, SigmaOut_xa0] = GMR(model_xa0,
Data_xa0(1:numbOfTrain,1:reproductionEND), 1:numbOfTrain, numbOfTrain+1);
```

Notice that in this case, and as mentioned in Chapter 4, the inputs used in the GMM function are the polar coordinates, while the values to be estimated are the FSC.

The presented code for GMM and GMR functions uses only a FSC (a_0) of X-axis. This is because of the implementation used in this experiment. In total, there is a GMM call and a GMR call for each FSC in X-axis and Y-axis.

After computing the GMM parameters, the GMR function is ready to be called and implemented in the Simulink model to control the rehabilitation robot.

B.2 Code for Chapter 6

The code used in Chapter 6 follows a similar methodology as the code used for Chapter 3 and 4. The steps to be followed are:

1. Record data from the experiments of the demonstration phase.
2. Load the recorded dataset.
3. Build a matrix using the recorded data following the SEDS function requirements.
4. Initialize the options to be used by the SEDS function.

5. Call the SEDS function.
6. Use the obtained SEDS results in the Simulink model.
7. Call the GMR function in real time from the Simulink model.

An example of the most important sections of the code used for Chapter 5 are presented. Notice that the SEDS functions were developed and explained in (Billard, 2011).

```
clear all
clc

% Load the recorded data during demonstration phase
load('2017_03_01_Curves_Test_7.mat')

%Creates a matrix with 4 columns (1:I values, 2:J values, 3:X values, 4:Y
%values), the matrix has the positions
Pos_Data = horzcat(III(:,2),JJJ(:,2),Master_Pos_Imp(:,2),Master_Pos_Imp(:,3));
%Creates a matrix with 4 columns (1:I values, 2:J values, 3:X values, 4:Y
%values), the matrix has the velocities
Vel_Data = horzcat(III(:,2),JJJ(:,2),Master_Vel_Imp(:,2),Master_Vel_Imp(:,3));

s=size(Pos_Data,1); %Numer of rows of the matix, it is used as limit
h=1; %Index for the rows of the matrix
r=1; %Index for the rows of the x_demos_pos matrix (it has a diferent index because it
is a new matrix)
q=1; %Index for the columns of the x_demos_pos matrix (it has a diferent index because
it is a new matrix)

%Take only the useful data in X-axis

while(h~=s)
    if(Pos_Data(h,1)~=Pos_Data(h,2)) %If i!=j then add those values to x_demos_pos
matrix
        i=h;
        r=1;
        while(Pos_Data(i,1)~=Pos_Data(i,2))
           x_demos_pos(r,q)=Pos_Data(i,3);
           y_demos_pos(r,q)=Pos_Data(i,4);
           i=i+1;
           r=r+1;
        end
        i=i-1; %We have to substract one unit because at the end of the main loop it
will add a unit again
        h=i;
        q=q+1;
    end
    h=h+1;
end

h=1; %Index for the rows of the matrix
r=1; %Index for the rows of the x_demos_vel matrix (it has a diferent index because it
is a new matrix)
q=1; %Index for the columns of the x_demos_vel matrix (it has a diferent index because
it is a new matrix)

%Take only the useful data in Y-axis

while(h~=s)
    if(Vel_Data(h,1)~=Vel_Data(h,2)) %If i!=j then add those values to x_demos_vel
matrix
        i=h;
        r=1;
        while(Vel_Data(i,1)~=Vel_Data(i,2))
```

```
            x_demos_vel(r,q)=Vel_Data(i,3);
            y_demos_vel(r,q)=Vel_Data(i,4);
            i=i+1;
            r=r+1;
        end
        i=i-1; %We have to substract one unit because at the end of the main loop it
will add a unit again
        h=i;
        q=q+1;
    end
    h=h+1;
end

%% Build the structure needed by the SED function
num_demo = 0;   %Index used to create demonstrations inside the structure
for f = 1:size(x_demos_pos,2)
    i=1;
    num_demo = num_demo+1;
    temp_val=horzcat(x_demos_pos(:,num_demo),y_demos_pos(:,num_demo));  %temporal
value used to build the structure with a demonstrated data
    while(i<=size(x_demos_pos,1))
        if (temp_val(i,1)~=0)
            xy_pos(i,:)=temp_val(i,:);
        end
        i=i+1;
    end
    demos{num_demo} = xy_pos';
    clear xy_pos
end

% Initialize variaables needed by the SEDS
for f = 1:size(demos,2)
    x = demos{1,f};
end

% Pre-processing
dt = 0.001; %The time step of the demonstrations
tol_cutting = 0.005; % A threshold on velocity that will be used for trimming demos

% Training parameters
K = 3; %Number of Gaussian funcitons

% A set of options that will be passed to the solver. Please type
% 'doc preprocess_demos' in the MATLAB command window to get detailed
% information about other possible options.
% options.tol_mat_bias = 10^-6; % A very small positive scalar to avoid
%                               % instabilities in Gaussian kernel [default: 10^-15]

options.display = 1;          % An option to control whether the algorithm
                              % displays the output of each iterations [default: true]

options.tol_stopping=10^-10;  % A small positive scalar defining the stoppping
                              % tolerance for the optimization solver [default: 10^-
10]

options.max_iter = 1500;       % Maximum number of iteration for the solver [default:
i_max=1000]

options.objective = 'mse';    % 'likelihood': use likelihood as criterion to
                              % optimize parameters of GMM
                              % 'mse': use mean square error as criterion to
                              % optimize parameters of GMM
                              % 'direction': minimize the angle between the
                              % estimations and demonstrations (the velocity part)
                              % to optimize parameters of GMM
                              % [default: 'mse']

%% SEDS learning algorithm
```

```
[tmp , tmp, Data, index] = preprocess_demos(demos,dt,tol_cutting); %preprocessing
datas

[Priors_0, Mu_0, Sigma_0] = initialize_SEDS(Data,K); %finding an initial guess for
GMM's parameter
[Priors,Mu,Sigma]=SEDS_Solver(Priors_0,Mu_0,Sigma_0,Data,options); %running SEDS
optimization solver

[y, Sigma_y, beta] = GMR(Priors, Mu, Sigma, x,1:d,d+1:2*d)
```

After computing the SEDS parameters, the system is ready to use the GMR function in the Simulink model used to interact with the rehabilitation robot.

APPENDIX C: DATA ANALYSIS

In this thesis, different data analysis methods were used to compare the obtained results. This appendix presents a brief introduction and definition of these methods.

C.1 Correlation Coefficient

The correlation coefficient (CC) are widely used in statistics to measure the relationship between two variables. One of the most common types of CC is the Pearson's correlation. The Pearson correlation coefficient is defined as:

$$\rho(A,B) = \frac{1}{N-1}\sum_{i=1}^{N}\left(\overline{\frac{A_\iota - \mu_\iota}{\sigma_A}}\right)\left(\frac{B_i - \mu_B}{\mu_B}\right) \quad \text{(B-1)}$$

where *A* and *B* are the two variables to be compared, *N* is the number of observations in each variable, X and X are the *mean* and *standard deviation* of *A*, X and X are the *mean* and *standard deviation* of *B.*

The results can be interpreted as:

- A CC of 1 means that for every positive increase in one of the variables, there is also a positive increase in the other variable.
- A CC of -1 means that for every positive increase in one of the variables, there is also a negative decrease in the other variable.
- A CC of 0 means that for every increase, there is not a positive or negative increase.
- Taking the absolute value of the CC gives the relationship strength. The larger the number, the stronger the relationship between both variables.

C.2 Mean Square Error

The mean square error (MSE) is used to tell how close a regression line is to a set of points. It takes the distance from the points to the regression line and squaring it. This

squaring helps to remove any possible negative sign and to weight large distances. In other words, it finds the average of a set of errors.

The MSE equation is given by:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \tag{B-2}$$

where $\hat{Y}$ is a vector of *n* predictions, and *Y* is a vector of observed values of the predicted variable.

C.3 Euclidean Distance

The Euclidian distance (ED) is the distance between two vectors *x* and *y* of *n* data points, and it is defined as:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{B-3}$$

It is the square root of the sum of squared differences between corresponding elements of the two compared vectors.