

AN INTELLIGENT MULTIMODAL UPPER-LIMB REHABILITATION  
ROBOTIC SYSTEM

by

ALEXANDROS LIOULEMES

Presented to the Faculty of the Graduate School of  
The University of Texas at Arlington in Partial Fulfillment  
of the Requirements  
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

JUNE 2017

AN INTELLIGENT MULTIMODAL UPPER-LIMB REHABILITATION  
ROBOTIC SYSTEM

The members of the Committee approve the doctoral  
dissertation of Alexandros Lioulemes

Dr. Fillia Makedon  
Supervising Professor

---

Dr. Vassilis Athitsos

---

Dr. Manfred Huber

---

Dr. Vangelis Karkaletsis

---

---

Dean of the Graduate School

---



Copyright © by Alexandros Lioulemes 2017

All Rights Reserved

I dedicate my Doctoral Dissertation to my parents, Eftalia and Giorgos, who taught me to do what I love, my sister Barbara, who used to scold me whenever I was doing something that I didn't love, and Angie, who loves my passion for what I do.

## ACKNOWLEDGEMENTS

I would like to thank my supervising professor Dr. Fillia Makedon for constantly motivating and encouraging me, and also for her invaluable advice during the course of my doctoral studies. I wish to thank my academic advisors Dr. Vangelis Karlaletsis, Dr. Vassilis Athitsos and Dr. Manfred Huber for their interest in my research and for taking time to serve in my dissertation committee. Last but not least, I would love to thank all my close and far away friends for being in my life and working hard with me for our future success and humanity's progress.

June 12, 2017

## ABSTRACT

### AN INTELLIGENT MULTIMODAL UPPER-LIMB REHABILITATION ROBOTIC SYSTEM

Alexandros Lioulemes, Ph.D.

The University of Texas at Arlington, 2017

Supervising Professor: Dr. Fillia Makedon

A traffic accident, a battlefield injury, or a stroke can lead to brain or musculoskeletal injuries that impact motor and cognitive functions and can drastically change a person's life. In such situations, rehabilitation plays a critical role in the ability of the patient to partially or totally regain motor function, but the optimal training approach remains unclear. Robotic technologies are recognized as powerful tools to promote neuroplasticity and stimulate motor re-learning. Moreover, they deliver high-intensity, repetitive, active and task-oriented training; in addition, they provide objective measurements for patient evaluation.

The primary focus of this research is to investigate the development of a safe human-robot interaction assessment and training system by utilizing physiological, kinematic and dynamic modalities. Such system places the user in the robot's control loop, by feeding back a patient's biomechanical, physiological and cognitive states. A proposed vision-based upper-limb monitoring system and a developed adaptive haptic guidance control mechanism will involve human intentions to generate adaptive perception and behaviors for the Barrett WAM robotic arm. To facilitate this,

a combined integration of computer vision, artificial intelligence, and human-robot interaction research are employed on the multi-sensing robotic platform.

Computational methods for a multimodal upper-limb robot-aided system are proposed in this dissertation; first, a virtual reality environment that assesses the user's physiological and psychological stages; second, an interface capable of estimating a patient's performance utilizing motion analysis and pattern recognition methods; third, an unobtrusive method for reconstructing upper-limb kinematics during robot-aided tasks with end-effector machines using Microsoft Kinect's skeletal tracking is presented and experimentally validated; fourth, an adaptive haptic guidance robotic controller is employed to modulate the complexity of the assigned motor tasks and increase the hand coordination abilities of the user. Finally, we show seven applications of robots in assistive environments, and we present the human-robot interaction usability case studies that are critical evaluation components of this thesis.

## TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . . .	v
ABSTRACT . . . . .	vi
LIST OF ILLUSTRATIONS . . . . .	xii
LIST OF TABLES . . . . .	xviii
Chapter	Page
1. Introduction . . . . .	1
1.1 Motivation . . . . .	1
1.2 Proposed System . . . . .	2
1.3 Multimodal Modular System . . . . .	4
1.4 Dissertation structure . . . . .	5
1.5 Research Questions . . . . .	6
1.6 Ethical Concerns . . . . .	7
2. Robot-assisted Rehabilitation . . . . .	8
2.1 Stroke Rehabilitation . . . . .	9
2.2 Rehabilitation Robotics . . . . .	10
2.3 Robot-based Assessment . . . . .	13
3. Data Acquisition . . . . .	16
3.1 Data Acquisition . . . . .	16
3.1.1 Motion Data . . . . .	16
3.1.2 Force data . . . . .	17
3.1.3 Physiological data . . . . .	17
3.1.4 Psychological data . . . . .	18

3.2	Sensors . . . . .	22
3.2.1	Robotic Arm . . . . .	22
3.2.2	Force Sensing . . . . .	23
3.2.3	Microsoft Kinect . . . . .	24
3.2.4	Microsoft Band . . . . .	25
3.2.5	Delsys - EMG . . . . .	26
3.2.6	Eye Tracking . . . . .	26
3.2.7	NONIN Onyx . . . . .	27
4.	Upper-limb Kinematics and Dynamics using Kinect . . . . .	28
4.1	Kinect Skeletal Tracking Applications and Limitations . . . . .	28
4.2	Upper-limb Kinematics . . . . .	30
4.2.1	Human Arm Kinematic model . . . . .	31
4.2.2	DH table parameters . . . . .	33
4.2.3	System Overview . . . . .	39
4.2.4	Experimental Motion Results . . . . .	42
4.3	Quantitative Analysis . . . . .	45
4.3.1	Analysis of Results . . . . .	47
4.3.2	Algorithm . . . . .	51
4.4	Joint torque evaluation . . . . .	54
4.4.1	Experimental setup at the shoulder's joint . . . . .	55
4.4.2	Experimental Analysis of the electromyographic data . . . . .	57
5.	Haptic Force-fields and control strategies . . . . .	61
5.1	Related Work . . . . .	61
5.2	Haptic Device Dynamics . . . . .	62
5.3	Haptic Path . . . . .	62
5.3.1	Haptic forces . . . . .	63

5.3.2	Haptic Control . . . . .	65
5.3.3	Haptic Experiments . . . . .	66
5.3.4	Haptic Response . . . . .	67
6.	Artificial Intelligence for adaptive human-robot interaction . . . . .	70
6.1	Self-managed physical therapy . . . . .	70
6.1.1	Theoretical Background . . . . .	71
6.1.2	Motion Analysis . . . . .	75
6.2	Real-time Tele-rehabilitation Robot-based system . . . . .	79
6.2.1	Exercise Analysis . . . . .	80
6.3	An adaptive robot-based therapy . . . . .	83
6.3.1	Performace-based Assessmenet . . . . .	84
6.3.2	Network Communication . . . . .	85
6.4	System Evaluation . . . . .	86
7.	Developed Applications in Human-Robot Interaction . . . . .	89
7.1	Robot-based Tele-rehabilitation system . . . . .	89
7.1.1	Experimental Setup and Results . . . . .	90
7.2	Human-Robot Delivering Interaction in Assistive Environments . . . . .	94
7.2.1	Proposed Framework . . . . .	95
7.2.2	Experimental Evaluation . . . . .	100
7.3	3D Mapping of Visual Attention for Smart Rehabilitation . . . . .	102
7.3.1	Visual 3D heat-maps generation . . . . .	103
7.3.2	Hardware and Software . . . . .	105
7.4	Safe Human-Drone Interaction in Warehouse Environments . . . . .	106
7.4.1	System architecture . . . . .	107
7.4.2	Experiments . . . . .	113



7.5	An Interactive Robot-based Vocational Assessment Game using Lego Assembly Paradigm . . . . .	116
7.5.1	Decision-Making . . . . .	117
7.5.2	Experimental Results . . . . .	120
7.6	Using Virtual Reality to Program Robotic Manipulators . . . . .	123
7.6.1	Experimental Hypothesis and Case Study . . . . .	124
7.6.2	Experimental Results . . . . .	126
7.7	Multimodal Analysis of Serious Games for Cognitive and Physiological Assessment . . . . .	128
7.7.1	Hypotheses . . . . .	128
7.7.2	Experimental Procedure . . . . .	129
7.7.3	Results and Discussion . . . . .	130
8.	Conclusions and Future Work . . . . .	134
Appendix		
A.	Forward and Inverse Kinematics of Barrett Arm . . . . .	137
B.	List of Publications . . . . .	143
REFERENCES . . . . .		148
BIOGRAPHICAL STATEMENT . . . . .		160

## LIST OF ILLUSTRATIONS

Figure	Page
1.1 Physical configuration of our proposed robotic rehabilitation system. . . . .	2
1.2 Proposed architecture for the robotic rehabilitation system. . . . .	3
1.3 Software architecture schema of the proposed system. . . . .	4
2.1 Demands for therapists . . . . .	8
2.2 Upper-limb rehabilitation robot categories. . . . .	12
2.3 Examples of end-effector robots. . . . .	13
2.4 Examples of exoskeleton robots. . . . .	14
3.1 Graphical User Interfaces, Serious Games and Exergames developed for upper-limb robotic rehabilitation applications. . . . .	18
3.2 The left figure depicts the Puck servomotor controller and the right figure illustrates the spherical isometric workspace of the Barrett WAM Arm. . . . .	23
3.3 Barrett Arm F/T sensor . . . . .	24
3.4 Microsoft Kinect 2.0 . . . . .	24
3.5 Microsoft Band 2 . . . . .	25
3.6 Trigno Wireless System. . . . .	26
3.7 User in front of an eye tracker. . . . .	26
3.8 Nonin Pulse and Oximeter device. . . . .	27
4.1 From an single input depth image, a per-pixel body part distribution is inferred. Picture captured by [67]. . . . .	29
4.2 4 DOF Kinematic Model of the Human Arm . . . . .	32

4.3	Anthropomorphic data for the human body segments captured by [21]	39
4.4	Vision-based subsystem overview . . . . .	40
4.5	Captured Elbow and Wrist positions by the Kinect (blue line). Filtered Elbow data (red line) . . . . .	41
4.6	The above images present four frames for a particular exercise, where jerky motions from the kinect is inevitable. Note, that the Kinect doesn't estimate precisely the position of the left arm, because of the same depth information with the rest of the body. . . . .	43
4.7	The top four diagrams show the evolution of the exercise motion for the raw Kinect model. The bottom four diagrams present the corrected motion based on the polynomial fitting motion estimation. . . . .	44
4.8	The image of the left side depicts the polynomial fitting of the estimated joint angles from the IK solver. The fitted solutions are shown in red color, while the raw angles are in blue. The right image shows the velocities of the fitted joint angles . . . . .	45
4.9	Angular accelerations for the four joints . . . . .	46
4.10	Estimated Torques (N/m) for the four joints . . . . .	47
4.11	Physical Experimental Setup: VICON markers are attached in the 3D printed mounts, which are wrapped around the participant's joints. The mounts were placed on the wrist, elbow, shoulder and chest. . . . .	48
4.12	Simple exercise: In each subfigure, the above three images show the motion of the human arm from the Kinect, System and VICON scope. The lower subfigures show the skeleton tracker, the polynomial fitting process of the system and the physical set-up of the exercise. . . . .	49
4.13	Simple Exercise: Distance and velocity comparison . . . . .	50

4.14	Complex exercise: In each subfigure, the above three images show the motion of the human arm from the Kinect, System and VICON scope (Zoom in). The lower subfigures show the skeleton tracker, the polynomial fitting process of the system and the physical set-up of the exercise. . . . .	51
4.15	Complex Exercise: Distance and velocity comparison . . . . .	52
4.16	Muscle to sensor placement . . . . .	55
4.17	Shoulder Abduction . . . . .	56
4.18	Shoulder Forward Flexion . . . . .	57
4.19	Shoulder Abduction . . . . .	58
4.20	Shoulder Forward Flexion . . . . .	59
5.1	Prescribed exercise represented by a haptic path. . . . .	63
5.2	Prescribed exercise represented by a haptic path. . . . .	64
5.3	The impedance haptic path controller implemented in the Barrett WAM robot. A control chart overview. . . . .	65
5.4	Volunteer chronic stroke patient. . . . .	66
5.5	Stroke patient's error deviation for three difference proportional values: A) $P = 50$ ; B) $P = 100$ ; C) $P = 800$ . . . . .	67
5.6	Unimpaired user's error deviation for three difference proportional values: A) $P = 50$ ; B) $P = 100$ ; C) $P = 800$ . . . . .	68
6.1	Human Robot and Game Interaction . . . . .	71
6.2	The procedure of the trajectory classification . . . . .	72
6.3	Hidden Markov Model configuration . . . . .	74
6.4	Projection in the 3 planes . . . . .	76
6.5	Accuracy estimation for different number of classes. . . . .	77
6.6	Accuracy estimation in comparison to the number of segments. . . . .	78

6.7	Trajectory classification accuracy. Comparison results for different number of classes and different number of segments for each trajectory.	79
6.8	Left - Exercise Analysis Flow Diagram; Right - Patient's recovery progress in four sessions. The score error deviation over sessions are A.) 52.38, B.) 25.83, C.) 8.31 and D.)7.53. . . . .	82
6.9	Proposed home-based robotic rehabilitation system. . . . .	84
6.10	Fuzzy Inference System for the performance assessment. . . . .	85
6.11	Input and output membership functions of Fuzzy Inference System. . . . .	86
6.12	Rule-based for Fuzzy Inference System. . . . .	86
6.13	System Architecture. . . . .	87
7.1	Tele-Rehabilitation System. On the right figure the therapist assigns exercises and on the left figure the patient performs them. . . . .	89
7.2	Questions given during survey. . . . .	91
7.3	The mean value of the responses from the survey (The red lines correspond to the therapist interface answers and the blue lines with the patient interface). . . . .	92
7.4	Face detection and recognition framework. . . . .	94
7.5	The task of our proposed framework for secure task execution. . . . .	95
7.6	Application and Kinect feature extraction and matching for person detection. . . . .	96
7.7	Face detection examples captured from the application and from Kinect camera of the robot. . . . .	97
7.8	Experiments for each customer in different layout. . . . .	99
7.9	The person detection classification accuracy. . . . .	100
7.10	Comparison of the identification accuracy of Eigenfaces and Fisherfaces for different numbers of people. . . . .	101

7.11	Head mounted eye-tracker and visual attention pipeline. . . . .	102
7.12	Human recognition and drone’s maneuverability. . . . .	106
7.13	Each straight line has a unique representation in polar-coordinates ( $\rho$ , $\theta$ ). . . . .	109
7.14	Image after the Canny edge detector and have been applied the Hough transform algorithms. . . . .	109
7.15	Lines we are interested after the pruning procedure. . . . .	110
7.16	Representation of the hough lines in the left and right wall. . . . .	112
7.17	Pedestrian detection for secure avoidance proposes as the AR.Drone pass through them. . . . .	113
7.18	Sequence of frames that show the AR.Drone position (red rectangle) when detects a human being in front of it and try to avoid him. . . . .	114
7.19	Compensate the yaw angles of the AR.Drone with the captured images.	115
7.20	Interactive Human-Robot Learning Framework. . . . .	116
7.21	Decision Tree Cross-Validation evaluation from collected labeled train- ing samples over time. . . . .	120
7.22	The decision tree model of the NAO robot’s decision making system. .	121
7.23	Four memory-game steps showing the NAO robot’s feature extraction and decision making results. . . . .	122
7.24	Real-time Human and Virtual Robot manipulation . . . . .	123
7.25	Barrett WAM task design Graphical User Interface. . . . .	124
7.26	Overview of the three different interface systems . . . . .	125
7.27	Average time to perform a task . . . . .	126
7.28	Number of collisions per task . . . . .	127
7.29	Design Serious Games for Human-Robot Assessment and Training. . .	128
7.30	Average Move Delay for L1 and L4. . . . .	132

7.31	Pearson's correlation between Move & Eye Delay. . . . .	133
8.1	Fuzzy Inference System for user' arousal condition. . . . .	134
8.2	ANFIS system for membership function tuning. . . . .	135

## LIST OF TABLES

Table		Page
4.1	DH Table (modified) for 4-DOF Human Arm . . . . .	33
4.2	Correlation values between torque and electromyographic signal. . . . .	59
7.1	Participants Demographics . . . . .	130
7.2	Results of the Serious Games . . . . .	130



## CHAPTER 1

### Introduction

#### 1.1 Motivation

A job or traffic accident, a misfortune event or an unforeseen stroke can lead to brain or musculoskeletal injuries, that impact motor and cognitive functions. Modern physical rehabilitation has proven to be instrumental in the ability to partially or fully heal patients with impaired motor capabilities. During the last two decades, the use of robotic instruments for upper-limb rehabilitation has increased, as they provide an accurate evaluation of motor recovery and automate simple tasks that burden caregivers. However, what is poorly faced in the design of current robotic rehabilitation instruments, is the incorporation of the user in the robot's control loop to be able to adapt the training procedure.

Contemporarily, as the number of people that require physical rehabilitation has increased, there is a growing need for new computational methods to evoke the effectiveness of robot-based therapy, as a complementary rehabilitation method that can accurately quantify, track, and personalize rehabilitation over time. Also, the need has arisen to create low-cost home-based robotic instruments that are simple to use, accessible and provide smart assessment and adaptable training. Finally, artificial intelligence research has reached to the point of constructing intelligent agents that are able to perceive and act independently, emulating the human reasoning system.

## 1.2 Proposed System

In this thesis, a modular intelligent robotic rehabilitation system that can adjust the rehabilitation regimen and monitors the user performance with an engaging, unobtrusive and safe way, is presented. Our work heads towards a multisensing upper-limb robotic rehabilitation system, able to adapt its behavior according to the user's global performances.

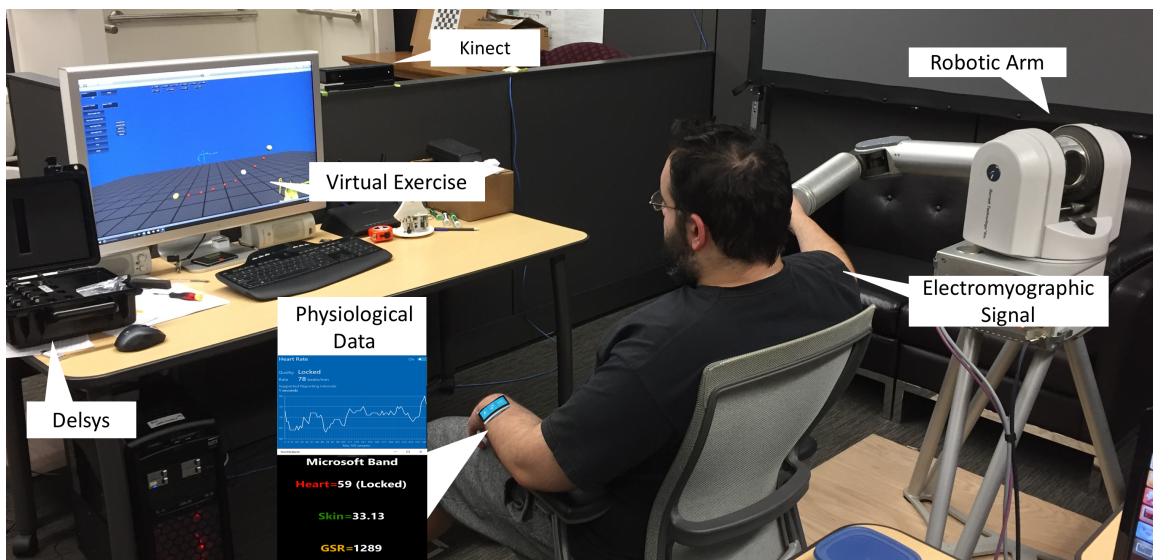


Figure 1.1: Physical configuration of our proposed robotic rehabilitation system.

Figure 1.1 shows the physical configuration of our system, which is expected to provide personalized rehabilitation exercises by adjusting its resistive and supportive behavior according to an intelligence controller. The system will detect the user's physical weaknesses and will interact with the user based on the rehabilitation regimen. The most interesting feature of this system is its potential to emulate a human therapist's reasoning skills and modify treatment routines accordingly using artificial intelligence.

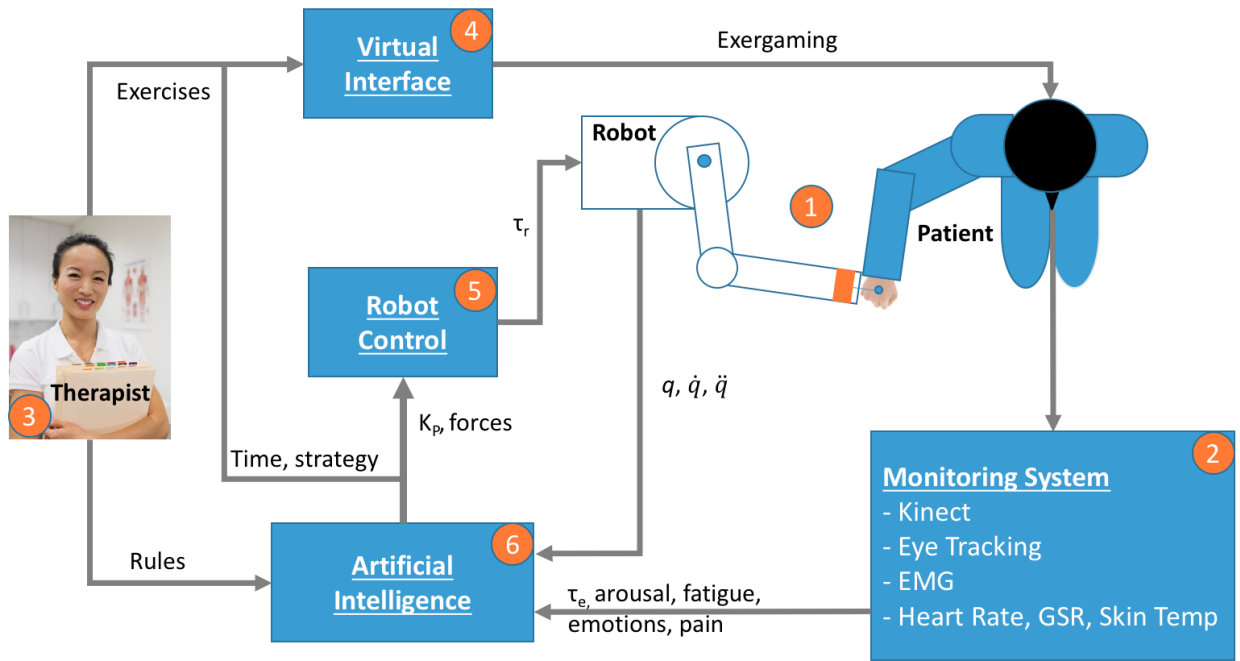


Figure 1.2: Proposed architecture for the robotic rehabilitation system.

Figure 1.2 depicts our modular proposed architecture, which is consisted of the following modules:

1. The human-robot interaction, where the user is attached to the robotic arm and perform exercises.
2. The monitoring system, which extracts the user's physiological and psychological data, and tracks the user arm's and eye's motions.
3. The therapist, who defines the rehabilitation regimen based on abstract rules in a linguistic manner and records exercises in the 3D space (trajectories).
4. The Virtual Interface, which is used for tele-rehabilitation scenario and displays exergames and serious games.
5. The robot control regulates the robot's input torques signal based on the exercise and therapy requirements.

6. The Artificial intelligence acts as a high intelligence system that shifts the robot's gain and resistance during the exercises, and assesses the user weaknesses based on his global performances.
7. Last, a detailed technical description of the evaluation issues, challenges and solutions of the proposed architecture has been simulated and analyzed.

### 1.3 Multimodal Modular System

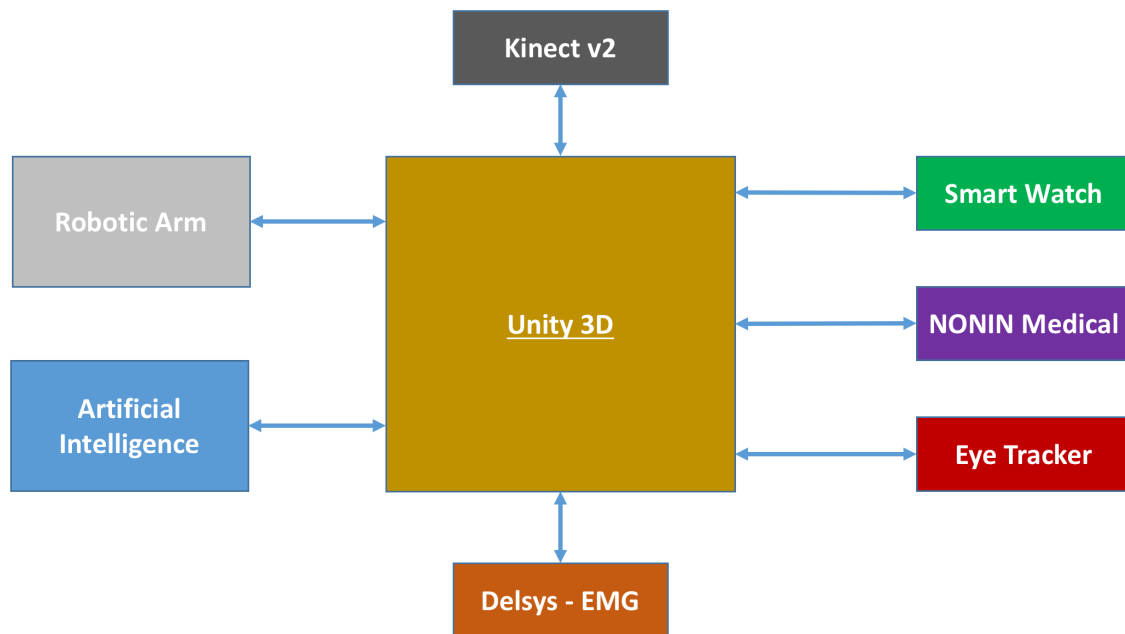


Figure 1.3: Software architecture schema of the proposed system.

This multimodal modular system has been developed integrating the Unity 3D [73] cross platform with the sensors required for the rehabilitation robotic system. Particularly, a robotic platform and five other monitoring sensors have been connected in a multi-class hierarchy using inter-process communication protocols. Such

a modular architecture allows the incorporation as well as the integration of more devices in the future. This proposed architecture was developed integrating three programming languages running on two operating systems (Fig. 1.3). The robotic arm and its haptic controller were executed in the real-time operating system, called Xenomai, which is a Linux machine. The game interface along with the physiological and monitor sensing devices are running in a Windows 10 operating system under C programming language. Finally, the analysis as well as the adaptive controller is deployed in a Matlab environment and is communicating with the Unity interface through local socket integration.

#### 1.4 Dissertation structure

This dissertation thesis is constructed as follows:

- In Chapter 1, we present our proposed intelligent multimodal upper-limb rehabilitation robotic system.
- In Chapter 2, the need for upper-limb rehabilitation is presented and the traditional methodologies for stroke rehabilitation and evaluation are discussed. Also, the state of the art on rehabilitation robotics is analyzed and the main technologies and techniques are presented.
- In Chapter 3, the functional components such as sensors, robotic device and graphical user interfaces for user data acquisition and processing as well as for the extraction of the performance indicators are described.
- In Chapter 4, a reliable method for reconstructing the upper-limb kinematics and dynamics of a patient during end-effector robot-based exercises is presented. The experimental tests with ground truth data and the correlation of the upper-limb dynamics with electromyographic data are reported.

- In Chapter 5, the application of the Haptic, force-fields robot controller for the user hand coordination is demonstrated. Also, different control strategies and protocols for intelligent adaptive robot-based exercises are presented.
- In Chapter 6, Artificial Intelligence techniques such as Machine Learning, Pattern Recognition, and Fuzzy Inference System are utilized in order to assess the user's performances, identify the user's errors and adjust the rehabilitation treatment.
- In Chapter 7, we present seven applications that were developed in part of this thesis in order to study safe human-robot interaction cases and usability user studies.
- In Chapter 8, conclusions and future work are reported.

## 1.5 Research Questions

During our work, four research questions arise with the development of this thesis:

1. Can a multisensing framework assess the user's cognitive and physical performances?
2. Can a vision-based system, such as the Kinect camera, identify the users' wrist position when they interact with external objects, like a robotic arm?
3. Can an end-effector robotic arm derive the user's upper-limb joints' torque?
4. Can a robotic manipulator adapt its control parameters according to the user performance?

## 1.6 Ethical Concerns

It is a well-known fact that human-robot interaction research studies the ways that human can use robots in their life for the different type of applications that are tedious, risky and requires precision. Especially this is a high priority for medical and health care applications. For example, right now Artificial Intelligence diagnosis tumors, patients are taking charge of their health with vital sensors, virtual reality makes possible to treat patients in a continent ways. These disruptive technologies are changing healthcare entirely. However, there are many ethical concerns about the future of health care [64]. Ethicists have raised their concerns about the prospect ways that intelligence, autonomy and robots will take care of elderly in the future. Different questions ask if robots will replace the nurses, if robots could deliver the same quality of attention and even more concerns.

Shortly, the growing number of the elderly population and the existing shortage of therapists and caregivers who assist disabled individuals tends to be increased and will impose serious problems as well as various challenges for the society. This requires a higher number of professional health care staff and more space for medical institutions leading to higher costs in general. It seems inevitable that medical robots, automation, and artificial intelligence will fill the gap by complementing the burden of therapists and be able to execute repetitive tasks with high precision. That does not mean that will replace many jobs in healthcare but will take over some tasks that make healthcare slower and more expensive. Moreover, new incredible opportunities will emerge in medicine if only physicians can acquire new skills and improve the existing once. Many specialists will have more time for patients and better insight into the disease; it's up to each of us to hold our skills and make ourselves irreplaceable in this brave disruptive new world of healthcare.

## CHAPTER 2

### Robot-assisted Rehabilitation

In the near future, the growing number of the elderly population and the existing shortage of therapists and caregivers who assist disabled individuals tends to be increased and will impose serious problems as well as various challenges for the society. This requires a higher number of professional health care staff and more space for health care institutions leading to higher costs in general. The demands for therapist will be bigger and this will lead to big waiting list or burden therapist work.

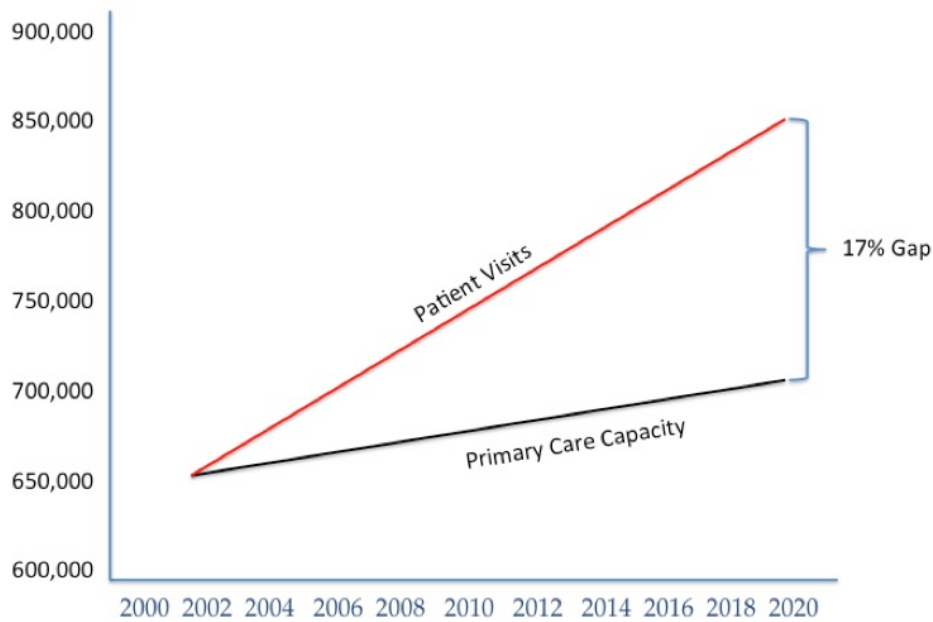


Figure 2.1: Demands for therapists

It is estimated that by 2020 the gap between patient visits and the primary care capacity will be  $159,300 * PCPs$  shorter, as Figure 2.1 shows below. The assistive



robotic devices have the potential to address these problems and can be successfully exploited to augment therapists skills in promoting neuroplasticity and stimulating motor re-learning since they allow administering high-intensity, repetitive, active and task-oriented training.

## 2.1 Stroke Rehabilitation

Stroke is a serious health-care problem observable all around the world [78]. Approximately 16 million people experience a stroke worldwide per year, of which about two-thirds survive. 85 percent of stroke survivors recover partially, and about 35 percent of them suffer from a major disability [37]. The most popular impairment caused by stroke is motor impairment, which can be regarded as a loss of muscle function control, or limitations in limb movements or mobility. [45]

In recent years, a variety of methods have been introduced to deal with motor impairment in the upper-limb; these include the following:

- Electrical Stimulation is used in the treatment of hemiparesis to enhance sensory awareness, strengthen a weakened body part (such as the arm) and improve range of motion [69].
- Cortical stimulation is the electrical stimulation of the brain cortex [32].
- Modified constrain-induced therapy (mCIT) is an intensive motor practice that consists in restricting the use of the unimpaired limb and forcing the patient to use the affected limb [28].
- Motor imagery is the process of imagining the movement of the affected limb. This mental practice activates some motor areas of the brain and some muscles as if the patient is really doing an activity [71].

Traditional motor rehabilitation for stroke patients consists of individual sessions with a physical/occupational therapist that manages repetitive motor exercises

and assists the patient. The interaction between therapist and patient has been classified into four main classes [48]:

- Passive movement, where the patient remains relaxed, while the therapist moves his/her joints. This approach mainly aims at maintaining range of motion at joints and flexibility in muscles and connective tissues.
- Active-assisted movement, where the patient attempts to move a joint or the affected limb, and the therapist assists the patient as needed. This approach is used when the patient cannot complete the desired movement independently.
- Active-Resistive movement, where the patient has to perform a movement with some degree of resistance (gravity, additional weights, elastic band, or the therapist). This approach is used with mild-moderate patients.
- Bilateral movement, where the patient moves the impaired limb imitating the simultaneous movement of the unimpaired limb.

## 2.2 Rehabilitation Robotics

Rehabilitation robotics, or even better can be called, rehabilitation intelligent mechanical training devices that assist the sensorimotor functions (e.g. arm, hand, leg, ankle) of impaired individuals. They are used mainly as therapy tools that boost the therapeutic training and assessment of the patient suffering from motor impairment, especially due to a stroke. The use of robots offers several potential advantages in stroke rehabilitation since it enables:

- high-intensity, repetitive, active and task-oriented training;
- movement constraints' application;
- repeatability and precisely controllable assistance or resistance;
- labour-intensive and patient independent training;

- the acquisition of objective kinematic and dynamic metrics that can be exploited for:
  1. the evaluation of patient residual motor capabilities and therapy outcomes,
  2. the administration of patient-tailored therapies that adapt the assistance to subject motor outcomes,
  3. the investigation on pathology-related planning and motor strategies;
- the use of game scenarios for augmenting patient involvement.

Therefore, robotic technologies can be exploited to augment therapist skills and can be seen as advanced tools available to therapists. Moreover, it must be considered that robotic systems cannot replace those abilities that are typical of a human operator.

In line with the types of patient-therapist interaction, robotic devices can be programmed in different ways for applying forces that guide, stimulate or contrast patient movement [39]:

1. in passive mode, the robotic device moves the arm of the patient along a pre-planned trajectory, while the patient remains relaxed
2. in active-assisted mode, the robotic device provides assistance during patient motion if the patient is not able to complete the movement
3. in active-resisted mode, the robotic device opposes a pre-determined resistance to the movement executed by the patient.

Robots that help patient for upper-limb rehabilitation can be grouped into two categories [47] as Figure 2.2 shows:

1. End-effector robots, where the physical contact with the patient is restricted to the user's wrist (Figure 2.3);
2. Exoskeleton robots, where the patient is wearing the machine and the physical contact is all along the arm or part of it (Figure 2.4).

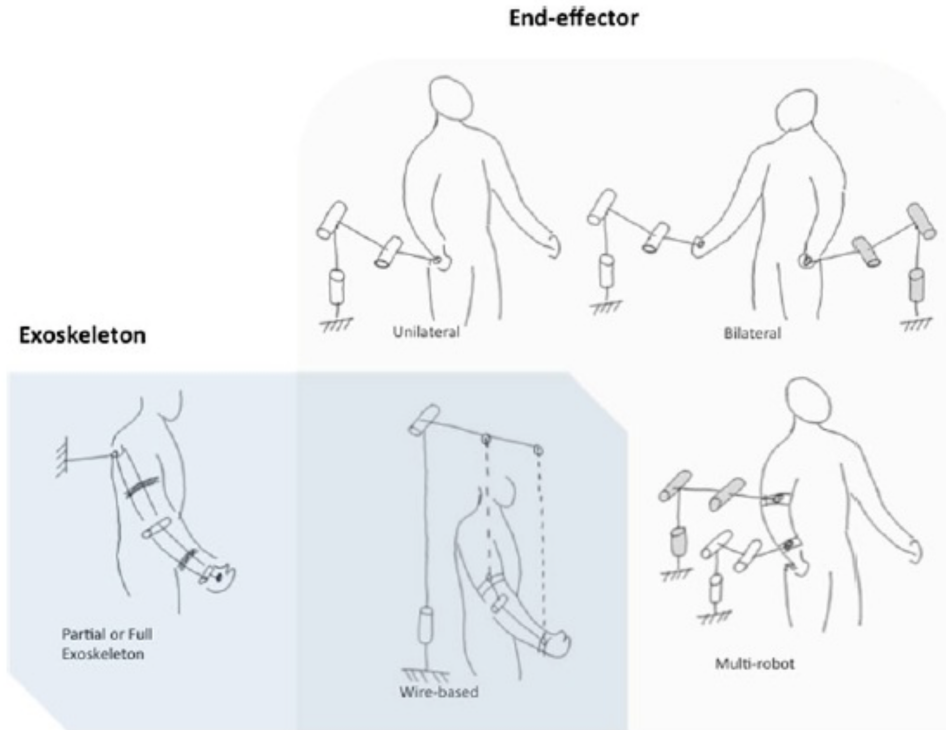


Figure 2.2: Upper-limb rehabilitation robot categories.

Due to their straightforward design, the end-effector robots were the first devices to be utilized in stroke rehabilitation research. On the other hand, the exoskeleton robots enable both accurate measurement and application of torques to specific joints, as well as accurate recording and monitoring of individual joint motion trajectories. Unfortunately, the latter constrains the user's range of motion due to the device configurations and are not simple to use. The former one, can not determine the posture of the upper limb with only one interface. Finally, according to clinical trials, which compared these two types of rehabilitation robots, proved that the end-effector devices offer better arm recovery rather than the exoskeleton arms.



Bi-Manu-Track System



Braccio di ferro



MIT-Manus System



MIME System

Figure 2.3: Examples of end-effector robots.

### 2.3 Robot-based Assessment

The traditional evaluation of stroke motor and functional impairment performed through clinical scales suffers from serious drawbacks, such as therapists' decision which can be subjective. For example, the rehabilitation therapists may change the parameters of the exercise or activities (commonly referred to as grading) between or during treatment sessions, based on confounding patient factors such as pain or fatigue. However, rehabilitation robots are equipped with sensors that can allow accurate measurements of movement kinematics (e.g. limb trajectories) and kinetics (interaction forces). These measures can be fruitfully exploited for quantitatively assessing patient motion performance through opportune indicators that can be combined with clinical scores for a more exhaustive evaluation of the patient.



ARMin System



L-EXOS System



iPAM System



Gentle/S System

Figure 2.4: Examples of exoskeleton robots.

Robot-based performance indicators can be grouped into three categories [4]:

- Kinematic indicators, where they quantify spatial and temporal features of patient movement and are defined either in the Cartesian space or in the arm joint space.
- Kinetic indicators, where they quantify force, work, energy consumption and power related to patient movement.
- Neuromechanical indicators, where they quantify, for example, the viscoelastic properties or the mechanical impedance of the upper limb at rest.

For clarity of presentation, it must be pointed out that this dissertation is essentially focused on point-to-point tasks, in which the patient hand has to move along a line/trajectory between known start and end points.

Several indicators currently used in rehabilitation robotics for quantifying motor performance and each of them is relative to a specific feature of patient motor ability.

- Accuracy is a measure of the difference between the trajectory performed by the patient and the desired path. This kinematic measure can be calculated for any movement for which the desired movement path can be defined.
- Target error is a quantification of patient capability to move in the direction of the target.
- Duration measures the time employed by the patient to perform the task.
- Path length is the measure of the trajectory traveled by the patient to reach the target and is considered a measure of movement efficiency.
- Smoothness is a measure of how gradually a movement is changing. Smoothness is a typical feature of healthy motion; it often lacks in stroke subjects' motion that is characterized by a velocity profile with multiple peaks and deep valleys (arrest phases).
- Movement coordination is the quantification of the spatial and/or temporal coordination between different upper limb joints during multijoint tasks.
- Amount of assistance is a measure that quantifies the patient ability to execute the motor task without the robot assistance.
- Force direction error quantifies the patient ability to exert force in the desired direction (target direction).

## CHAPTER 3

### Data Acquisition

In this chapter, the functional components such as sensors, robotic device and graphical user interfaces for user data acquisition and processing as well as for the extraction of the performance indicators are described.

#### 3.1 Data Acquisition

The proposed system is receiving as input motion, force, physiological and psychological data.

##### 3.1.1 Motion Data

User motion data are acquired by exploiting the sensory system embedded in the robotic machine. In particular, on the basis of the type of employed robotic machine:

- if an anthropomorphic exoskeleton robot is used, hand pose data and arm joint angles are easily provided by the robot;
- if an end-effector machine is employed, only hand motion data are available from the robotic device.

In this second case, a developed method for reconstructing the entire upper-limb kinematics (i.e. shoulder, elbow and wrist position) during robot-aided tasks with end-effector robot is applied and it is presented in Chapter 4.



### 3.1.2 Force data

The sensory system needed to the acquisition of the user and robot interaction forces is either fully or partly embedded in the robotic machine. Therefore, depending on the type of robot used (exoskeleton versus end-effector robot) and/or the type of sensors embedded in the robotic machine, it has been chosen to exploit:

- the robot force sensors at the end-point level;
- the robot torque sensors at the joint level.

In order to obtain the interaction forces in the Cartesian space, the robot statics relation is applied, i.e.:

$$\vec{\tau} = J^T(\vec{q})\vec{F} \quad (3.1)$$

where  $\vec{\tau}$  represents the joint torque vector and  $\vec{F}$  is the corresponding force/torque vector in the Cartesian space which can be represented as:

$$\vec{F} = \begin{bmatrix} f_c \\ t_c \end{bmatrix} \quad (3.2)$$

where  $f_c = [f_x, f_y, f_z]^T$  is the Cartesian forces and  $t_c = [t_x, t_y, t_z]^T$  is the Cartesian torques.

### 3.1.3 Physiological data

In order to evaluate the upper-limb muscular force, power and fatigue of user undergoing robot-aided therapy, the surface electromyographic (EMG) activity of couples of agonist-antagonist muscles of the upper-limb has been considered. The raw EMG recording contains very important information that only a targeted signal analysis can extract. Therefore, in order to extract those signal features that are related to muscular performance, a carefully processing procedure has been designed.

In particular, it has been chosen to consider signal information related to both the time domain and the spectrum domain and, hence, two separated processing procedure have been applied. Moreover, physiological sensors receive heart rate, skin temperature, galvanic skin response (GSR) using the Microsoft Band 2 device as we will elaborate more in the next section.

### 3.1.4 Psychological data

The section below covers all the Graphical User Interfaces (GUIs), the Serious Games and Exergames that have been developed for cognitive and physical test studies, during the integration of the proposed multimodal upper-limb rehabilitation system.

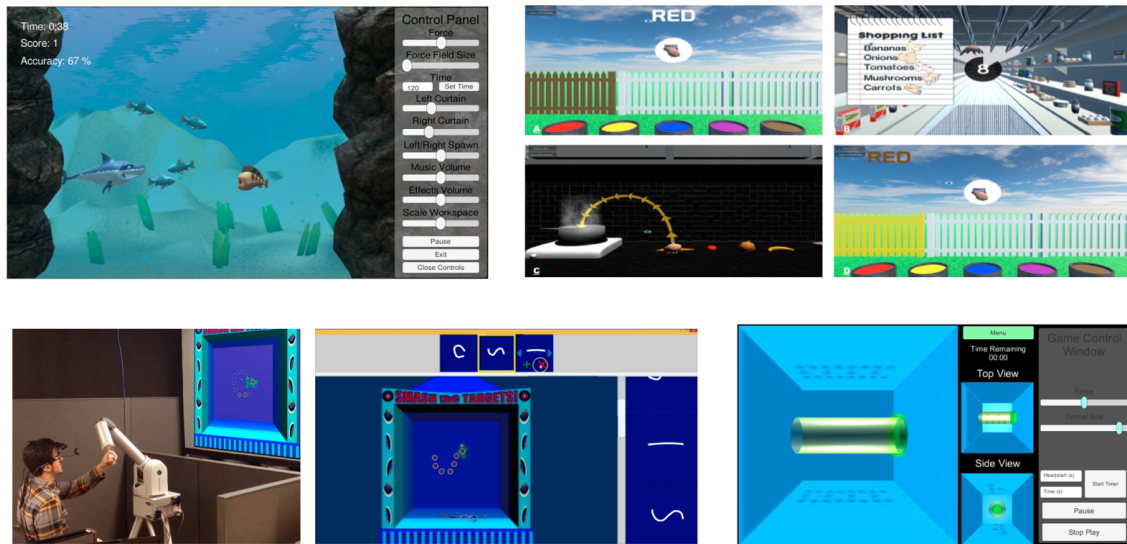


Figure 3.1: Graphical User Interfaces, Serious Games and Exergames developed for upper-limb robotic rehabilitation applications.

#### 3.1.4.1 Graphical User Interfaces for Medical Applications

A Tele-rehabilitation framework that enables interaction between therapists and patients and is a combination of a graphical user interface and a high dexterous robotic arm is presented. Our game consists of moving one hand to hit a sequence of targets that are placed along an exercise gesture path. To make this more interesting, we borrow the aesthetics from traditional ball-toss carnival games. In these games, you toss a ball at a series of targets (cans, round target, or other visually intriguing objects) in an attempt to knock them down or break them. This is nearly identical to the actions we need the patients to complete. Additionally, as there are a wide variety of visually distinct carnival games with similar goals, we can easily alter the visuals of our virtual carnival game while keeping our gameplay the same. This provides more visual and audio variety to the user keeping them interested for longer.

To provide a large degree of adaptability for therapy sessions, our system utilizes two screens at the same time. One screen is for the patient and displays only the game. This screen may be visible by the therapist depending on if he is in the same location/facility as the patient. The second screen allows the therapist to adjust the exercise program in real time by drag and drop of the exercise trajectories in the exercise list on the right side of the screen. Running along the top of the therapist's screen is a horizontally scrollable section that visually shows the sequence of exercises for this session as a series of pictures, placed sequentially in the order that they have been or will be completed. The exercises that have been completed are located on the left side of the list. The exercise currently being performed by the patient is after the completed exercises and is indicated by a yellow border. The exercises that have yet to be completed are located on the right side of the list, after the current exercise. These uncompleted exercises are changeable and can be reorganized at will.

#### 3.1.4.2 Cognitive and physical therapy

Cognitive therapy is generally used to assess and retrain specific brain functions that may have been impaired due to physical injuries, congenital abnormalities, or degenerative diseases. Examples of such cases are strokes, traumatic brain injuries (TBI), cerebral palsy (CP), and Alzheimer’s disease; all of which share similar symptoms of depression, cognitive impairment, social difficulties and often communication-related issues [68][70][79]. Over the past few decades, computer-based cognitive therapy has been developed and employed to aid in the rehabilitation process [80]. In a previous experiment [80], a computer-aided program was implemented to alleviate symptoms of certain mental health issues, such as depression and anxiety. In addition to computer-aided cognitive therapy, there are many types of technology-assisted physical therapy. Recent examples include robot-aided rehabilitation, which is commonly used for the repair of upper-limb movement following a stroke [40]. Furthermore, extensive use of virtual reality games involves flight simulators or physical coordination tasks, such as grasping an object and placing it into a basket [57]. Traditionally, cognitive and physical therapy have been separated and often counterproductive [1]. Multiple doctors often disagree with the medical plan of action and the treatment centers are usually found in different locations, which causes more distress for the individual. As the literature concerning the use of serious games in rehabilitation develops, a consideration for the combination of both cognitive and physical therapy in the same session should be addressed [40].

#### 3.1.4.3 Eye-hand coordination

Eye-hand coordination is a combined physical and cognitive behavior that is part of performing everyday tasks, and it has been studied in activities involving

wooden block sets, archery, sporting performance, computing-games, and even tea-making [36]. In its absence, most people would be unable to carry out even the simplest of actions like picking up a book from a table or playing a video game. Clinical assessment tools employ this control mechanism to assess the sensorimotor impairment of stroke patients and to evaluate their recovery performance before and after rehabilitation treatment [40]. Additionally, in Alzheimer patients, the timing of eye-hand coordination tasks has been studied for its use of examining neuro-degeneration over time [19]. Hereby, the evaluation of the eye-hand coordination makes it important to identify user's behavior and agility. Traditionally, cognitive and physical therapy have been separated and often counterproductive [1]. Multiple doctors often disagree with the medical plan of action and the treatment centers are usually found in different locations, which causes more distress for the individual. As the literature concerning the use of serious games in rehabilitation develops, a consideration for the combination of both cognitive and physical therapy in the same session should be addressed [40].

#### 3.1.4.4 Reaction and activity response time

Recent research has found that the combination of both cognitive and physical exercises significantly increased reaction time in elderly populations [38]. This suggests that the mind and body are connected through multiple facets. There are two important response times that can be measured while using virtual reality in rehabilitation. When combined together, these response times could aid in both cognitive and physical rehabilitation. The first time metric is the reaction time, which refers to the period that the user needs to respond to the stimulus once presented. This time is measured using the eye-tracker, and it is a measure of the user alertness. This kind of time information is important and can be used to monitor the progress of

users with cognitive impairment, especially when increasing the difficulty of a task. For example, people with closed head injury (CHI) show a decrease in vigilance while performing complex tasks [46]. The second time metric is the activity time, which is the time needed to complete the task once the visual cues are recognized. This activity time can be used in addition to the inertial measurement sensors embedded in the robotic manipulator to evaluate the progress of a physically impaired user. This metric can be applied to people suffering from a stroke since they require a longer time to accomplish active tasks [23]. When combined together, these response times could aid in both cognitive and physical rehabilitation.

#### 3.1.4.5 Exergames

Exergames are the video games that are used exercising purposes. These games utilize technology to track the user's body parts, like the Kinect camera. According to researchers, exergames provide real improvement to the users' fitness, and they aim to make the video gaming experience more fun.

### 3.2 Sensors

This section lists all the sensors and robotic equipments that have been utilized for our proposed system.

#### 3.2.1 Robotic Arm

Our work takes advantage of the advanced capabilities of the Barrett Arm in dynamic adaptation, force-feedback, and torque sensing, in order to deliver a safe, computer-guided physical therapy regimen. The Barrett Whole Arm Manipulator (WAM) Arm is a highly dexterous backdrivable manipulator. It is the only commercially available robotic arm with direct-drive capability supported by Transparent

Dynamics between the motors and joints, so its joint-torque control is unmatched and guaranteed stable. It is built to outperform today's conventional robots by offering extraordinary dexterity, zero backlash, near-zero friction and spherical isometric workspace approximately 2 meters in diameters (Figure 3.2). Finally, it is equipped with the world's smallest servomotor controller "Puck" that offers incredibly low power consumption and ultra-high brushless-servo performance enabling application such as force-field enabled medical surgery.

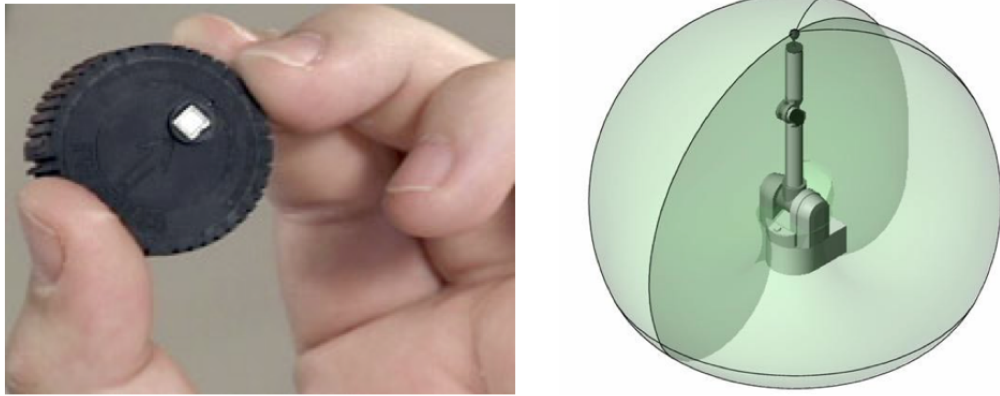


Figure 3.2: The left figure depicts the Puck servomotor controller and the right figure illustrates the spherical isometric workspace of the Barrett WAM Arm.

### 3.2.2 Force Sensing

The Barrett 6-Axis Force/Torque Sensor is a completely self-contained sensing package that expands the force sensing capability of WAM and BarrettHand systems. The Force/Torque Sensor is available as an option when purchasing a WAM Arm or BarrettHand. The base of the sensor attaches to the tool plate of the WAM Arm (4-DOF and 7-DOF)\* and the top plate of the sensor attaches to the base of the BarrettHand 28X series.

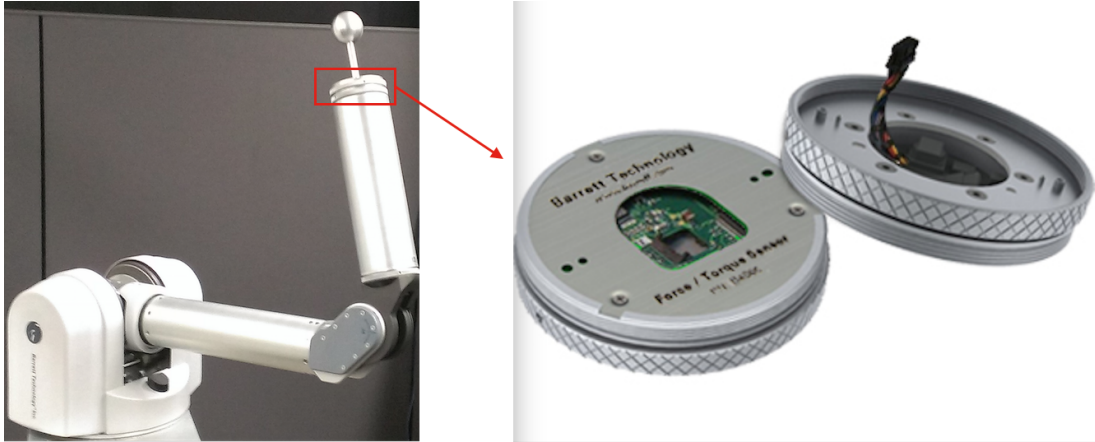


Figure 3.3: Barrett Arm F/T sensor

As with the Barrett Puck, the Force/Torque Sensor is an entirely self-contained unit. There is no need for an external cable or large controller unit; all of the necessary amplification and processing electronics are built into one of the industry's thinnest packages. With the calibration data preloaded into the sensor's non-volatile memory, the onboard 32-bit DSP processes the signal from all the strain gages and outputs 3 forces and 3 torques. All that is needed to start taking measurements are Barrett's standard internal 48-V power and CANbus communications.

### 3.2.3 Microsoft Kinect



Figure 3.4: Microsoft Kinect 2.0

The Microsoft Kinect v2 is a unique device produced essentially for gesture recognition and is a more refined version of Kinect v1. PrimeSensor device is in its base [63] and in addition to VGA resolution video ( $1920 \times 480$  pixels), it can also capture depth images ( $512 \times 424$ ) resolution. A laser, and infrared camera



and the structured light beams are used for capturing depth informations. Finally, The skeletal tracker of Kinect v2 truly outperforms its previous version. It tracks 6 people instead of 3 and 26 joints rather than 20 of Kinect v1. The utilization of the Kinect’s skeletal tracking data will be important in our unobtrusive upper-limb kinematic model reconstruction and torque data validation.

### 3.2.4 Microsoft Band

The Microsoft Band was a smart band that was capturing the activity features of the user. The ten sensors (i.e. optical heart rate monitor, three-axis accelerometer gyrometer, GPS, microphone, ambient light sensor, galvanic skin response sensors, UV sensor, skin temperature sensor, and capacitive sensor) that were incorporated into the band, were able to produce fitness tracking and suggest health plans. This band will be used by our proposed system in order to incorporate user’s physiological data into the closed-loop human-robot control architecture.



Figure 3.5: Microsoft Band 2

### 3.2.5 Delsys - EMG



Figure 3.6: Trigno Wireless System.

The *Trigno<sup>TM</sup>* Wireless EMG System has been designed to detect the muscle contraction using the EMG signals. The system provides real-time data streaming from the EMG sensors integrating 16 EMG and 48 accelerometer analog channels. In our proposed system, the EMG signals will verify the correlation between the torque values ex-

tracted by our vision-based kinematic mechanism and the real muscle contraction data.

### 3.2.6 Eye Tracking

Eye-tracking is a technology that calculates the user's eye gaze while he is interacting with a computer monitor. Such a device enables the user to interact with their eyes instead of using mouses or keyboards. Moreover, eye trackers collect information about the user's gaze data and also the pupil dilation. In this research, we used the eye tracking data information to get users' behavioral signature while they were interacting with our serious games as well as to develop methods for human-robot interactions. The Eye Tribe Tracker was utilized for the first

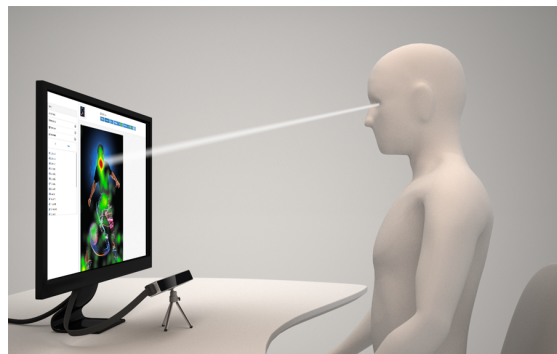


Figure 3.7: User in front of an eye tracker.

set of experiments (Figure 3.7) and for the second set Dr. Christopher Mcmurrough developed an head-mounted eye tracker.

### 3.2.7 NONIN Onyx



Nonin Medical provides a finger pulse oximetry device with scientifically proven accuracy in challenging situations such as patients with small dark skin tones. The device that we used in our research was the Onyx 9500 (Fig. 3.8) and managed to provide data qualification regarding the data of the Microsoft Band 2.0.

Figure 3.8: Nonin Pulse and Oximeter device.

## CHAPTER 4

### Upper-limb Kinematics and Dynamics using Kinect

This chapter introduces a unique design for an unobtrusive reconstruction of the upper-limb kinematics and dynamics of users during rehabilitation tasks with end-effector robotic arm.

#### 4.1 Kinect Skeletal Tracking Applications and Limitations

Shotton et al. [67], describe a system that assigns a discrete label  $w = \{1, \dots, 31\}$ , indicating which of 31 body parts is present at each pixel based on a depth image  $x$ . The resulting distribution of labels is an intermediate representation in a system that proposes a possible configuration of the 3D joint positions in the Microsoft Kinect gaming system (Figure 4.1). The classification was based on a forest of decision trees: the final probability  $P_r(w|x)$  is an average (i.e., a mixture) of the predictions from a number of different classification trees. The goal is to mitigate against biases introduced by the greedy method with which a single tree is trained.

Within each tree, the decision about which branch a data point travels down is based on the difference in measured depths at two points, each of which is spatially offset from the current pixel. The offsets are inversely scaled by the distance to the pixel itself, which ensures that they address the same relative positions on the body when the person moves closer or further away to the depth camera. The system was trained from a very large data set of 900,000 depth images which were synthesized based on motion capture data and consisted of three trees of depth 20. Remarkably,

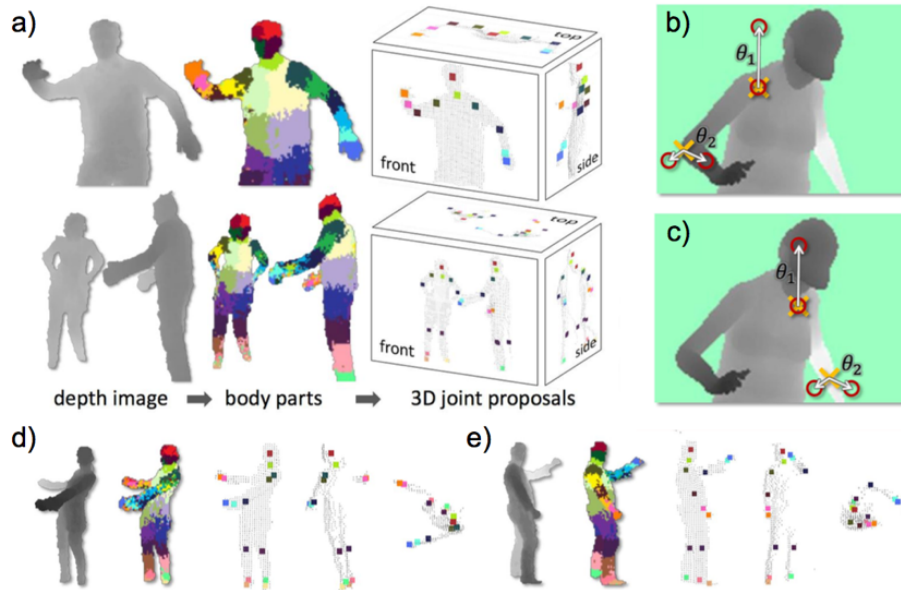


Figure 4.1: From an single input depth image, a per-pixel body part distribution is inferred. Picture captured by [67].

the system is capable of assigning the correct label 59%, of the time and this provides a very solid basis for the subsequent joint proposals.

Kinect recognizes human motions utilizing research ideas from machine learning and computer vision. One of the major challenges of this technology is noiseless as well as the incompleteness of some human postures which is caused from the occlusions of some body parts with other body parts or external objects. For that particular reason, some of the Kinect's application requires the users to face direct the sensor which is not possible real life scenarios. To solve this problem, we came up with user's upper limb kinematic model that can predict the user motion when the are occlusion with an external object, such as a robotic manipulator is our care.

## 4.2 Upper-limb Kinematics

A novel system that can demonstrate the potential to track and estimate the torques that affect the human arm of an individual that performs rehabilitation exercises with the use of Kinect v2, is described. The system focuses on eliminating the jerky motions captured by the Kinect with the incorporation of robotic mechanics methodologies that have been applied in the field of robotic mechanical design. In order to achieve this results, the system takes full advantage of the dynamic and kinematic formulas that describe the motion rigid bodies. Lastly, a simulation experiment is depicted to demonstrate the results of the system.

As adaptive rehabilitation and personalized therapy becomes increasingly essential for the reintegration of traumatized individuals to society, the need has arise to create accurate and safe motion analysis systems that doesn't rely on wearable sensors. To be more elaborative, with the data collected from the Kinect, the system provides a precise estimation of the motion parameters (position, velocity and acceleration) and torques that affect the shoulder and elbow of the patient. The system emphasizes the use of the Microsoft Kinect v2 as opposed to other systems that require a plethora of different sensors such as embedded accelerometers, EMGs or even wearable exoskeleton arms [22] [76] and mechanical manipulators [49].

A considerable amount of research has been conducted in the fields of computer vision, human-computer and human-robot interaction to track the human body. Whereas for entertainment or rehabilitation applications [2][30][43], each of the above fields utilizes different technologies and diverse methodologies to track the state of human body[82]. In our work, we combined a variety of these techniques to track and calculate the motion parameters and torques that affect the shoulder and elbow of an individual who performs rehabilitation exercises. Specifically, research in the area of human exoskeletons shows that the human arm can be mathematically rep-

resented as a kinematic chain [59][60] of seven degrees of freedom (DOF), much like a mechanical manipulator. By making this assumption, we can express the relationship of the human joint's rotation and translation in relevance to a world frame by using the Denavit-Hartenberg (DH) parameters [15]. Moreover, we can derive the forward and inverse kinematic equations of the human arm to obtain a relationship between the position and orientation of the end effector (wrist) with the rotation of the joints. These set of equations are extremely useful, because we can calculate the rotational position, velocity and acceleration of the joints over a specific trajectory. This makes the estimation of the torques that affect the human arm feasible with the use of the (RNE) method, which is a method commonly mentioned in the relevant bibliography [26] [74].

In comparison to other studies, to obtain the necessary positions for our calculations we have considered an alternative marker-less and low-cost solution [67]. Our system takes advantage of the Kinect v2 skeleton tracking algorithm to track the position of the wrist. Unfortunately, due to the probabilistic nature of the Kinect skeleton tracker, the positions are collected with certain inaccuracies under specific circumstances [77]. For this reason, the system eliminates jerky data obtained from the Kinect, though a polynomial fitting process in the joint space that is derived from the inverse kinematic equations of our human kinematic model. Thus, the proposed system utilizes techniques in computer vision and robotic mechanics to solve the human arm tracking problem.

#### 4.2.1 Human Arm Kinematic model

In this section, we provide a thorough analysis of the biomechanical model that the proposed system utilizes to track the human arm. As mentioned above, the human arm can be represented as a kinematic chain, much like a robotic arm. Since

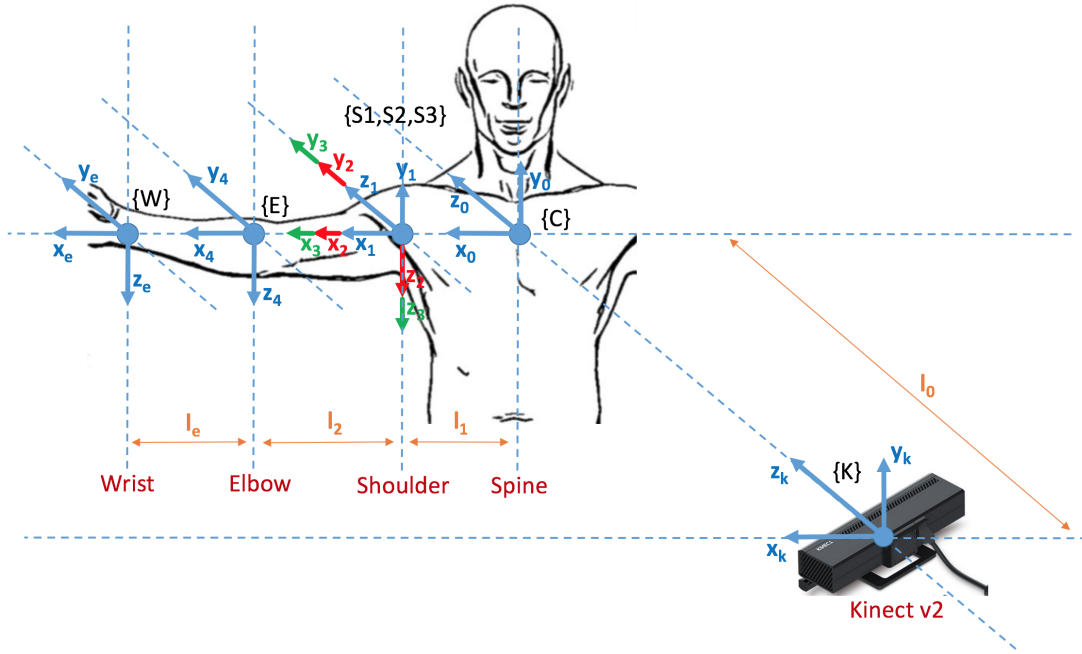


Figure 4.2: 4 DOF Kinematic Model of the Human Arm

the system focuses in the behavior of the shoulder and elbow, we designed a 4 DOF kinematic chain to express the shoulder glenohumeral rotation and the elbow flexion. Figure 1 above, provides a graphical illustration of the developed human arm model.

A summary of the frames description can be seen in Table 1. The kinematic chain begins from frame  $\{K\}$ , which acts as the world frame of the model. Notice that frame  $\{K\}$  denotes where the Kinect is stationed, meaning that when we derive the forward and inverse kinematic equations of the model, all the frame positions are related to the Kinect directly. Next, frame  $\{C\}$  represents the chest of the human who is positioned  $l_0$  meters along the  $z_k$  axis of the Kinect. As expected, frames  $\{S1\}$ ,  $\{S2\}$  and  $\{S3\}$  describe the glenohumeral rotation of the shoulder. Frame  $\{S1\}$  and  $\{S2\}$  rotate around the axis  $z_1$  and  $z_2$  as shown in Figure 1. Note, that to avoid the formation of an Euler gimlock in the shoulder, which would make the solution of the inverse kinematics extremely complicated, frame  $S3$  rotates around the axis



$x_3$  [15]. Frame {E} follows by representing the flexion of the elbow around  $z_4$  axis. Finally, the position of the end effector or wrist {W} is located along the axis  $x_4$  of the elbow frame.

#### 4.2.2 DH table parameters

Before we formulate the forward and inverse kinematic equations of the human arm kinematic chain, we have to derive the DH parameters, which provide a relation between the frames of the model. Table 2 depicts the modified DH parameters as described in the previous section and in Figure1.

Table 4.1: DH Table (modified) for 4-DOF Human Arm

i	$\alpha_{i-1}$	$a_i$	$d_i$	$\theta_i$
0	0	0	$l_0$	0
1	0	$l_1$	0	$\theta_1$
2	90	0	0	$\theta_2$
3	$\theta_3$	0	0	0
4	0	$l_2$	0	$\theta_4$
e	0	$l_3$	0	0

##### 4.2.2.1 Forward Kinematics

Based on the modified DH table (Table 2) we can determine the rotation and translation of frame  $i - 1$  to  $i$  according to the following matrix :

$${}^{i-1}T_i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & \alpha_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

$${}^k_e T = {}^k_0 T * {}^0_1 T * {}^1_2 T * {}^2_3 T * {}^3_4 T * {}^4_e T \quad (4.2)$$

In the above equations,  $c\theta$  stands for the cosine of  $\theta$  and  $s\theta$  stands for the sine of  $\theta$ . Based on the multiplication that are shown above the general transformation from the Kinect frame to the human's wrist is:

$${}^k_e T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & l_e r_{11} + l_2 c_1 c_2 + l_1 \\ r_{21} & r_{22} & r_{23} & l_e r_{21} + l_2 s_1 c_2 \\ r_{31} & r_{32} & r_{33} & l_e r_{31} + l_2 s_2 + l_0 \end{bmatrix} \quad (4.3)$$

$$\left\{ \begin{array}{l} r_{11} = c_1 c_2 c_4 - c_1 s_2 c_3 s_4 + s_1 s_3 s_4 \\ r_{12} = -c_1 c_2 s_4 - c_1 s_2 c_3 c_4 + s_1 s_3 c_4 \\ r_{13} = c_1 s_2 s_3 + s_1 c_3 \\ r_{21} = s_1 c_2 c_4 - s_2 s_2 c_3 s_4 - c_1 s_3 s_4 \\ r_{22} = -s_1 c_2 s_4 - s_1 s_2 c_3 c_4 - c_1 s_3 c_4 \\ r_{23} = s_1 s_2 s_3 - c_1 c_3 \\ r_{31} = s_2 c_4 + c_2 c_3 s_4 \\ r_{32} = -s_2 s_4 + c_2 c_3 c_4 \\ r_{33} = -c_2 s_3 \end{array} \right. \quad (4.4)$$

#### 4.2.2.2 Inverse Kinematics

For the derivation of the inverse kinematic equations we consider the positions of the wrist and elbow as the known variables and the joint angles as unknown variables. Traditionally, in robotics the orientation and position of the end effector is the only known. However, in this particular case, instead of using the wrist's orientation, we

also use the position of the elbow to find the joint angles, because we can directly obtain it from the Kinect.

$${}^k_e T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x_e \\ r_{21} & r_{22} & r_{23} & y_e \\ r_{31} & r_{32} & r_{33} & z_e \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

$${}^k_4 T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x_4 \\ r_{21} & r_{22} & r_{23} & y_4 \\ r_{31} & r_{32} & r_{33} & z_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.6)$$

Position of Elbow:

$$\begin{cases} x_4 = l_2 * c_1 * c_2 + l_1 \\ y_4 = l_2 * s_1 * c_2 \\ z_4 = l_2 * s_2 + l_0 \end{cases} \quad (4.7)$$

Position of Wrist (End Effector):

$$\begin{cases} x_e = l_3 * (c_1 * c_2 * c_4 - c_1 * s_2 * c_3 * s_4 + \\ s_1 * s_3 * s_4) + l_2 * c_1 * c_2 + l_1 \\ y_e = l_3 * (s_1 * c_2 * c_4 - s_1 * s_2 * c_3 * s_4 - \\ c_1 * s_3 * s_4) + l_2 * s_1 * c_2 \\ z_e = l_3 * (s_2 * c_4 + c_2 * c_3 * s_4) + l_2 * s_2 + l_0 \end{cases} \quad (4.8)$$

Angles Derivation  $(\theta_1, \theta_2, \theta_3, \theta_4)$ :

$$\begin{cases} s_2 = \frac{z_4 - l_0}{l_2} \\ c_2 = \pm \sqrt{(1 - s_2^2)} \\ \theta_2 = \text{atan2}(s_2, c_2) \end{cases} \quad (4.9)$$

$$\begin{cases} s_1 = \frac{y_4}{l_2 * c_2} \\ c_1 = \pm \sqrt{(1 - s_1^2)} \\ \theta_1 = \text{atan2}(s_1, c_1) \end{cases} \quad (4.10)$$

$$\begin{cases} c_4 = \frac{l_2^2 + l_3^2 + \sqrt{((x_e - x_1)^2 + (y_e - y_1)^2 + (z_e - z_1)^2)}}{2 * l_2 * l_3} \\ s_4 = \pm \sqrt{(1 - c_4^2)} \\ \theta_4 = \text{atan2}(s_4, c_4) \end{cases} \quad (4.11)$$

$$\begin{cases} c_3 = \frac{z_e - z_4}{l_3} \\ s_3 = \pm \sqrt{(1 - c_3^2)} \\ \theta_3 = \text{atan2}(s_3, c_3) \end{cases} \quad (4.12)$$

#### 4.2.2.3 Recursive Newton Euler (RNE)

In robotics dynamics, the Recursive Newton Euler method is often used to solve the inverse dynamics problem. Specifically, it is used to provide an estimation of the torques that affect the robot's joints given the angles, velocities and accelerations of the joints [15][24]. However, the RNE method can be applied to every kinematic chain, as long as the motion parameters and the mass of the links are known. In our case, since we are measuring the torques that affect the human arm, we estimated the mass of the human upper limbs by taking into consideration the relative relation between

the human weight and the weight of the human arm. [21] Figure 2 in the next page depicts the analogies of anthropomorphic data that were taken into consideration in the scope of this study.

The RNE method is divided into two steps as seen below. In the first step (Outward iteration), the method calculates the relative angular and linear motion parameters from the starting joint to the end effector. The second step (Inwards iteration) iterates backwards and provides an estimation of the moments and torques that affect the joints.

#### 4.2.2.4 Outward iteration

$$0 \rightarrow 3$$

$\omega$  : relative angular velocity of joints

$\dot{\theta}$  : joint angular velocity

$\ddot{\theta}$  : joint angular acceleration

$\hat{Z}$  : the axis of rotation

$P$  : matrix indicating the direction of the center of mass

$m$  : mass of link

$c$  : center of mass

$F$  : linear Force applied in the center of mass

$N$  : moments

$$\left\{ \begin{array}{l}
{}^{i+1}\omega_{i+1} = {}^{i+1}_i R * {}^i\omega_i + \dot{\theta}_{i+1} * {}^{i+1}\hat{Z}_{i+1} \\
{}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}_i R * {}^i\dot{\omega}_i + {}^{i+1}_i R^i * \omega_i \times \dot{\theta}_{i+1} * {}^{i+1}\hat{Z}_{i+1} + \\
\ddot{\theta}_{i+1} * {}^{i+1}\hat{Z}_{i+1} \\
{}^{i+1}\dot{v}_{i+1} = {}^{i+1}_i R ({}^i\dot{\omega} \times {}^i P_{i+1} + {}^i\omega_i \times ({}^i\omega_i \times {}^i P_{i+1}) + {}^i\dot{v}_i) \\
{}^{i+1}\dot{v}_{C_{i+1}} = {}^{i+1}\dot{\omega}_{i+1} \times {}^{i+1}P_{C_{i+1}} + {}^{i+1}\omega_{i+1} \times ({}^{i+1}\omega_{i+1} \times \\
{}^{i+1}P_{C_{i+1}}) + {}^{i+1}\dot{v}_{i+1} \\
{}^{i+1}F_{i+1} = m_{i+1} * {}^{i+1}\dot{v}_{C_{i+1}} \\
{}^{i+1}N_{i+1} = {}^{C_{i+1}}I_{i+1} * {}^{i+1}\dot{\omega}_{i+1} + {}^{i+1}\omega_{i+1} \times {}^{C_{i+1}}I_{i+1} * \\
{}^{i+1}\omega_{i+1}
\end{array} \right. \quad (4.13)$$

#### 4.2.2.5 Inward iteration

$f$  : Force Propagation

$\tau$  : Torque applied to the joints

$n$  : Accumulative torque applied to the joints

$4 \rightarrow 1$

$$\left\{ \begin{array}{l}
{}^i f_i = {}_{i+1}^i R * {}^{i+1} f_{i+1} + {}^i F_i \\
{}^i n_i = {}^i N_i + {}_{i+1}^i R * {}^{i+1} n_{i+1} + {}^i P_{C_i} \times {}^i F_i + {}^i P_{i+1} \times \\
{}_{i+1}^i R * {}^{i+1} f_{i+1} \\
\tau_i = {}^i n_i^T * {}^i \hat{Z}_i
\end{array} \right. \quad (4.14)$$

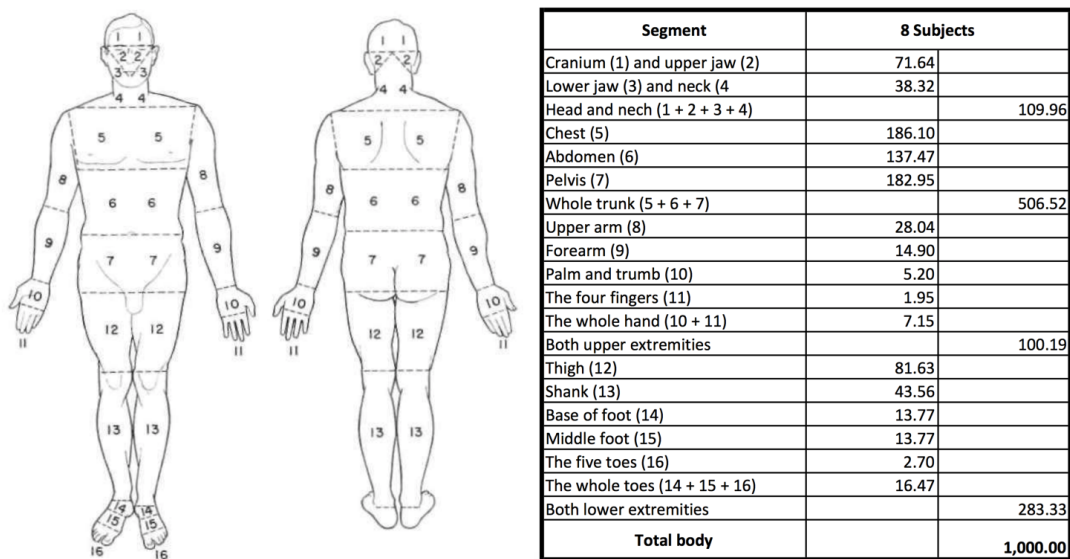


Figure 4.3: Anthropomorphic data for the human body segments captured by [21]

#### 4.2.3 System Overview

A block diagram of the proposed subsystem can be seen in Figure 4.4. The diagram summarizes the majority of the processes that compose the overall system. As an input to the system, the user must capture the person who is performing the exercise with the Kinect, according to the configuration that Figure 1 suggests. Once the trajectory of the subject's arm has been captured, the Kinect passes the cartesian positions of the chest, shoulder, elbow and wrist frames to the first unit of the system that reconstruct raw model. The raw model provides an abstract illustration of the physical configuration that was captured by the Kinect.

The system then applies a median filter to the raw data to eliminate any abnormal behavior from the skeleton tracking algorithm of the Kinect. This result of this module produces a smooth trajectory that is used by the system's Inverse Kinematics Solver (IK Solver) to provide an estimation of the angles of the human arm joints. Note that this unit utilized the inverse kinematics equations that were described in

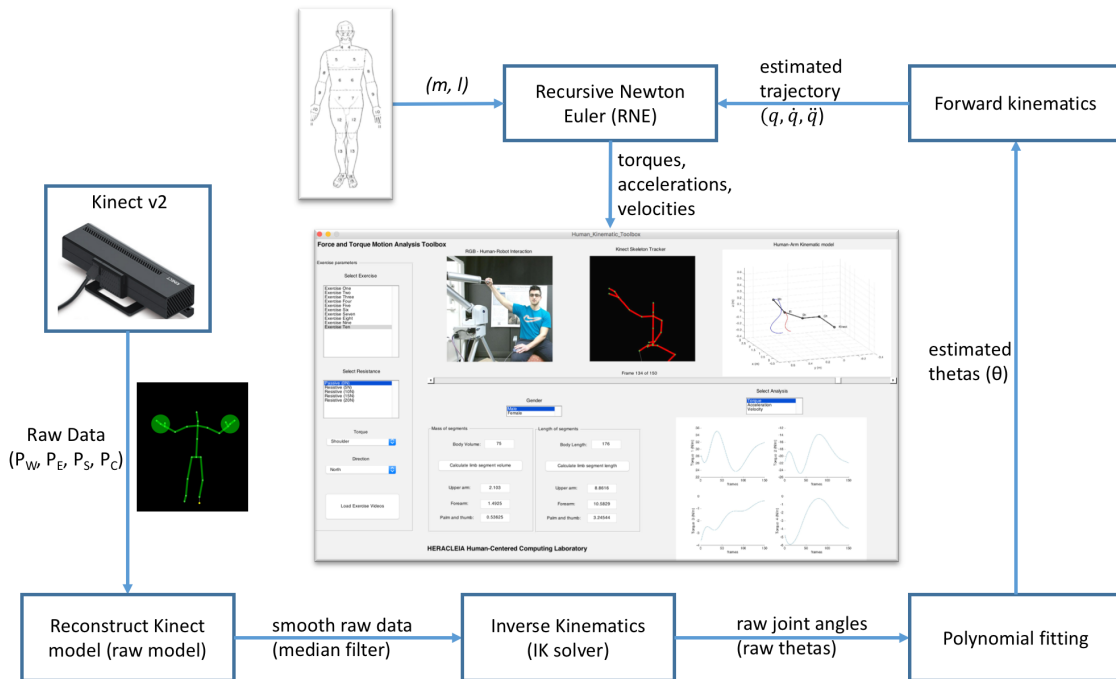


Figure 4.4: Vision-based subsystem overview

the previous section. At this point the system has produced the first estimation in joint space. In the next iteration, the system must make sure that all data in joint space are characterized by a polynomial profile. This happens because the motion of all rigid bodies, such as our model, must be expressed with a polynomial function that can produce a second, third or even fourth derivative (jerk)[15].

Once the polynomial fitting process is completed, the system recreates the kinematic model according to the forward kinematic equations and calculates the velocities and accelerations of the human arm joints. Lastly, the system provides an estimation of the torques that affect the subject's arm with the RNE method and presents all of the results in the graphical user interface as denoted in Figure 4.4.



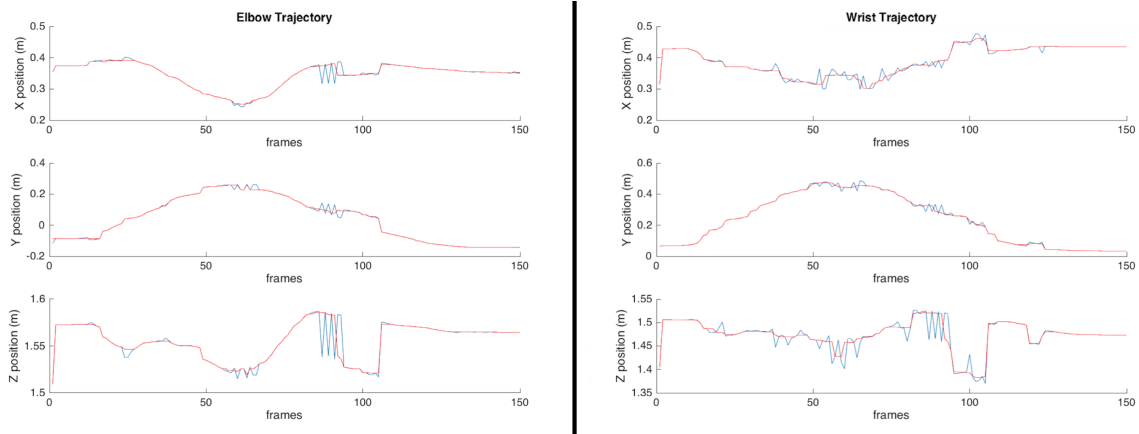


Figure 4.5: Captured Elbow and Wrist positions by the Kinect (blue line). Filtered Elbow data (red line)

#### 4.2.3.1 Assumptions

Before we continue to the experimental results section, we must mention certain assumptions that led to the system’s development. The following hypothesis are centered around the idea that we try to estimate the kinematic and dynamic properties of the human arm with techniques that have been applied to mechanical manipulators. Our assumptions can be summarized as follows:

- First of all, since our study focuses on tracking the human shoulder and elbow, we disregarded the degrees of freedom that the human wrist provides. Traditionally, the human arm is represented with 7 DOFs (3 at the wrist, 1 at the elbow and 3 at the shoulder). However our kinematic model is limited to only 4 DOFs. The inclusion of a 7 DOF model in our system would be extremely challenging, due to the Kinect’s inability to provide an accurate estimation of the wrist’s rotation.
- Second, we did not include in our calculations any external forces that are extruded to the human arm. The reason behind this decision is the absent of a

wrist frame, which would suggest the incorrect propagation of the external force in the kinematic chain by the RNE unit of the system. Additionally, our model doesn't consider any relevant friction between the joint links as it is difficult to simulate the effects of muscle fatigue.

- Last but not least, the assumption that the human arm can be abstracted as a rigid body kinematic chain is incorrect, because one of the key principles of rigid body dynamics is that the modeled body does not succumb to deformation. This is clearly an incorrect statement in our case.

#### 4.2.4 Experimental Motion Results

The team conducted various exercises using the Barret WAM arm robotic manipulator in order to extensively validate the behavior of the system. The robotic arm was used in the experiment as an ad hoc simulator of various rehabilitation exercises. Figure 4.6 shows the physical configuration of the experiment and the captured Kinect skeleton tracker. The results indicate that the system greatly improved the initial estimation of the Kinect. Specifically, in Figure 4.6 the upper four sub figures show the gradual evolution of the captured Kinect trajectories of the human wrist and elbow in blue and red colors. It can be clearly seen from the rapid fluctuations of the trajectories that the Kinect does not regard the physical properties of the human arm as these trajectories doesn't correlate with the motion of the real human arm. However, the four bottom sub pictures of Figure 4.6, that demonstrate the final estimation of the system in equivalent time frames provide a more accurate description of the actual physical trajectory that the human arm followed in the exercise.

It is our omission not to refer to the predominant reason that causes these inaccurate readings of the Kinect v2. Traditionally, if in this particular instance the human arm was performing a free motion without any wearable attachment to

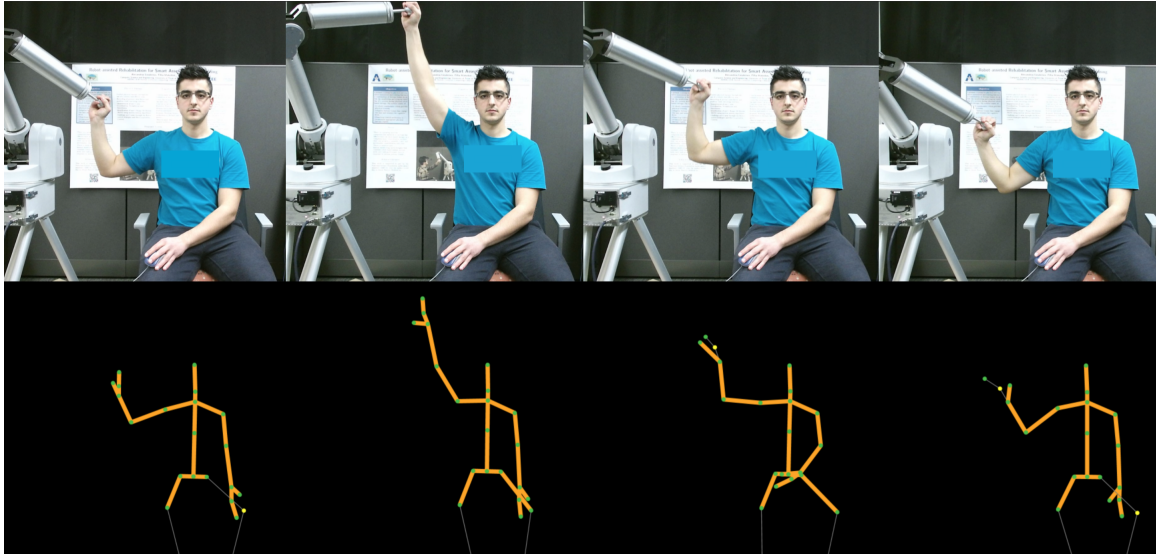


Figure 4.6: The above images present four frames for a particular exercise, where jerky motions from the Kinect is inevitable. Note, that the Kinect doesn't estimate precisely the position of the left arm, because of the same depth information with the rest of the body.

the wrist, the Kinect should be more accurate and without fluctuations. But in this setup, since the participant is grabbing the end-effector of the robotic arm, the Kinect's depth sensor regards the robotic arm as a natural extension of the human's arm and thus provides false readings in certain frames.

Up to this point, the user of the system can perceive how the system obtained the raw data from the Kinect and produced an accurate estimation of the real motion. We will further enrich the experimental results section, by providing additional details for the intermediate steps of the system, which are detrimental for the final derivation of the torque calculations. Starting from the manipulation of the raw Kinect data, the system applies a median filter to eliminate any irregular variation in collected data. Figure 4.5 describe how the median filter is applied in the current experimental set up. Obviously, the median filter removed certain values in the captured elbow and

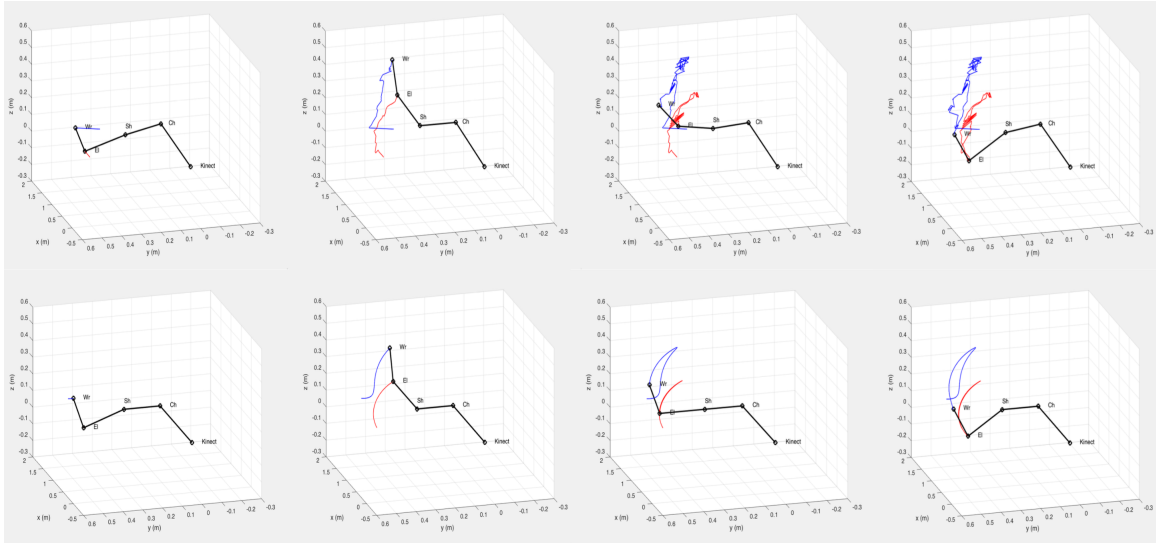


Figure 4.7: The top four diagrams show the evolution of the exercise motion for the raw Kinect model. The bottom four diagrams present the corrected motion based on the polynomial fitting motion estimation.

wrist trajectory that would otherwise make the derivation of the inverse kinematics solution unsolvable. After the trajectory filtering is done, the system performs the polynomial fitting process in the joint space obtained from the solution of the inverse kinematic equations and the filtered trajectories. The left side of Figure 4.8 shows analytically polynomial fitting process of the raw angles obtained from the solution of the inverse kinematics equations and the filtered trajectories. Furthermore, the systems derives the velocities and accelerations through the derivation of the joint space trajectories. The results are shown in the right side Figure 4.8 and 4.9.

To conclude with the experimentation results section, we have to present the torque estimations of the system in this particular experiment. Figure 4.10 provides an illustration of the torque history that was estimated from the RNE module, according to the motion parameters and has already been presented. Note that as we mentioned in section 3, the RNE method also requires the mass of the upper and

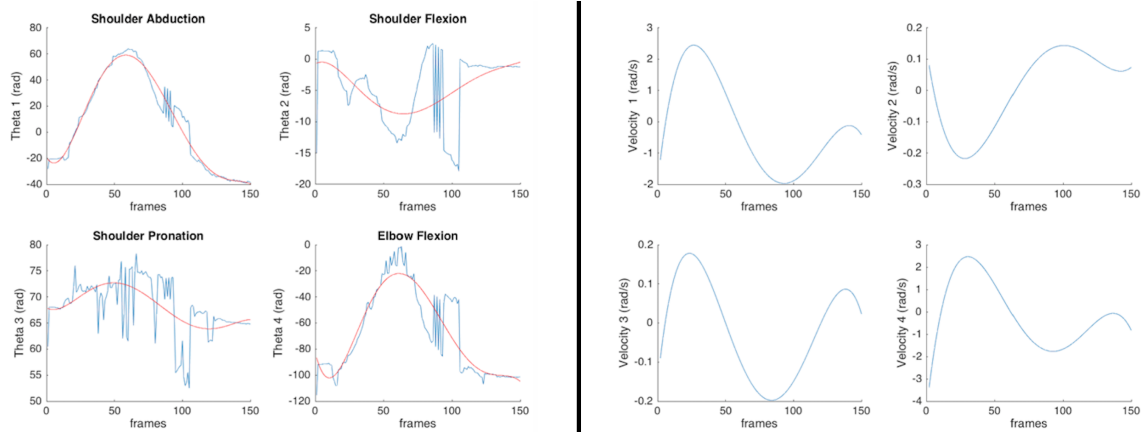


Figure 4.8: The image of the left side depicts the polynomial fitting of the estimated joint angles from the IK solver. The fitted solutions are shown in red color, while the raw angles are in blue. The right image shows the velocities of the fitted joint angles

lower section of the human arm. According to Figure 4.3 we derived the equivalent link's masses ( $m$ ) of the participant. At this point, the user of the system should get an intuitive understanding of the gravitational forces that the arm succumbs to as it performs the trajectory of the exercise. To be more specific, it is clear from the torque history that the torque values in the shoulder are higher than the torque values in the elbow, because the shoulder supports all the weight of the arm. Also, it is clear that when the arm moves from an initial semi flexion position to a full upwards flexion position and the back to original state, a proportion of the gravitational force that is extruded upon the arm shift from joint 1 to joint 2 and then back at joint 1. Lastly, it can be seen than since the arm performs small pronation movements, there is little to no torque applied to arm's third joint.

### 4.3 Quantitative Analysis

We evaluated the computational effectiveness of the motion analysis system through a series of physical experiments that involve the Microsoft Kinect v2 and the

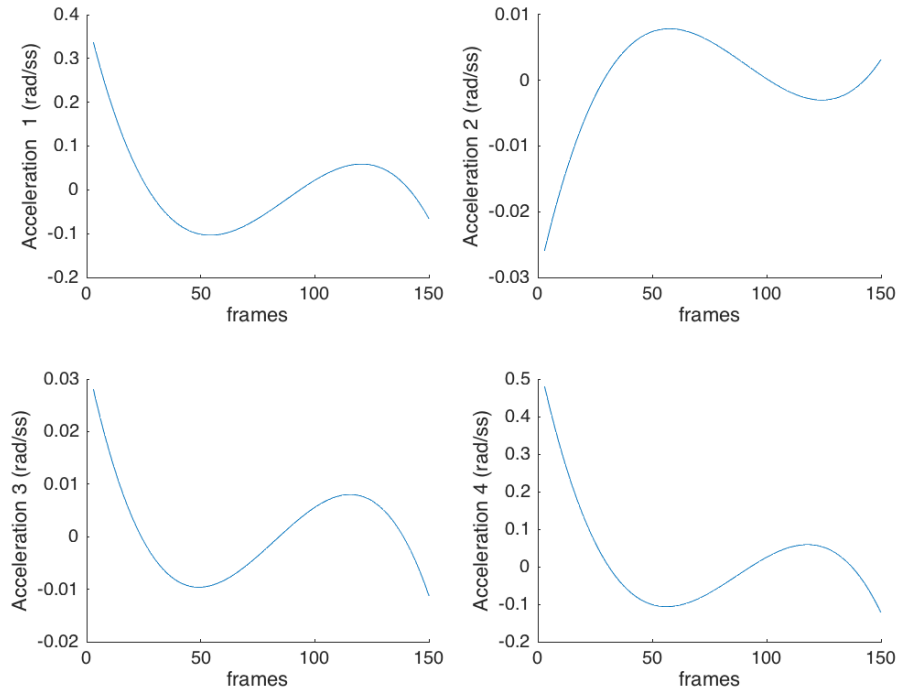


Figure 4.9: Angular accelerations for the four joints

VICON system. As seen from Figure 4.11, the participant was called to perform 10 different rehabilitation exercises with the Barret Arm, while his upper right limb was captured by the Kinect and VICON system simultaneously. The data collection from the three devices was deemed possible via network socket programming. From the Kinect and VICON recorded data, we measured the cartesian position and velocity of the participant’s wrist and elbow and we compared them against the estimations of our motion analysis system.

The VICON is able to track the position and orientation of multiple rigid structures equipped with reflected markers at a rate of 100 Hz with sub-millimeter accuracy [72]. To achieve this precision level, we placed the VICON markers on top of custom made 3D printed mounts, which were then attached on the participant’s right body as seen in the Figure 4.11. The mounts were designed as collars that surrounded

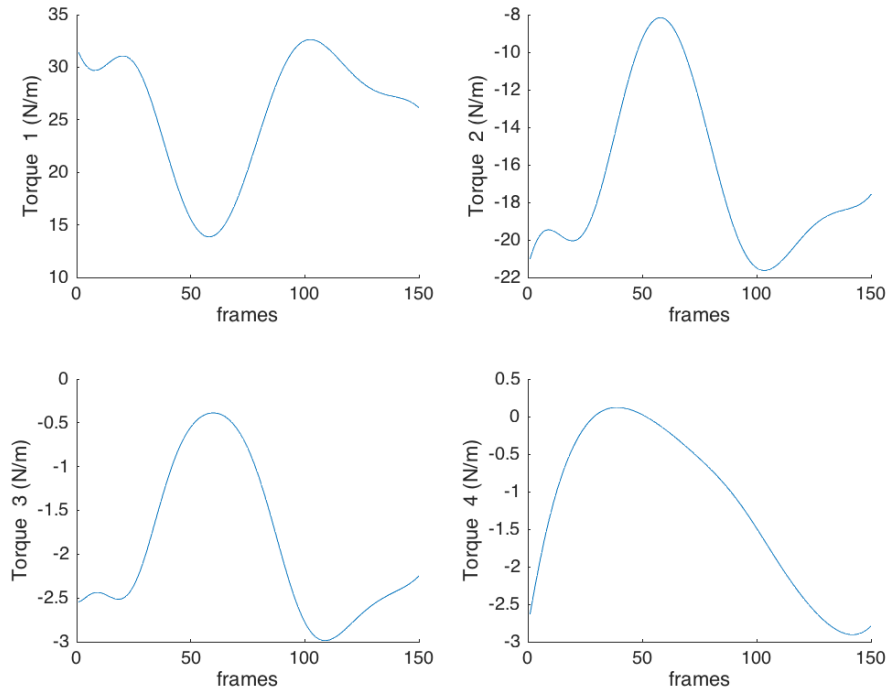


Figure 4.10: Estimated Torques (N/m) for the four joints

the participant’s limbs. Each mount carried three markers placed in a triangular configuration around the mount, so that the triangle centroid points at the joint’s origin (approximately). As a final note, the Kinect and VICON world frames were placed manually as the configuration Figure 4.11 suggests. This means that there is a small offset value (less than half of a cm) in between the Kinect and VICON world frames, since it is practically impossible to place the VICON wand in exactly the same location as the Kinect.

#### 4.3.1 Analysis of Results

According to the above experimental set-up, in this subsection we will present two different exercises that can be used for our quantitative evaluation process with ground truth data. Specifically, Figure 4.12 and Figure 4.14 provide an illustration

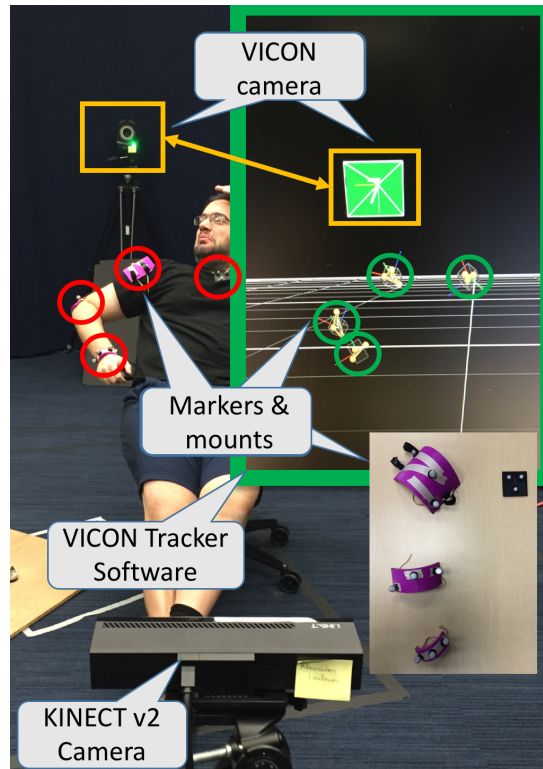


Figure 4.11: Physical Experimental Setup: VICON markers are attached in the 3D printed mounts, which are wrapped around the participant’s joints. The mounts were placed on the wrist, elbow, shoulder and chest.

of these exercises. We selected these two, because in the first one the participant’s arm performs a free occlusion-less motion, while in the second the user’s elbow and shoulder are partially blocked from his wrist and the robotic arm end effector. This is particularly important to note, because traditionally the Kinect predictions are not characterized by sudden fluctuations. However, in the event of an occlusion, the Kinect probabilistic skeleton tracker provides inaccurate predictions.

For the first exercise, the three upper graphs of Figure 4.12 show the evolution of the participant’s motion according to the Kinect data, the estimations of our system’s Kinematic Model and the VICON system. Since the VICON system is our ground



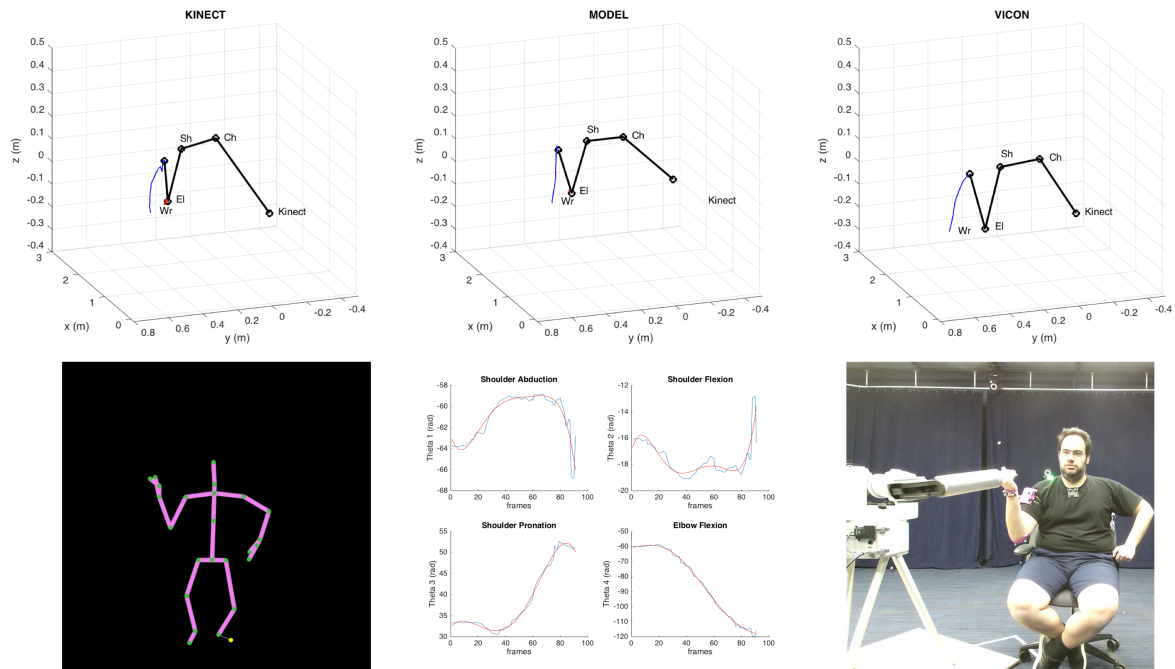


Figure 4.12: Simple exercise: In each subfigure, the above three images show the motion of the human arm from the Kinect, System and VICON scope. The lower subfigures show the skeleton tracker, the polynomial fitting process of the system and the physical set-up of the exercise.

truth, we have to numerically calculate how close to the VICON's trajectory are the estimations of our system and the Kinect.

In Figure 4.13, the blue line stands for the absolute difference between the position of the system's Kinematic Model and the position of the VICON data,  $|P_M(t) - P_V(t)|$ , while the red line represent  $|P_K(t) - P_V(t)|$ , which is the absolute difference between the Kinect and the VICON positions. As expected, for the entire history of these two graphs, the System's position estimation stands below, meaning that our System's predictions are closer to the VICON data. Nevertheless, we notice that during the last section of the exercise after frame 70, the System's and Kinect's predictions are almost aligned.

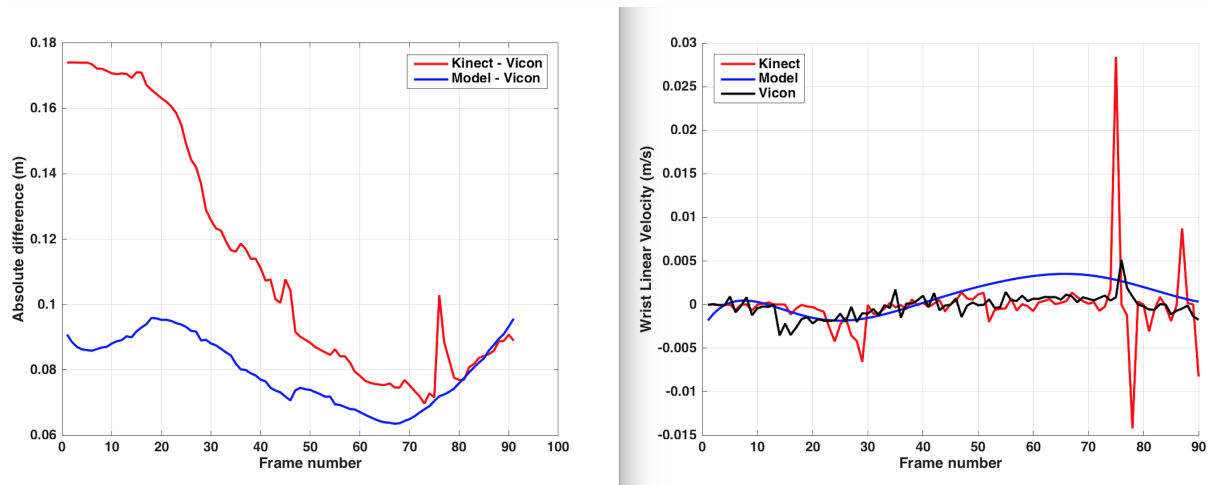


Figure 4.13: Simple Exercise: Distance and velocity comparison

As explained above, the Kinect can predict human motion accurately in occlusion free environments. Moreover, Figure 4.13 shows the velocity profiles of the VICON data, the Kinect data and our system’s estimation. Although it is difficult to precisely determine which velocity profiles are closely related, it is very clear that the Kinect’s velocity profile is characterized by sudden fluctuations that deviate from the VICON. This happens, because our system interpolates a polynomial function in the joint space data that have been captured by the Kinect, as seen from the lower graph of Figure 4.12 .

To continue with the analysis of the second and more complex exercise in Figure 4.14, we notice in Figure 4.15 that the distance between the Kinect and the VICON positions greatly differentiate from that of our Model and the VICON for the entire history of the two trajectories. As expressed earlier, we anticipated this result, because this exercise is not occlusion-free, which has a detrimental effect on the Kinect’s accuracy.

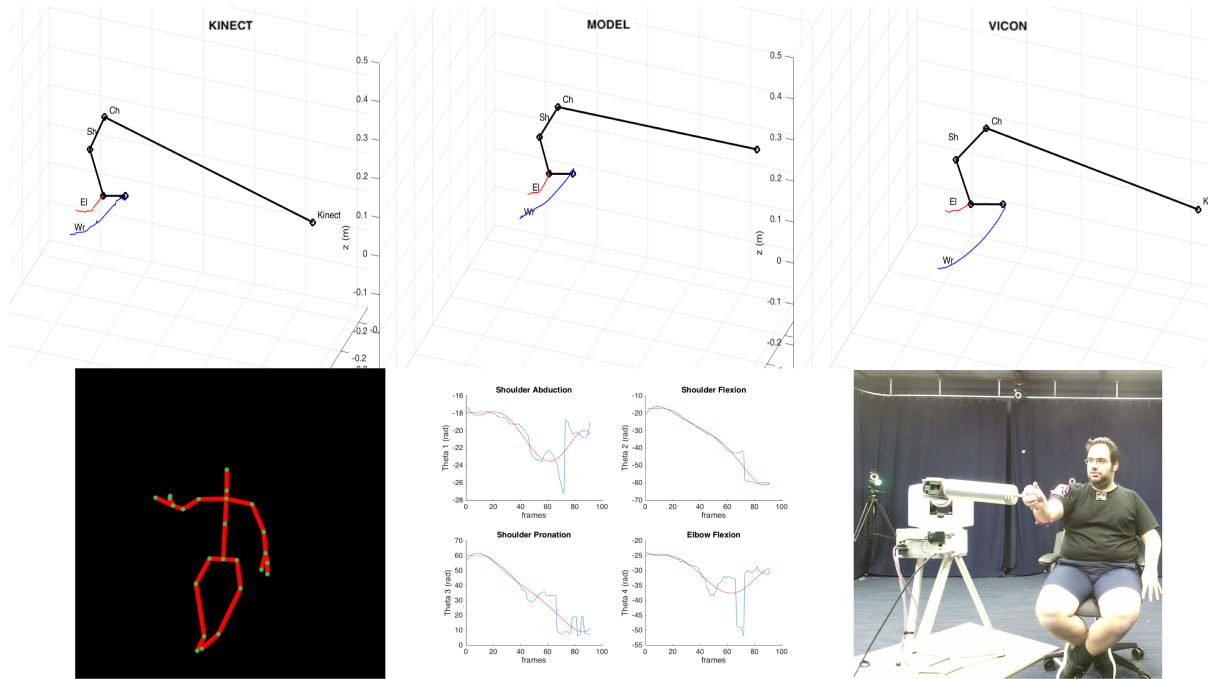


Figure 4.14: Complex exercise: In each subfigure, the above three images show the motion of the human arm from the Kinect, System and VICON scope (Zoom in). The lower subfigures show the skeleton tracker, the polynomial fitting process of the system and the physical set-up of the exercise.

Furthermore, in Figure 4.15 it becomes very clear that the abnormal behaviors of the Kinect around occlusions results in even greater fluctuations in the velocity profile history. Lastly, as part of the evaluation process, it is paramount to mention that certain fluctuations in both VICON velocity profiles in Figures 4.13 and 4.15, are a result of down sampling the VICON data from 100 fps to 30 fps in order to match to the Kinect's frame rate.

### 4.3.2 Algorithm

Algorithm 1 presents the steps for the extraction of the human arm kinematics and dynamics from a Kinect camera. As an input, the system must capture the person who is performing the exercise with the Kinect, according to the configuration that

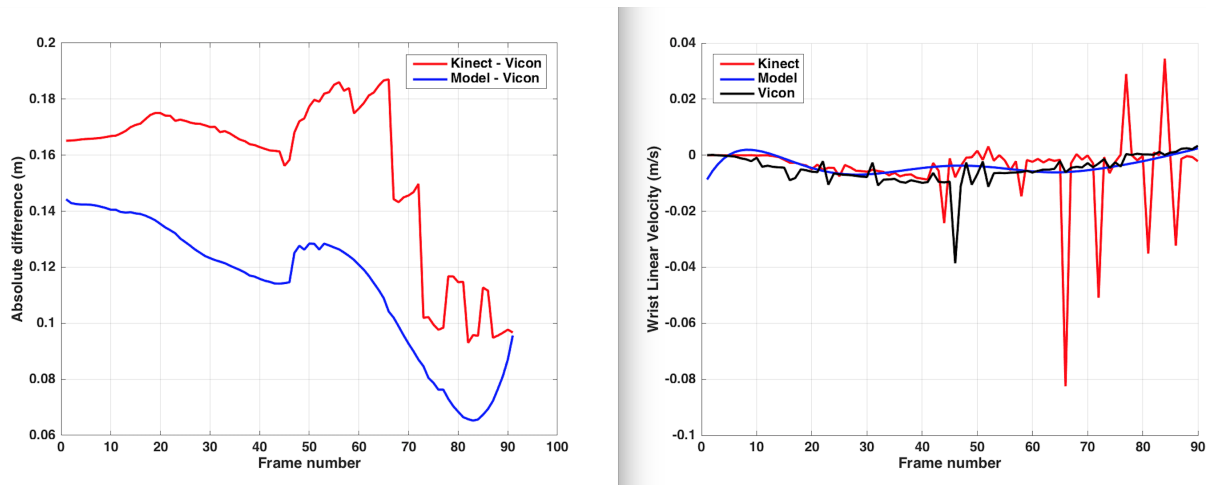


Figure 4.15: Complex Exercise: Distance and velocity comparison

Figure 4.2 suggests. Once the trajectory of the subject’s arm has been captured, the Kinect passes the cartesian positions of the chest, shoulder, elbow and wrist frames to the first unit of the system.

The system then applies a median filter to the Kinect data to eliminate any abnormal behavior from the skeleton tracking algorithm of the Kinect. The result of this module produces a smooth cartesian trajectory that is used by the Inverse Kinematics Solver (IK Solver) to provide an estimation of the angles of the human arm joints. Afterward, the system produces the first estimation in joint space. In the next iteration, the system makes sure that all data in joint space are characterized by a polynomial profile function. This happens because the motion of all rigid bodies, such as our Kinematic Model, must be expressed with a polynomial function that can produce a second, third or even fourth derivative (jerk)[15]. Lastly, as an output, the system provides an estimation of the torques that affect the subject’s arm with the RNE method.

---

**Algorithm 1** Steps to calculate human arm dynamics using Kinect camera system and a robotic arm

---

- 1: **INPUT1:** A sequence  $\{P_t\}_{t=1}^N$  of frames recordings from Kinect, where each  $P_t = (P_{W_t}, P_{E_t}, P_{S_t}, P_{C_t})$  consists of the cartesian position of the wrist, elbow, shoulder and chest.
  - 2: **INPUT2:** Import user height and weight and extract anthropomorphic data for the human body segments for the length and mass of the upper and lower section.
  - 3: **INPUT3:** Add the external forces  ${}^B f_{robot}$  from the robotic arm that are exerted to the user's wrist.
  - 4: Reconstruct a raw model from the captured  $\{P_t\}_{t=1}^N$  frames according to the proposed kinematic model using the Homogeneous transformations in equations 4.3 and 4.7.
  - 5: Apply a moving median filter to the raw position data.
  - 6: Utilize the proposed Inverse Kinematics (IK Solver) to generate an estimation of the joint angles  $\{\theta_r(t)\}_{t=1}^N$  and  $r = [1, \dots, 4]$
  - 7: Apply a higher-order polynomials to the  $\theta_r(t)$  in order to fit the joints estimation  $\theta_e(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5$  to the trajectory sequence (exercise).
  - 8: Generate estimated angles:  $\{\theta_e(t)\}_{t=1}^N$  and  $e = [1, \dots, 4]$
  - 9: Recreate the kinematic model according to the forward kinematic equations 4.3 and 4.7.
  - 10: Apply the Recursive Newton-Euler [20] dynamics algorithm (RNE) and propagate the external force  ${}^B f_{robot}$  from the robotic arm to the user wrist joint.
  - 11: **OUTPUT:** Export the human arm joint velocity, acceleration and torque profiles  $(q, \dot{q}, \ddot{q})$  for the recorded trajectory sequence with the applied forces of the robotic arm.
-

#### 4.4 Joint torque evaluation

Joint torques are of main importance for physicians and occupational therapists to analyze the effects of rehabilitation and to obtain an indicator of patient's functional capacity to perform a motion [5]. A joint's strength is assessed through the measurement of the maximal joint torque, which represents the resultant action of all muscles crossing the joint. Manual muscle testing (MMT) is a measure of upper and lower body strength that occupational and physical therapists often complete as part of a clinical evaluation and to measure progress in therapy. MMT is a graded scale (typically on a scale from zero to five) that is used to assess patients with neurological or orthopedic impairments [16]. A score of zero indicates that there is not any muscle contraction to five indicates that strong pressure can be applied. Many issues arise because MMT can be subjective based on many factors. The validity and reliability of MMT are dependent upon a variety of factors including training of the therapist; the patient's diagnosis, pain level, and other physiologic issues; which muscle is tested; the position of the patient; hand placement of therapist during testing; and variability between therapists [35].

The rehabilitation therapists may change the parameters of the exercise or activities (commonly referred to as grading) between or during treatment sessions, based on confounding patient factors such as pain or fatigue [17]. For example, the therapist may change the number of repetitions, the number of sets, and/or the amount of resistance given to the patient. These parameters may remain consistent over time or need to be changed during each session based on the patient's performance and muscle fatigue. Multiple researchers have attempted to generate models for muscle fatigue based on joint torques and muscle contraction levels. For example, the authors in [58] utilized electromyographic data and derived an analytical muscle model, taking into account physiological and anatomical data, to estimate the joints' torque.

This model helps them to generate joint torques and stiffness values while the user is interacting with a rehabilitation instrument.

#### 4.4.1 Experimental setup at the shoulder's joint

In order to fully validate that the torque estimation derived by our biomechanical model is correct, we conducted a series of experiments that involves primitive arm movements that isolate the shoulder axis and muscle activations. Figure 4.16 shows the Delsys sensors placement in the user's arm and the muscles area that are associated.

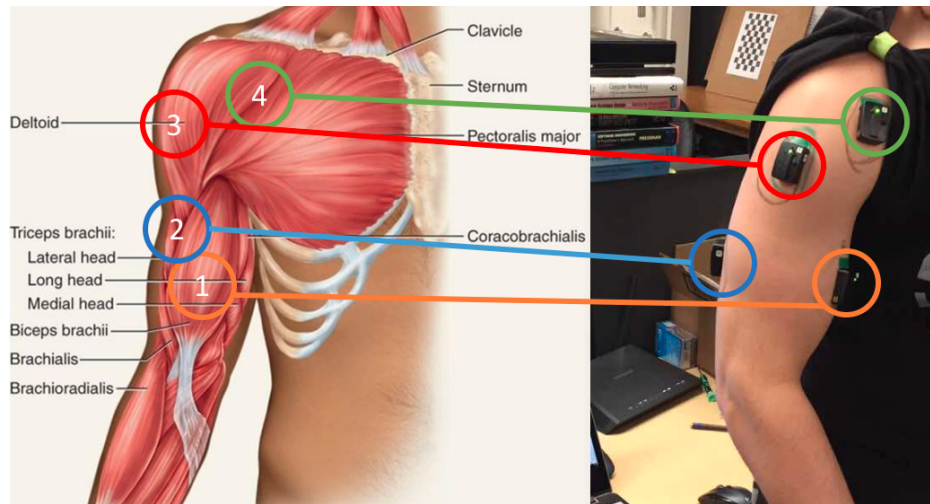


Figure 4.16: Muscle to sensor placement

Our initial goal is to correlate the joints' frame placement, according to the Figure 4.2, with the muscles that are triggered and move the shoulder at each axis. For this reason, sensor 3 has been placed on the Lateral Deltoid muscle, sensor 4 has been placed to the Anterior Deltoid muscle area, connecting to the clavicle, and sensor 1 and 2 were placed to the biceps and triceps respectively. The exercise that is first chosen is the shoulder abduction (Figure 4.17). This allows the first frame of

the shoulder to rotate along axis  $z_1$  in the positive direction. The second exercise is the shoulder forward flexion (Figure 4.18) that allows the second shoulder frame to rotate along axis  $z_2$ .

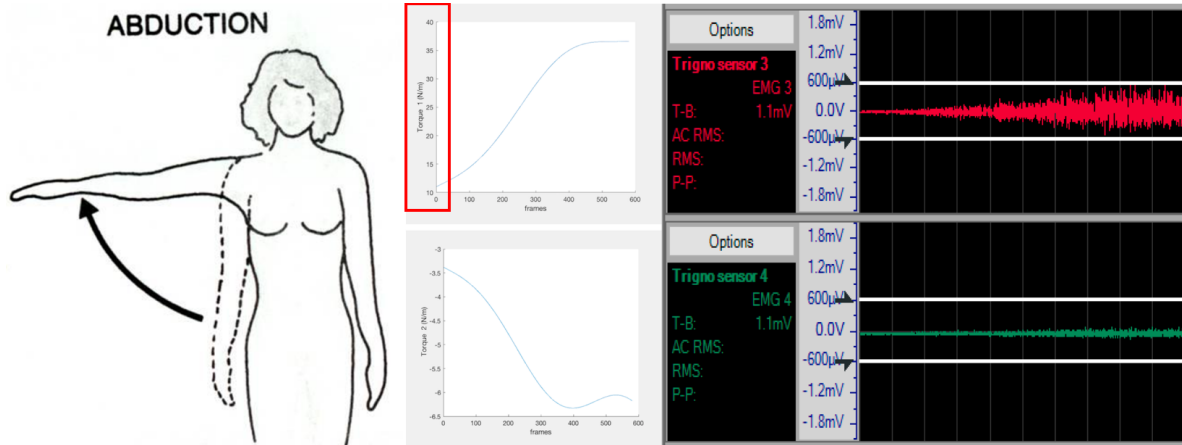


Figure 4.17: Shoulder Abduction

From the experimental results, in Figure 4.17, it is obvious that the first exercise triggers the third sensor more which correlates the deltoid's muscle movement. The torque values of the frame 1 at the beginning are close to 11 N/m and when the shoulder is fully abducted they reach 36 N/m. For the frame 2, the absolute torque values increased slightly exactly like the correspondence muscle contraction (sensor 4).

The second experimental results (Figure 4.18) show the opposite torque value estimation which corresponds with the muscles' activation. The torque values of the second frame are increased from 9 N/m to 34 N/m relatively as sensor 4 jumps. The frame 1 torque values show some discrepancy but this is caused because of the axis  $z$  is crossed while the user is flexing forward his arm. Also, sensor 3 is reacting to this motion as the deltoid muscle is triggered slightly. It should be mentioned that



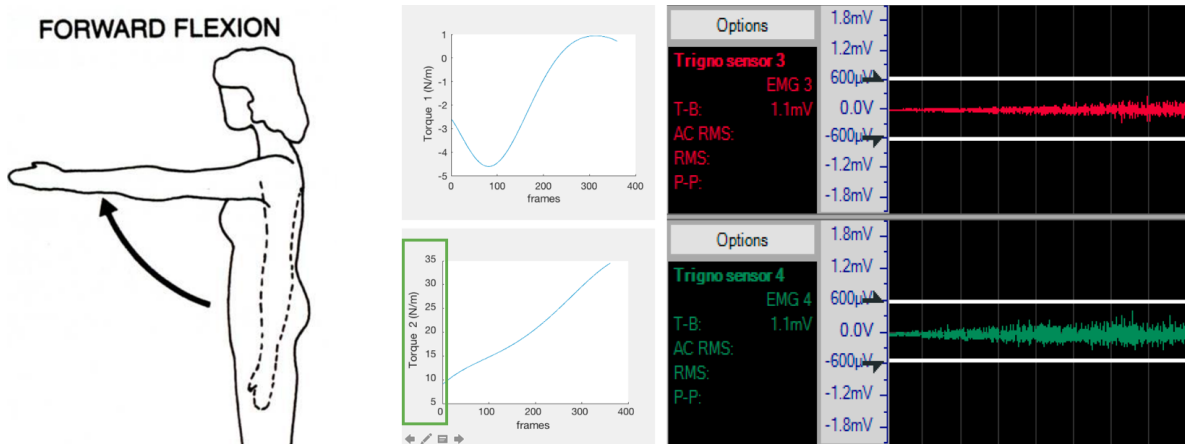


Figure 4.18: Shoulder Forward Flexion

is difficult to isolate the muscle's activation at the shoulder as they are wrapped together to help the shoulder's rotation to the three axes.

#### 4.4.2 Experimental Analysis of the electromyographic data

To analyze the electromyographic (EMG) data, the collected signal was first filtered. Filtering was done in 3 stages: High Pass filter, Low Pass filter and Notch Filter. A butterworth filter was used to design these filters. The corner frequency of the high-pass filter was 10 Hz while the corner frequency of the low-pass filter was 500 Hz and the frequency of the notch filter was 50 Hz. This process removed any noise below 10 Hz, above 500 Hz and at 50 Hz.

After the filtering process, the peaks of the EMG were found. These peaks were used to find a relationship between the torque extracted from the Kinect data and the EMG extracted from the Delsys. We used the inbuilt peak detection function in MATLAB to detect the peaks. Furthermore, both the EMG and the torque data were downsampled to 1 HZ, resulting in one data point per second. This was done

for EMG data too. Peak data at each second was calculated as the mean of the EMG peaks for 500 ms on either side of the second mark.

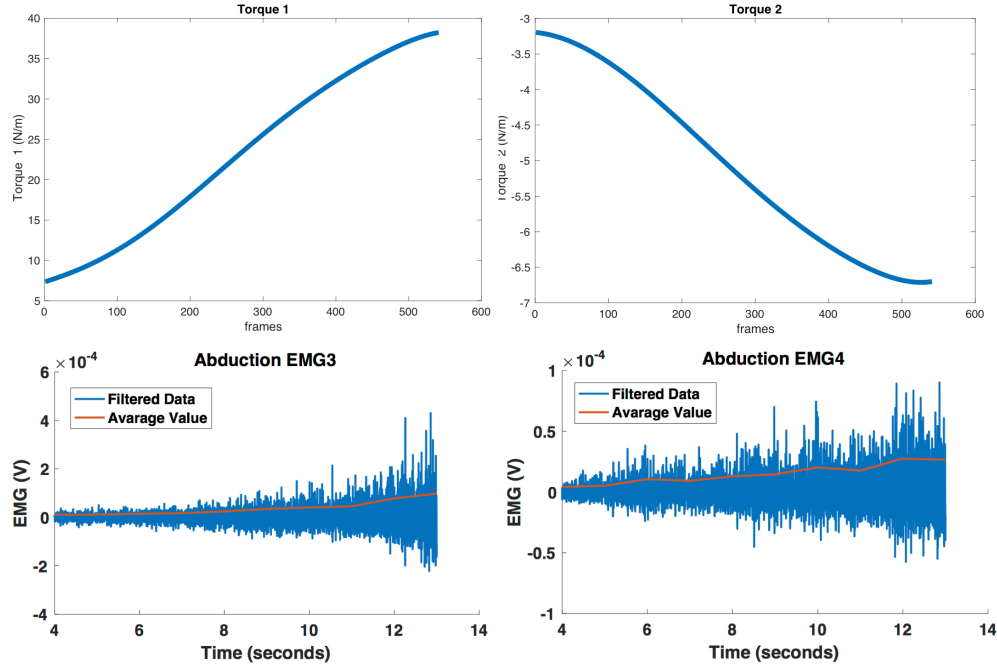


Figure 4.19: Shoulder Abduction

Lastly, the relationship was found by using Kendall's Rank Correlation method. This is a nonparametric correlation method. It operates by assigning ranks to each datapoint and calculating the concordant and the discordant pairs. Consider a data point in a set, any data point below the considered one is assumed to be a concordant pair if the rank for the new data point is smaller than the rank for the considered data point. It is a discordant pair if the rank for the new data point is greater than the initial data point. Kendall's correlation calculates  $\tau$  by using the following formulae:

$$\tau = \frac{\sum D - \sum D}{\sum D + \sum D} \quad (4.15)$$

Table 4.2: Correlation values between torque and electromyographic signal.

Exercise	Sensor	Tau	P
Abduction	<i>EMG3</i>	1	5.51E-07
	<i>EMG4</i>	-0.82222	3.58E-04
Forward Flexion	<i>EMG3</i>	-1	4.96E-05
	<i>EMG4</i>	1	4.96E-05

where  $C$  are the Concordant Pairs and  $D$  are the Discordant Pairs. This yields a value between -1 and 1 where -1 indicates a strong negative correlation and '+1' indicates a strong positive correlation. 0 indicates no correlation.

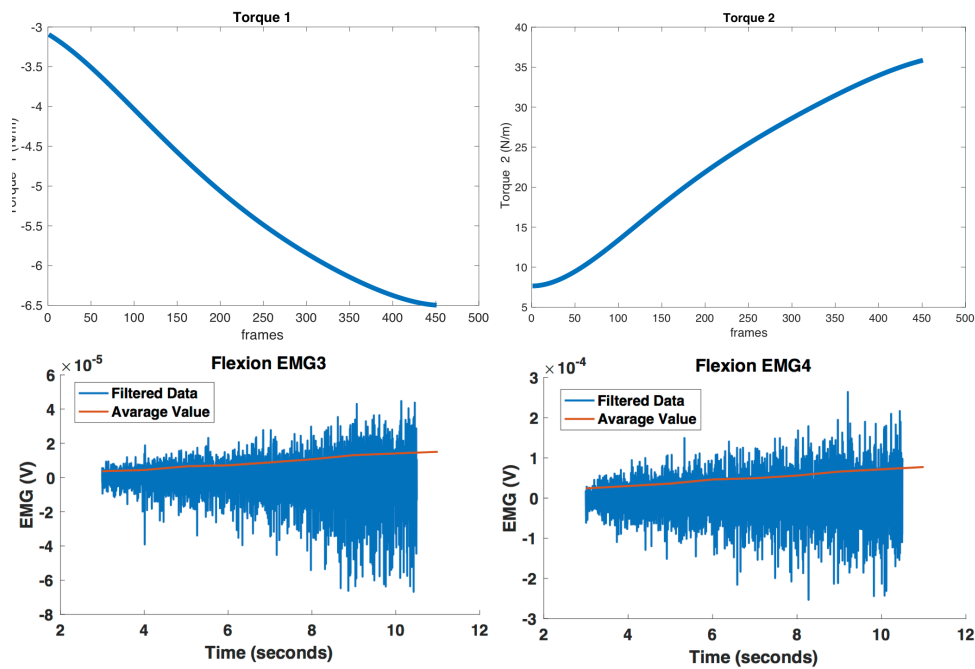


Figure 4.20: Shoulder Forward Flexion

Figure 4.19 and 4.20 show the opposite correlation of the torque values estimated by our biomechanical model and the electromyographic filtered signal after the analysis. These results confirm our hypotheses for torque estimation per axes with the isolated muscle to electromyographic data analysis. Specifically, in Table

4.2 we can see that the correlation values for the shoulder abduction motion give  $\tau = 1$  and  $\tau = -0.82222$  for the EMG3 and EMG4 respectively. This means that the *torque1* value has linear increasing rate such as the EMG3 signal. On the other hand, *torque2* value follows closely the linear decreasing of the signal. An analogous trend is observed in the shoulder forward flexion EMG and torque correlation graph (Figure 4.20), because the increments rates are opposite. Thus, we can justify the torque values and claim that our biomechanical model can be used for shoulder torque estimation in rehabilitation exercises.

The vision module uses the Kinect's skeletal tracking to monitor the user's effort in an unobtrusive and safe way, by estimating the torque that affects the user's arm. The system's torque estimations are justified by capturing electromyographic data from primitive hand motions (Shoulder Abduction and Shoulder Forward Flexion).

## CHAPTER 5

### Haptic Force-fields and control strategies

In this chapter, the application of the Haptic robot controller for the user's hand coordination is demonstrated. Also, different control strategies and protocols for intelligent adaptive robot-based exercises are presented.

#### 5.1 Related Work

One simple exercise in rehabilitation is to repetitively follow pre-described trajectories to help users strengthen their weakened muscles or regain motor control. A haptic path can be defined as a virtual tunnel that uses force feedback to help users move through that path or constrain them from deviating in other directions. The authors in [6] use gait trajectories to help users while doing exoskeleton gait training on the treadmill. They proposed a haptic controller designed to be 'assist-as-needed' system, which can apply suitable forces on the patient's leg to help him move on the desired trajectory. Similarly, in upper limb rehabilitation, [61, 66] tracking the performance and progress of the users, can be achieved by comparing their measured trajectories with the Dynamic Time Warping (DTW) algorithm [29]. The literature has shown that haptic feedback/guidance can help the users improve their tracing abilities by following a prescribed trajectories [44, 27]. This haptic feedback can be by probing the user's hand through the path or by providing perpendicular forces that prevent the user's hand to deviate from the desired path.

## 5.2 Haptic Device Dynamics

The dynamics of a robotic manipulator that is never in contact with another external object are as follows:

$$D(q)\ddot{q} + (C(q, \dot{q}) + B)\dot{q} + G(q) = \tau \quad (5.1)$$

where  $D(q) \in R^{3 \times 3}$  defines the inertial matrix,  $C(q, \dot{q}) \in R^{3 \times 3}$  is the Coriolis and the centripetal forces matrix,  $G(q) \in R^{3 \times 1}$  defines the gravity forces vector,  $B \in R^{3 \times 3}$  indicates the viscous coefficient matrix, and  $\tau \in R^{3 \times 1}$  is the torque input. In our case the patient is driving the system, so  $\tau = \tau_c + \tau_h$ , and  $\tau_c \in R^{3 \times 1}$  is the guidance control and  $\tau_h = J^T f_h$  is a sign of the human-haptic interaction, where  $J^T$  stands for the transpose Jacobian of the haptic device, and  $f_h$  shows the patient's performance force vector.

## 5.3 Haptic Path

The robotic arm can help guide the user to follow a precise trajectory as dictated by previously recorded exercises done by a physical/occupational therapist. The user can attempt to perform the prescribed exercise and if he/she deviates from the prescribed trajectory, an appropriate correctional force is applied by the robotic arm to guide him/her back to the correct trajectory. Besides spatial, the deviation can also be temporal, i.e. the user performs the exercise much slower or much faster than the therapist. When either of the two deviation types occurs, an error-correction force is applied to bring the patient's hand position closer to the prescribed trajectory.

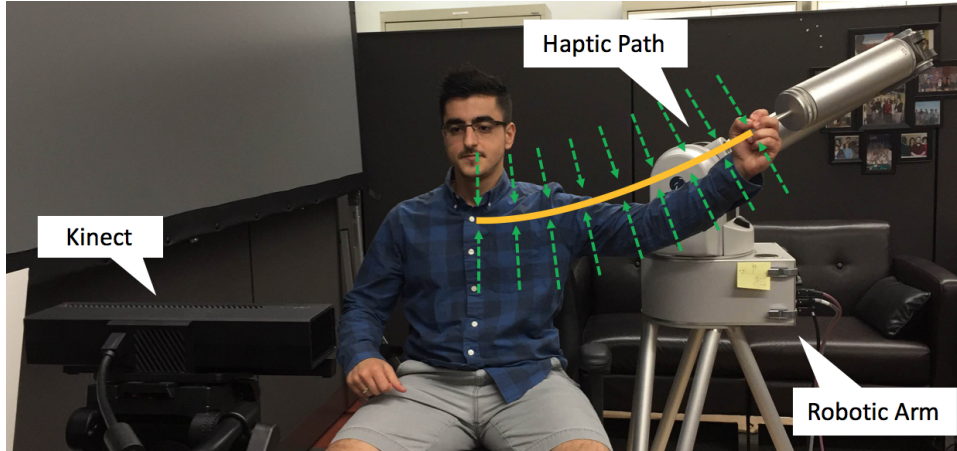


Figure 5.1: Prescribed exercise represented by a haptic path.

### 5.3.1 Haptic forces

One of the tasks that the patient should perform in our experiments, is to stay close to a prescribed trajectory in the 3D space. For that particular reason, a force-field assists the patient to stay close when he is deviating. A force-field (Figure 5.1) is provided from the given start position  ${}^B p_{start}$  to the end target position  ${}^B p_{end}$  of the path, as Figure 5.2 depicts. If the patient deviates from the given path, a perpendicular force will be activated in order to push the patient's arm to stay close to the path. The robot's end-effector position  ${}^B p_t$  searches for the closest point at the haptic path at each moment. The direction and magnitude of the force in the end-effector position  ${}^B p_t$ , is calculated by the  ${}^B p_N$  point and the absolute distance  $d_t$  respectively.

The haptic path has been reconstructed with the use of an impedance control mechanism that controls the position of the robot's end-effector ( ${}^B p_t$ ) at the corresponding trajectory point ( ${}^B p_{NN}$ ). The impedance control aims to increase or decrease the compliance (stiffness) of the robot in order to allow the user to deviate more or less from the predefined trajectory. This stiffness values ( $K$ ) constrains the

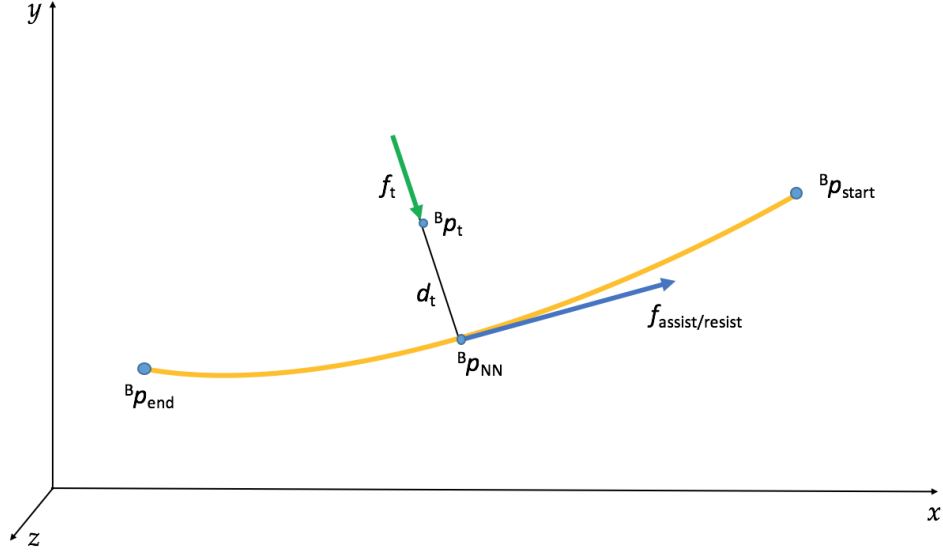


Figure 5.2: Prescribed exercise represented by a haptic path.

user to the trajectory and acts as the spring constant. The force generated ( $f_t$ ) is equivalent to  $f_t = K \times d_t$ . The proportional gain (P) that represents the stiffness of the force-field of the impedance controller, behaved similarly to the K spring constant value. By changing the P value we are able to bring the patient's hand closer to the therapist's prerecorded trajectory.

The haptic control law for that particular haptic path can be described properly by integrating the equation 5.1 to :

$$D(q)\ddot{q} + (C(q, \dot{q}) + B)\dot{q} + G(q) = \tau + J^T(\vec{q})[K_p(\vec{p}_{NN}) - \vec{p}_t] + f_{assist/resist} \quad (5.2)$$

where the factor  $K_p(\vec{p}_{NN}) - \vec{p}_t$  represents the  $f_t$  force generated perpendicular to the  $Bp_{NN}$  point, while the factor  $f_{assist/resist}$  is defined as the constant force that push or repel the user's hand towards a tangential direction with the prerecorded trajectory.



### 5.3.2 Haptic Control

The control chart of the proposed control system can be found in Figure 5.3. The Barrett WAM robot is directly interacting with patient's arm  $\tau_p$ . All motion parameters that associate the kinematics of the robot are measured with internal sensors. In our case, the measurements are provided through the Barrett WAM's Puck sensor that operates in 500 Hz.

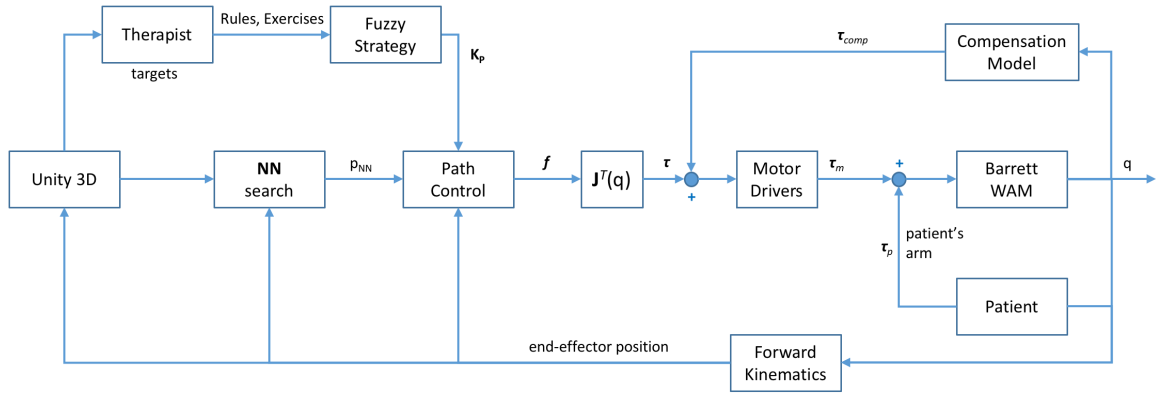


Figure 5.3: The impedance haptic path controller implemented in the Barrett WAM robot. A control chart overview.

At the above figure, the forward kinematics of the robot are utilized to calculate the end-effector position, which is supplied to the visual interface implemented in the Unity 3D game engine. This depicts the end-effector's trajectory as well as the start position  $p_{start}$  and target position  $p_{target}$  that defines the haptic path. Such information is used to calculate the nearest neighbor point  $p_{NN}$  on the track, and the tangential vector  $f_{assist/resist}$  by using the end-effector position. The transposed Jacobian  $J^T(q)$  is used to calculate the corresponding joint torques  $\tau$  that accelerates the robot. Additionally, the compensation model  $\tau_{comp}$  which is consisting of the

friction, gravity and spring compensation module, provides the necessary torque to keep the arm stationary.

### 5.3.3 Haptic Experiments

In order to test the compliance of the impedance controller, we recruited one chronic stroke patient (Fig. 5.4) and we conducted three experiments with different proportional values. Then, we analyzed the effects of the haptic controller by using the Dynamic Time Warping method to derive spatial or temporal error deviations in the user's cartesian trajectory. Figures 5.5 and 5.6 illustrates the Cartesian position in the plane during the haptic path exercise. In particular, the desired trajectory is shown with red targets to the stroke patient (Virtual Exercise) (Figure ??) and is represented by the red line in Figure 5.5. The stroke patient was instructed to perform each exercise (Haptic path) with the best of his abilities and try to reach all the red virtual targets.

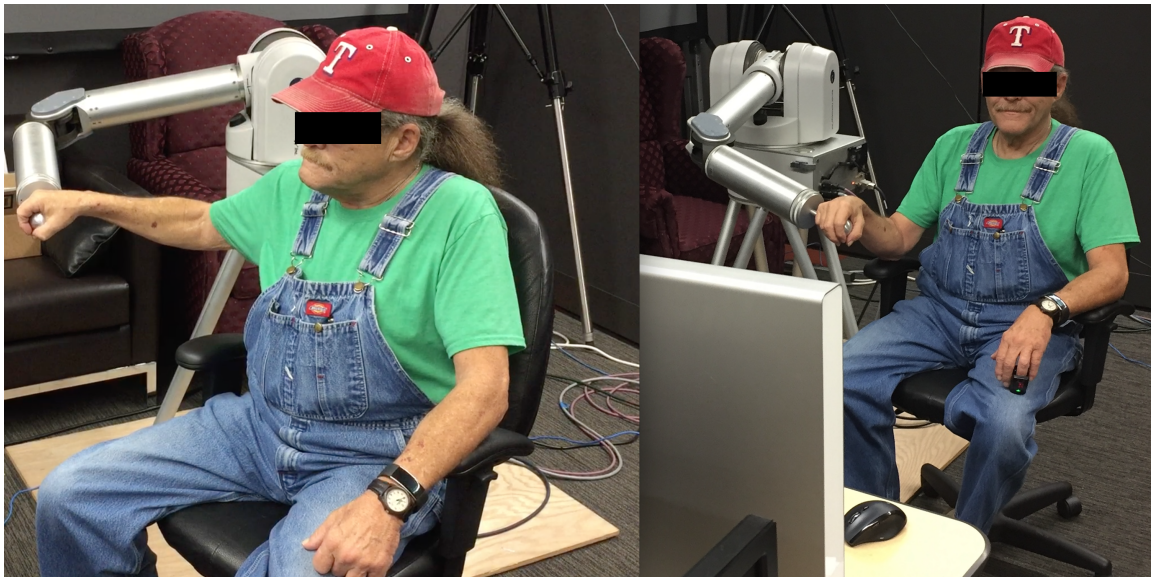


Figure 5.4: Volunteer chronic stroke patient.

### 5.3.4 Haptic Response

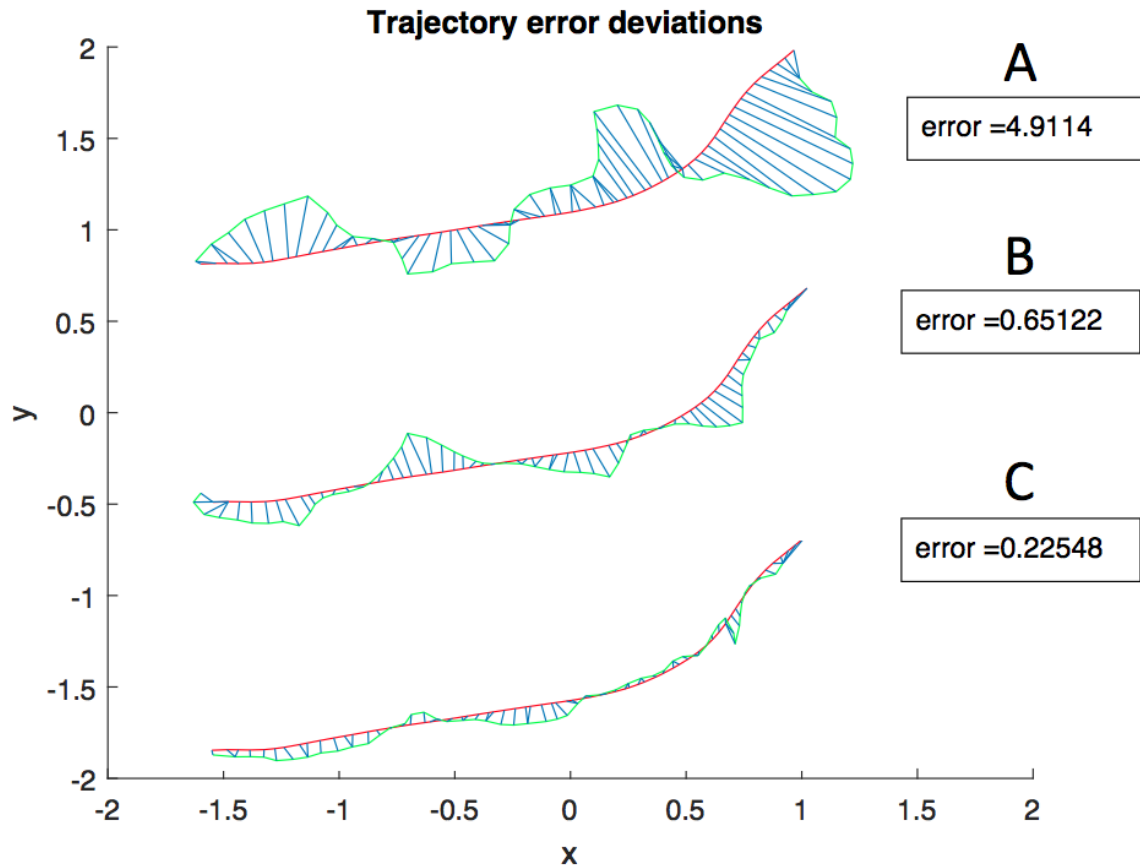


Figure 5.5: Stroke patient's error deviation for three different proportional values: A)  $P = 50$ ; B)  $P = 100$ ; C)  $P = 800$ .

Three exercises were performed with the stroke patient with small breaks of 5 minutes (Figure 5.5). In the first, exercise (A) the stroke patient was unassisted ( $P = 50$ ) and his error deviation was  $error = 4.9114$ . At the second execution (B), the stiffness value of the impedance controller was ( $P = 100$ ) and the stroke patient's error trajectory deviation from the prescribed path was  $error = 0.65122$ . Finally, we

increased the robot's assistance (C) with ( $P = 800$ ) and he managed to execute the exercise correctly ( $error = 0.22548$ ).

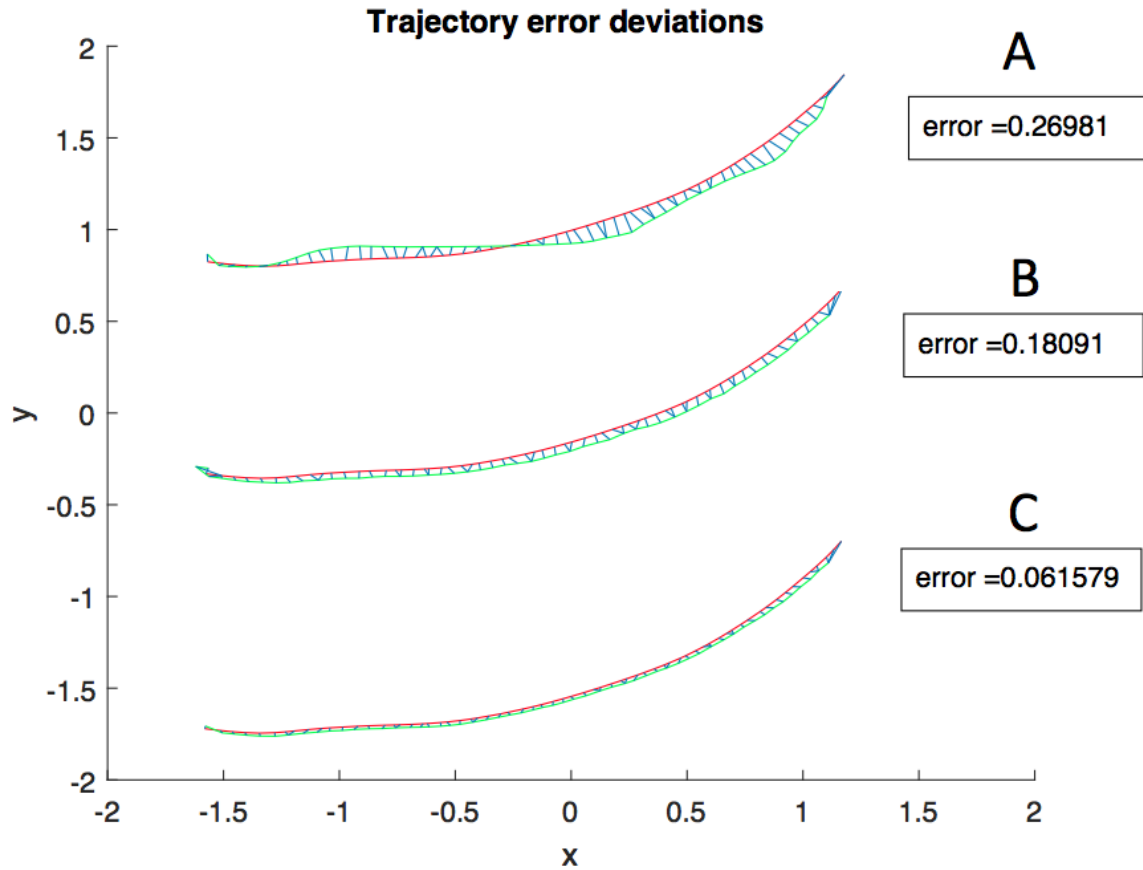


Figure 5.6: Unimpaired user's error deviation for three difference proportional values: A)  $P = 50$ ; B)  $P = 100$ ; C)  $P = 800$ .

Similar experiments were conducted with an unimpaired user. In Figure 5.6 the user's performance did not change drastically as he manages to control the motion of his hand successfully and his error deviation is getting better as long the robotic arm constrains him to the prescribed trajectory. It is clear that when the applied rendered forces constrain the users to the prerecorded exercises, the error deviation is getting

smaller. This phenomenon implies that patient/user will be able to increase hand coordination and improve motor skills with the passage of time. In the next chapter, we will introduce how we are able to evaluate the user's hand coordination using the DTW algorithm and how we can adapt the robot's haptic controller in order to guide and support the user's hand through the entire exercise.

## CHAPTER 6

### Artificial Intelligence for adaptive human-robot interaction

In this Chapter, we show how Artificial Intelligence methods such as Machine Learning, Pattern Recognition, and Fuzzy Inference System can be utilized, in order assess the user's performances, identify the user's errors and adjust the rehabilitation treatment. Three proposed interfaces have been developed (MAGNI, MAGNI 3D, and MAGNI Dynamics) to acquire user motion data and to adapt the exercise difficulty level dynamically.

#### 6.1 Self-managed physical therapy

The subsystem, "called MAGNI <sup>1</sup>", records the position of the subject's hand during game interaction with the robotic arm and analyzes this data using pattern matching and machine learning algorithms, in order to guide self-managed physical therapy. The purpose of this assessment tool is to encourage and engage the user in performing the exercises using a 3D balloon popping game. The general idea of our system is to learn the patient's weaknesses and his arm limitations when he interacts with the game and to adapt the game level of difficulty in the next sessions. The balloons have random positions at the beginning of the game, but as the patients interact with the game the balloons positions are targeted to force the user to reach out farther for some of them and to help improve their capabilities. The interaction of the user with the game provides us with motion trajectories which can be analyzed and interpreted to assess the patient's arm limitations and weaknesses. Since the

---

<sup>1</sup>MAGNI is the God of strength in Norse mythology

game requires the user to pop the balloons that appear in the game, the range of motion and forces that the user places upon the Barrett WAM Arm's end-effector can be decomposed to primary and basic exercise movements.

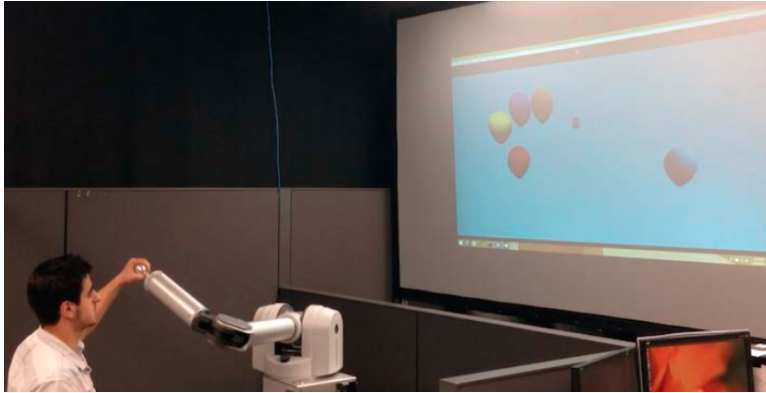


Figure 6.1: Human Robot and Game Interaction

### 6.1.1 Theoretical Background

Support Vector Machines (SVM) have demonstrated good classification performance and have widespread successful use in many pattern recognition problems. These classifiers rely mainly on the hyper plane optimization that maximizes the margin, or the distance between the separating hyper plane and the training examples nearest to the hyper plane [8]. We rely on Multi-Class Support Vector Machines to classify the patient exercises and movements when they interact with the 3D Balloon game. The training feature-set used by multi-SVM is the direction and the curvature obtained from the hand motion trajectories during exercise. In our system each trajectory passes a preprocessing step that divides it in equal-distance sub-trajectories before it is incorporated into the classification model, with high importance for the shape modeling. Figure 6.2 shows the training and the classification phases of our system.

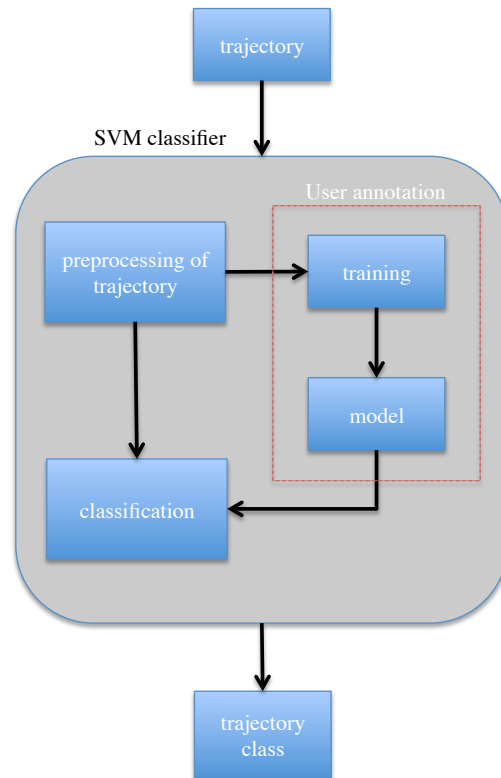


Figure 6.2: The procedure of the trajectory classification

The Hidden Markov Models (HMM) approach belongs to supervised learning and statistical modelling methods for sequential data. It has been used prominently and successfully in speech recognition and, more recently, in handwriting recognition and visual recognition of sign language. The sample model is described as a graph with four internal and two marginal states connected by oriented transitions. Moreover, there are six associated output vectors, as seen in figure 6.3.

The trajectory classification is similar to the speech recognition tasks [8]. A trajectory is a continuous quantity that can be described analytically as the position of the object in time. An object trajectory  $O$  is a potentially infinite sequence of state



vectors  $o(t) = [x, y, z, dx, dy, dz]$ , where the first three denotes the Cartesian position and the three last the direction.

The trajectory classification problem can be formulated as to identify the class  $c_i (i = 1..N)$  to which belongs the trajectory state sequence. The basic formulation of the problem is given by maximization of a conditional probability:

$$i^* = \arg \max_i P(c_i|O) = \arg \max_i \frac{P(O|c_i)P(c_i)}{P(O)} \quad (6.1)$$

We use Bayes theorem in (6.1), because we cannot evaluate  $P(c_i|O)$  directly. Assuming we know prior probabilities  $P(c_i)$  and  $P(O)$ , we are about to compute the likelihood  $P(O|c_i)$ ; the probability of the sequence  $O$  knowing the class  $c_i$ . To compute this, we should have a model  $M$  for class  $c_i$ . The model is a finite state automaton with  $K$  states generating sequence  $O$ . There are transition probabilities  $a_{k,j}$  between the states. Except first and the last state, states are emitting or generating output probability density function  $b_j(o(t))$ . In the figure 6.3, there is a sample configuration of  $A = [a_{k,j}] (k, j = 1..K)$ , the transition matrix, which defines the probability of transition to the next state for each combination of HMM states. The probability of passing an object  $O$  through a model  $M$  by a way  $X$ . is defined by equation 6.2.

$$P(O, X|M) = a_{x(o)x(1)} \prod_{t=1}^T b_{x(t)}(o_t) a_{x(t)x(t+1)}. \quad (6.2)$$

For the training and classifying procedure we have used the Hidden Markov Model (HMM) Toolbox for Matlab [56] with four mixtures of diagonal Gaussians. To classify a sequence into one of 6 classes (exercises), we trained up 6 HMMs, one per class, and then we computed the log-likelihood that each model gives to the test sequence; if the  $i_{th}$  model is the most likely, then declare the class of the sequence to be class  $i$ .

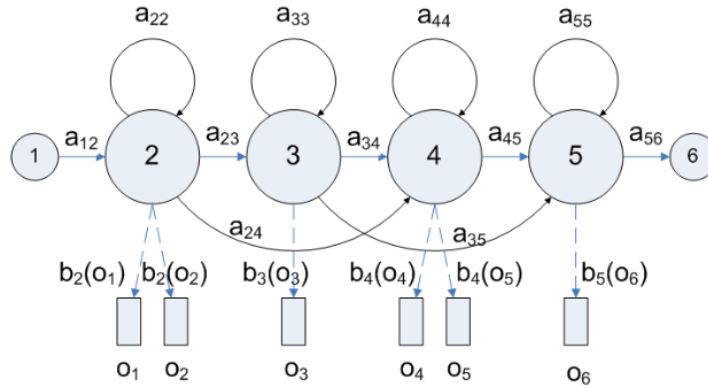


Figure 6.3: Hidden Markov Model configuration

A two-level exercise classification system has been developed. The first level uses the SVM classifier which can define the shape of each exercise (3 or 6 base shapes of exercises) according to the curvature and the internal direction of each trajectory. In the second level we use the Hidden Markov Model to identify the direction of each exercise. For example if the multi-SVM classifier provides us with the information that an exercise belongs to a line-shape then the HMM can identify the direction of this line according to the training direction. The system incorporates the HMM inside the Motion Analysis of the user with the 3D video game so as to identify the position and the orientation of his movement. In other words, for each exercise that the system asks the user to perform, we have trained 6 different orientations for each of them in order to identify the portions that the user can accomplish successfully.

All experiments were conducted on an Intel i7 machine with 8 Gigabytes of main memory, running MacOSX. Everything is implemented in Matlab. Additionally, LIBSVM [12] with a linear kernel is used to build a classifier, and the parameters of LIBSVM are set to the default values.

### 6.1.2 Motion Analysis

Each balloon, or each sequence of balloons, that the user tries to pop in the game, generates a 3D trajectory of the motion of the robotic arm's end-effector. This trajectory can be compared and tested with therapist's exercises' trajectories, used as gold-standards. In this work, we have developed a novel, two-level classification scheme to classify motion trajectories. In the first level we use the SVM classifier, which can identify the shape of each exercise according to the curvature and the internal direction of each trajectory, and in the next level we use the Hidden Markov Model to identify the direction of each exercise according to the centralized sub-trajectories points. The accuracy of the first level SVM classifier reaches 92% when the basic exercises are 6 (circles, line, u-shape, square, gamma and figure eight) and the second level Hidden Markov Model classifier provides us with 95-100% accuracy.

In this work, we have recorded 30 trajectories for each class. From this data, we used 20 trajectories of each class for training and 10 trajectories for testing. This means we have a total of 120 trajectories for training and 60 trajectories for testing. Also, we have split each trajectory in 20-70 sub-trajectories in order to obtain more information about their shape. The features that we have selected in order to classify the trajectories are:

- Curvature in sub-trajectories
- Direction of sub-trajectory
- The sub-trajectories are represented by their principal component analysis (PCA) coefficients.

In order to compute the curvature of each segment we have used the Hermann and R. Klette's formulas [31]. We projected each trajectory in the x-, y-, z- axis and we calculated the curvature in each 2D space. In figure 6.4: we have plotted the circle

trajectory in each axis: x = red circle, y=green circle and z = yellow circle (figure 6.4).

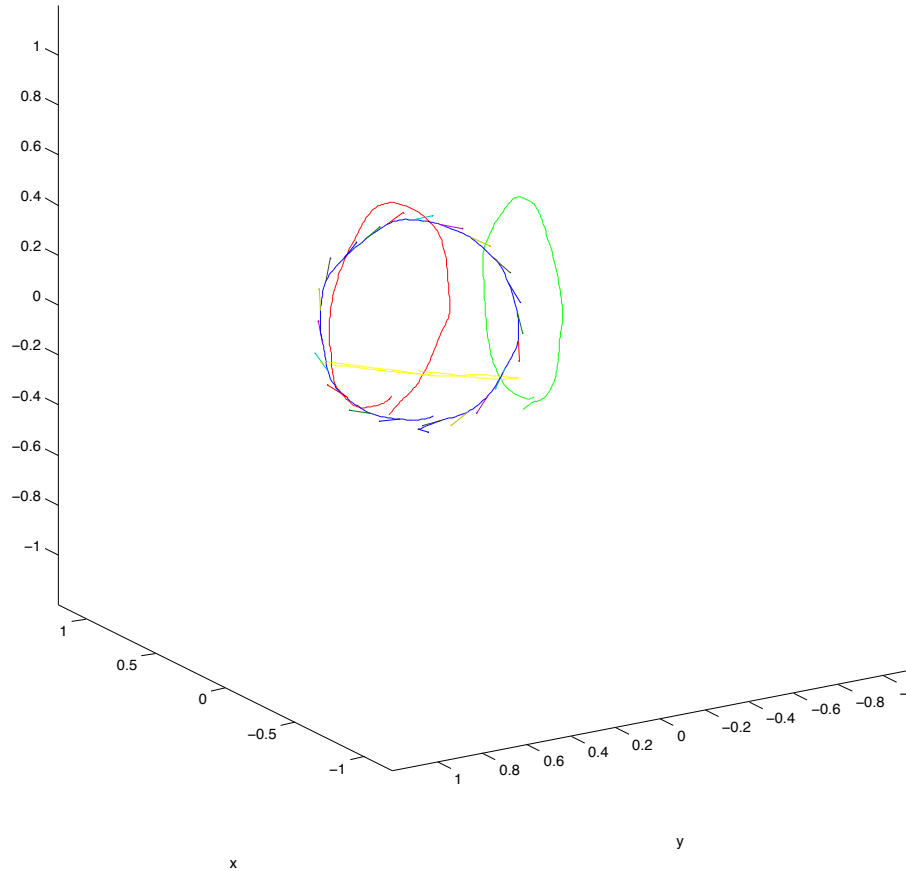


Figure 6.4: Projection in the 3 planes

Using the curvature analysis we extract the curvature of each segment in each dimension and create a vector of  $3 \times 30$  segments = 90-dimensional curvatures for each trajectory. Then, for each segment, we calculate the unit normal direction vector and we sum up all of them in order to generate the overall direction vector of the trajectory. This feature provides us with the direction information of each segment relative to the overall direction of the trajectory. Since the new feature space for each trajectory is

120 dimensions, we use the PCA coefficients to decrease the dimensionality. In figure 6.4 we have selected the 3D line that is produced from the robotic arm's end-effector and we are generating the overall direction vector from all the segments.

Before starting analysis and comparing the trajectories by using the DTW, as we did in our previous work, [62], we have to identify first their base shape and direction. For this reason, we classify each trajectory in 6 different classes in order to help us with the alignment phase using DTW and then estimate the error of each trajectory in each direction condition. The importance of our trajectory classification stage is that the system can adapt to the user's range of motion. Since we have preprocessed the trajectories, it allows us to have a scale and time invariant classification phase. That property enables the system to calculate outcomes from different personalized human variations and compare directly the user shape-trajectory data.

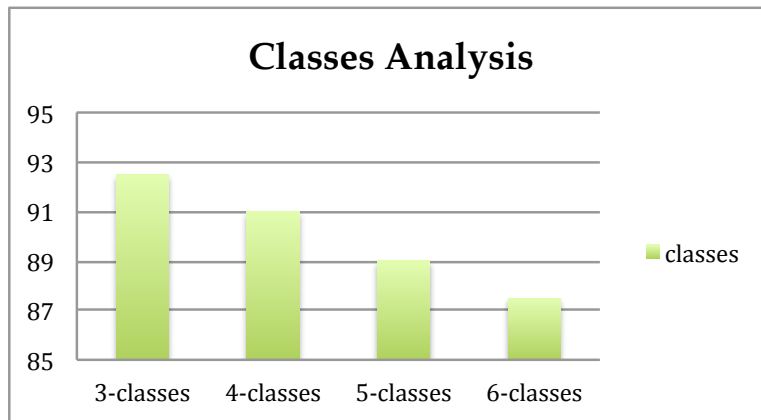


Figure 6.5: Accuracy estimation for different number of classes.

In the below diagrams we have tested the accuracy of the trajectory classification according to the number of classes and the number of segments that we split each trajectory into. Figure 6.5 shows the accuracy of the classification algorithm when the number of classes increases. The more classes we insert in our Multi-Class SVM

classifier, the less accurate it becomes. This is the obvious result when you have to deal with classification problems as the feature space limits the accuracy when the number of classes increases. In this experiment we have split each trajectory in 40 segments.

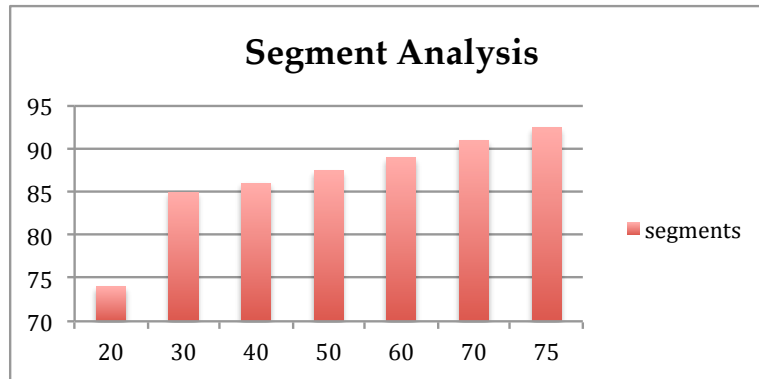


Figure 6.6: Accuracy estimation in comparison to the number of segments.

Figure 6.6 depicts the accuracy of the classification relative to the number of segments in each trajectory. As we mentioned before the features that we have selected have been applied in the sub-trajectories, so this parameter (number of segments) plays an important role in the classification process in order to define the shape and the properties of each class. For these kind of experiments, we used 6-classes and tested the classification accuracy with different numbers of segments (20-70). Bigger number of segments increases computational cost, but achieves better classification accuracy. In figure 6.7 we provide a combined representation of results when the number of segments and the number of classes increases. It's obvious from the graph that the number of segments in the classification problems increases the accuracy.

The results obtained indicate the robustness of the proposed method. Although our framework, at this stage, does not evaluate the user's engagement and satisfaction, it can successfully manage to evaluate the user's physiological performance through

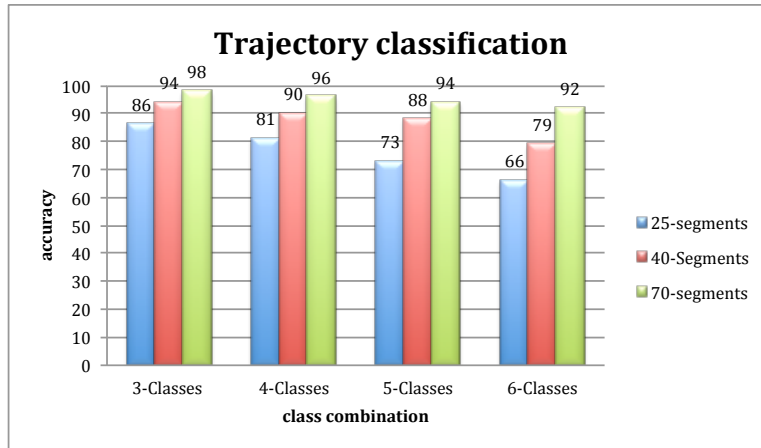


Figure 6.7: Trajectory classification accuracy. Comparison results for different number of classes and different number of segments for each trajectory.

a train-tested and well-defined exercise system. Thus, the system is able to record the patient’s exercise trajectories as he/she interacts with the game, to estimate the exercise score values regarding the range-of-motion, and to count the number of repetitions of each exercise.

## 6.2 Real-time Tele-rehabilitation Robot-based system

In this work, called "MAGNI 3D", we contribute technically in building an end-to-end prototype for the user’s real-time tele-rehabilitation experience by conducting experiments that allow the therapist to administer the patients remotely through a virtual exergame. Our algorithmic rehabilitation motion analysis contribution and the empirical studies of health-related information provide an important innovation in the Human-Computer Interaction (HCI) community. The upper limb health data captured from the user’s whole arm, coordinated with the data visualization of the user’s hand, performances, and scores, provides valuable information to physical and occupational therapists for the patient’s rehabilitation progress. The data collected will be used to provide input to the therapist for both monitorings the patient’s

progress over time and for offering recommendations about the next course of treatment. Surveys were given to the subjects to evaluate and deliver feedback on our prototype. The results and conclusions from the surveys will be incorporated in our system that can potentially be used in a clinical environment to improve the communication and interaction between a therapist and a patient in robot-aided therapy.

We evaluate the user’s exercises according to the prescribed therapist exercises using the Barrett WAM Arm in order to capture range of motion of the user’s upper limb. The purpose of this assessment tool is to motivate the user to perform some of the exercises, assigned in real-time by the therapist, using a 3D carnival-themed game. The contribution of this work is that wraps the Patient-Robot-Game (PRG) Interaction, Analysis and Database together in an integrated GUI that can be used in real-time by patients and therapists. Our research presents an innovative tele-rehabilitation system that tracks movements on a highly dexterous robotic platform to evaluate the range of motion associated with patient’s upper-limb.

### 6.2.1 Exercise Analysis

Dynamic time warping (DTW) [24] is a robust algorithm for measuring the similarity between two sequences which may vary in time or speed. We use the Multi-Dimensional DTW algorithm for the purpose of measuring the distance between the time-series representations of the exercise trajectories. These trajectories are the spatial coordinates received from the Barrett Arm corresponding to its location at any given time. The authors in [25] have used the same warping technique for sign language recognition and we incorporate the same analysis for measuring the exercise trajectory deviation.

Let  $R$  be an exercise trajectory. Here we can represent  $R$  as a time series  $(R_1, \dots, R_{|X|})$ , where each  $R_t$  is the spatial coordinate of the Barrett arm. Given a



reference trajectory  $R$  and patient trajectory  $P$ , DTW computes a warping path  $W$  that forms correspondences between features of  $R$  and  $P$ :

$$W = ((r_1, p_1), \dots, (r_{|W|}, p_{|W|})).$$

Here  $|W|$  is the length of the warping path, and pair  $(r_i, p_i)$  shows that feature  $r_i$  of  $R$  corresponds to feature  $p_i$  of  $P$ . The warping path follows rules as shown below:

- Boundary Conditions: This states that the first elements match ( $r_1 = 1, p_1 = 1$ ) and the last elements match ( $r_{|W|} = |R|, p_{|W|} = |P|$ ).
- Monotonicity and Continuity: This states that the alignment cannot go backwards and the alignment cannot skip elements ( $0 \leq r_{i+1} - r_i \leq 1, 0 \leq p_{i+1} - p_i \leq 1$ ).

The cost measure  $D(W, R, P)$  of a warping path  $W$  is the sum of individual local optimal distances  $d(R_{r_i}, P_{p_i})$ , corresponding to matching each  $R_{r_i}$  with the corresponding  $P_{p_i}$ . The local distance is the Euclidean distance between the corresponding features of the two trajectories. The DTW distance between trajectories  $R$  and  $P$  is defined as the cost of the lowest-cost warping path between  $R$  and  $P$ . We use this distance to calculate score for the game. This enables accurate score calculations for trajectory time series with similar shapes even if they may be out of phase in the time axis. We visualize the DTW optimal alignment to provide the therapist a better understanding of the patient trajectories' error deviations. Figure 6.8 shows the overall flow of the exercise analysis that is performed in our system. Our game score is computed as a combination of the DTW distance of the user's trajectory with reference trajectory and the number of targets hit by the user. The score is then normalized to be valid for all the different trajectories in our game.

The cost measure of a warping path is the sum of individual local optimal distances, corresponding to matching each with the corresponding. The local distance is the Euclidean distance between the corresponding features of the two trajectories.

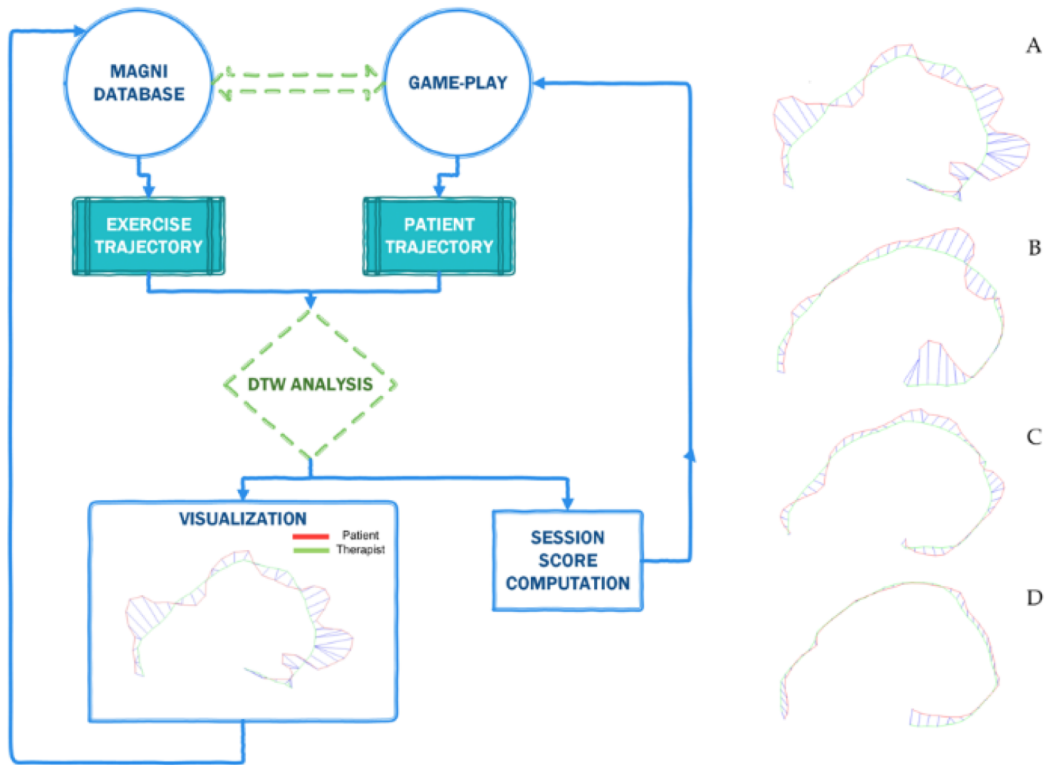


Figure 6.8: Left - Exercise Analysis Flow Diagram; Right - Patient's recovery progress in four sessions. The score error deviation over sessions are A.) 52.38, B.) 25.83, C.) 8.31 and D.)7.53.

The DTW distance between trajectories is defined as the cost of the lowest-cost warping path. We use this distance to calculate the score for the game. This enables accurate score calculations for trajectory time series with similar shapes even if they may be out of phase in the time axis. We visualize the Multi-Dimensional DTW optimal alignment to provide the therapist a better understanding of the patient trajectories' error deviations. Figure 6.8 above shows the overall flow of the exercise analysis that is performed in our system and an example of the patient's score and recovery progress over multiple sessions.

### 6.3 An adaptive robot-based therapy

A considerable amount of research has been conducted to implement a robotic rehabilitation system that adapts its behavior according to the patient's performance and physiological state. Rajibul et. al. [33], have presented preliminaries studies in developing a fuzzy logic intelligent system for autonomous post-stroke upper-limb rehabilitation. In their work, an intelligent system estimates the muscle fatigue of the patient and tunes the control parameters to generate different haptic effects. Badesa et. al. [3], have incorporated multisensory data in the control loop to adaptively and dynamically change in real-time the therapy. The aforementioned results demonstrate the potential to create a fuzzy system that adapts the robot's behavior and delivers personalized rehabilitation sessions. Similarly, in our work, we incorporated a fuzzy logic module that controls the haptic forces which are exerted upon the user.

Mandryk and Atkins [50] developed a fuzzy inference system to map physiological signals to the emotional experience when users were playing games. The training data is galvanic skin response (GSR), heart rate, and electromyography (EMG). The fuzzy inference system consists of input, output, membership functions, and if-then rules. In a fuzzy inference system, if-then rules determine the weighting factors, which are the parameters of the model. In their approach, these expert rules were the key point to influence the output of the fuzzy inference system.

This work presents a home-based robot-rehabilitation subsystem, called "MAGNI Dynamics", that utilized a vision-based kinematic/dynamic module and an adaptive haptic feedback controller. The system is expected to provide personalized rehabilitation by adjusting its resistive and supportive behavior according to a fuzzy intelligence controller that acts as an inference system, which correlates the user's performance to different stiffness factors.

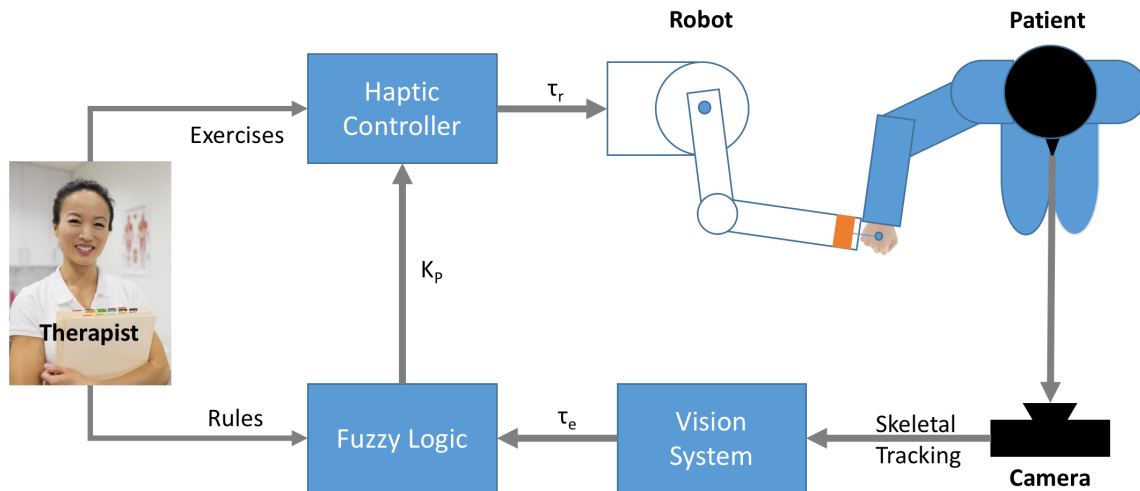


Figure 6.9: Proposed home-based robotic rehabilitation system.

As seen from Figure 6.9, the system is consisted by the following components: the RGBD camera sensor that provides skeletal tracking information, which is fed into the proposed vision system. Furthermore, the estimated torque ( $\tau_e$ ) of the user is passed to a fuzzy controller. The fuzzy controller acts as a high intelligence system that shifts the gains ( $K_P$ ) of the haptic controller, according to some abstract rules that have been defined by the therapist in a linguistic manner, and the performance of the user. As a result, the fuzzy intelligence system adjusts the control input signal ( $\tau_r$ ) of the robot to provide adaptive/assistive training.

### 6.3.1 Performace-based Assessment

In our case of rehabilitation robotic treatment, it is important to perceive the user's performance levels. For that particular reason, a performance model can be tuned according to the experts' decision making and be captured in a Fuzzy Inference System (FIS). The FIS's input directly comes from the user's efforts to accomplish the virtual reality game's goal. Most especially, the error deviation performing the

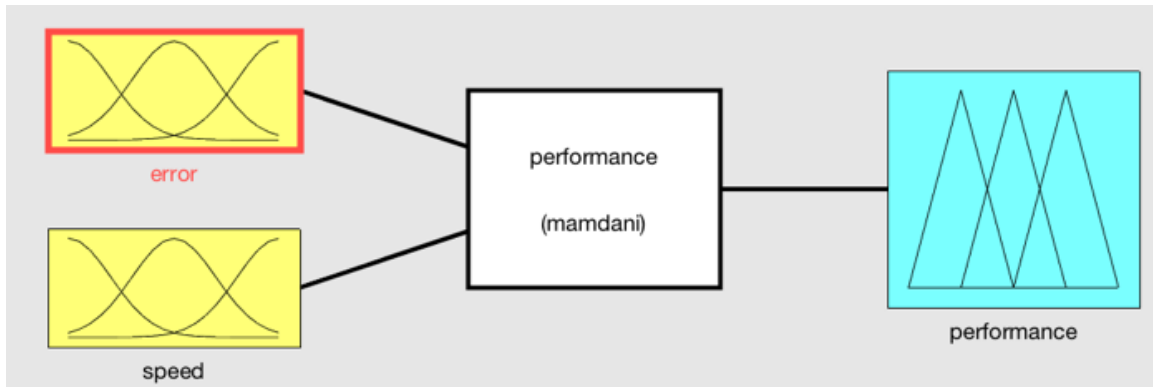


Figure 6.10: Fuzzy Inference System for the performance assessment.

prerecorded haptic path as well as the speed that the user accomplish it plays a critical role in our performance modeling. Figure 6.10 illustrates the intelligent performance estimator.

The Fuzzy Inference System utilizes a Mamdani-Assilian mechanism that employs fuzzy sets into the input and output membership functions (Figure 6.11.). The inference used min-max-centroid type and the fuzzified input values are correlated in rules utilizing the minimum T-norm (Figure 6.12).

On the other side, the output membership functions correlating to the rule is fined at the rule strength. The aggregation step utilized mamimum T-conorm to structure an output distribution from the output membership functions. The defuzzification step employes a centroid of area method to estimate the total crisp value.

### 6.3.2 Network Communication

To accomplish the communication between the Barrett WAM Arm Linux computer and the Unity 3D Game Windows System we established a low level software architecture scheme through real-time network socket inter-process communication

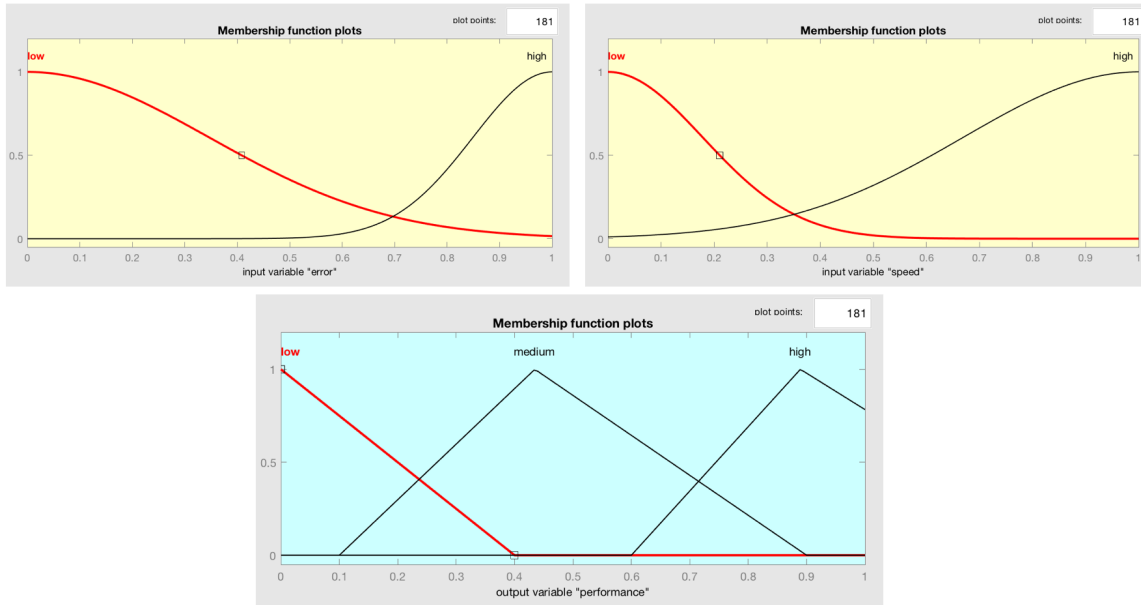


Figure 6.11: Input and output membership functions of Fuzzy Inference System.

1. If (error is low) and (speed is low) then (performance is low) (1)
2. If (error is low) and (speed is high) then (performance is medium) (1)
3. If (error is high) and (speed is low) then (performance is medium) (1)
4. If (error is high) and (speed is high) then (performance is high) (1)

Figure 6.12: Rule-based for Fuzzy Inference System.

across these two computers. Figure 6.13 shows the communication between Robot-Game by using *C++* and *C#* connection and the Game-Analysis through *C#* and *Java* languages.

#### 6.4 System Evaluation

One of the challenges for this system is the way that the haptic and the graphical interfaces are rendered. Most especially, the architecture that we have chosen allows us to build two independent correlated modules. The first module is the haptic control mechanism that exerts forces taking into considerations the rendering approaches that we follow. For example, tangential forces are exerted when we want to apply forces to

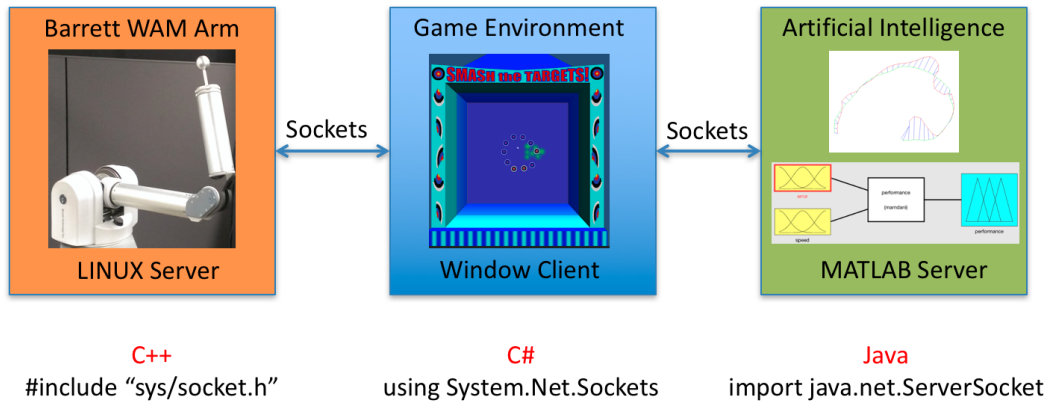


Figure 6.13: System Architecture.

a haptic path, and perpendicular forces are generating force-fields that constrain the user closer to the trajectory. Also, surfaces can produce normal forces that should be calculated to the robot's control interface.

The second module renders the visual interfaces in the Unity 3D platform and is depicting the robot's position to the screen. This architecture creates two parallel geometrical calculations, one for the haptic control and one for the visual illustration, and it's capable of sharing common robot's position data in order to produce the haptic/visual user perception. Such parallel calculations are unreliable solutions when serves medical applications. The dual streaming of position and force data can generate data discrepancies as well as delays to the system through lagging scenarios. Also, the independent system calculations may cause unstable networking pipeline streaming which provokes uncertainties.

For that particular reason, a better architecture that would administer such a real-time haptic and visual rendering modules will be the unit-module architecture. The calculations of the haptic forces as well as, the visual rendering modules should be generated only on one platform. For example, an idea would be the development of a plugin /driver that could connect and deploy the robotic arm with the Unity 3D

directly. All the geometrical haptic forces and the visual rendering/games features would have the ability to be generated and calculated from one physics engine, like the PhysicX that the Unity 3D platform comes with.



## CHAPTER 7

### Developed Applications in Human-Robot Interaction

In this Chapter, we present seven application of robots in assistive environments. The goal of these robotic applications is to share a work area and interact directly with users that need help. Their unique feature enables them to perceive their environment and individuals with sensors' aid as well as to incorporate intelligent algorithms that allow them to communicate with people, navigate autonomously and make decisions independently. Subsequently, experimental studies with users proved the acceptability and the trust factors in human-robot interaction research.

#### 7.1 Robot-based Tele-rehabilitation system

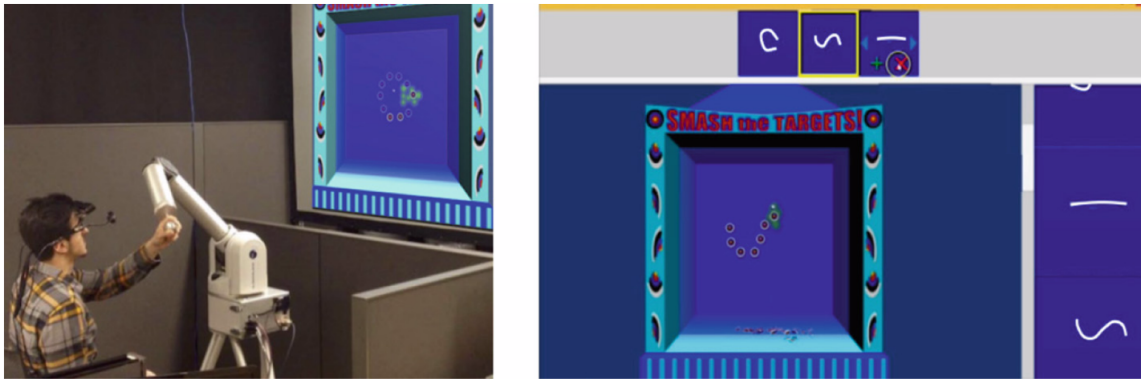


Figure 7.1: Tele-Rehabilitation System. On the right figure the therapist assigns exercises and on the left figure the patient performs them.

During the last two decades, robotic rehabilitation has become widespread, particularly for upper limb physical rehabilitation. Significant findings prove that

the efficacy of robot-assisted rehabilitation can be increased by motivation and engagement, which is offered by exploiting the opportunities of gamification and exergaming. This paper presents a tele-rehabilitation framework to enable interaction between therapists and patients and is a combination of a graphical user interface and a high dexterous robotic arm [29]. The system, called MAGNI, integrates a 3D exercise game with a robotic arm, operated by the therapist to assign in real-time the prerecorded tasks to the patients. We propose a game that can be performed by a patient who has suffered an injury to their arm (e.g. Stroke, Spinal Injury, or some physical injury to the shoulder itself). The experimental results and the feedback from the participants show that the system has the potential to impact how robotic physical therapy addresses specific patient’s needs and how occupational therapists assess patient’s progress over time.

#### 7.1.1 Experimental Setup and Results

In Human-Computer Interaction (HCI), role-play is a useful technique to develop an understanding of users’ needs and to evaluate design prototypes where access to users or environments is limited. In our work, we use therapeutic role-play [53] technique with our participants to gain feedback about the design of our system. Here we apply a procedure similar to the Goldfish bowl role-play technique where there are two participants in the role-play and many observers. We use two participants in the role-play, one as the therapist and the other as the patient. The developers of the system are the active and engaged observers who receive feedback from the participants on the design of the interface and the game-play experience. The system was tested with ten participants (three females and seven males) who actively took part in the therapeutic role-play technique while using our system and then providing feedback by filling out two surveys, namely MAGNI Game Survey and MAGNI

Therapist User Interface Survey. Both surveys contained Likert-like questions with 10 points scale with 1 being "Very Easy" and 10 being "Extremely Difficult." Our participants were all students from Computer Science Department with good vision and physical condition. Instructions were given to the participants to inform them what are the disabilities of stroke patients related to the arm movement, such that they could successfully participate in such a role-play. Each participant first played the game in the role of the patient and then used the user interface of our system in the role of the therapist.

The hardware that we require for our infrastructure is the Barrett WAM Robotic Arm, a Linux desktop computer to control the Arm, a Windows desktop computer to run the game and contain the database, one projector monitor, and networking hardware (LAN cable and router). All experiments of the User Interface were conducted on an Intel i5 4690 CPU @ 3.5GHz machine with 16 Gigabytes of main memory, running Windows 8.1 with NVIDIA GeForce GTX 780 graphics card.

<b>Label</b>	<b>Question</b>
<b>Q1</b>	Was it easy to understand the trajectory deviation between the therapist and the patient?
<b>Q2</b>	How easy was it to understand the interface during gameplay?
<b>Q3</b>	How easy was it to keep track of the patient's progress during gameplay?
<b>Q4</b>	Was the calibration phase easy to understand?
<b>Q5</b>	How easy was it to understand how to play?
<b>Q6</b>	How easy was it to understand how to interact?
<b>Q7</b>	How easy was it to control the game/hit targets?

Figure 7.2: Questions given during survey.

Figure 7.2 shows some of the questions that were presented during the two surveys. Figure 7.3 shows the mean value of the results taken from the surveys. This shows that the responses to our game and interface were mostly positive with the lowest mean being approximately 7.40. The higher values may be caused by the participants having prior experience with games and their work in other areas of Computer Science.

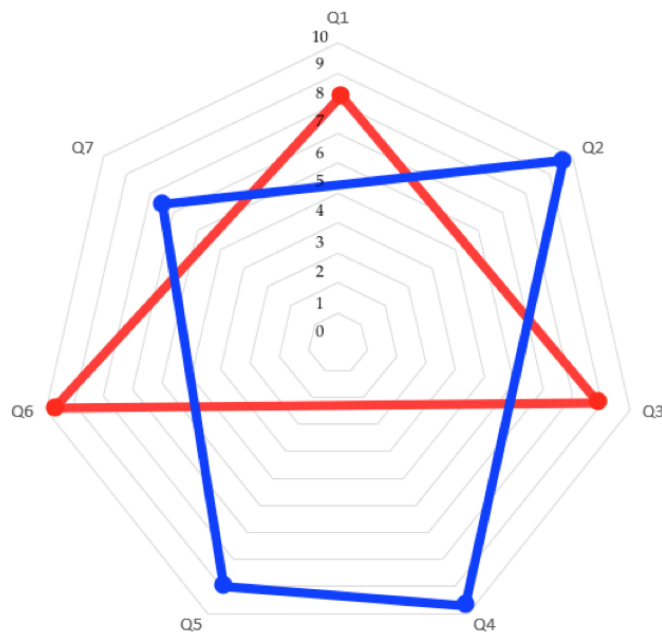


Figure 7.3: The mean value of the responses from the survey (The red lines correspond to the therapist interface answers and the blue lines with the patient interface).

The participants also provided additional comments on both the game and interface. The most common response was to grant the therapist the ability to manipulate the graphs. This way, the therapist can pan, rotate, or zoom in on the graph to see certain portions of the graph clearer. Another typical response was about the depth perception of the game. Even though the game was a 3D game, it appeared

to the participants as if it was 2D, making it difficult to understand the depth of the target. Lastly, participants wanted some additional information when adding or updating patients in the interface, such as user prompts and error messages that pop up and provide additional information.

## 7.2 Human-Robot Delivering Interaction in Assistive Environments

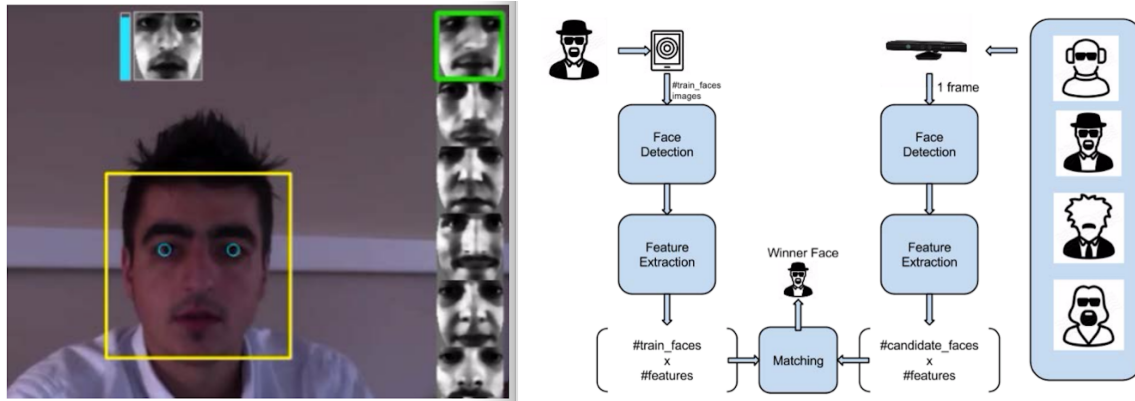


Figure 7.4: Face detection and recognition framework.

Integrating robotic platforms in smart home environments can improve the monitoring quality of daily activities. In this study, we explore a scenario where a robot provides a service to the users, which in our case is delivering a cup of coffee. The users place their order via an application, which at the same time captures a short video from their upper-body and their face to obtain information about their identity and to recognize them during the delivery phase. The proposed approach comprises three distinct steps [42]. At a first step the robot detects groups of people, then it captures information from their faces and their upper body and measures the distance between the probe and identifies the person with the higher probability. Finally, it approaches this person, performs an additional identification and delivers the cup of coffee. Through real-time preliminary tests under different illumination conditions, we verified that the robot could execute the task with high accuracy.

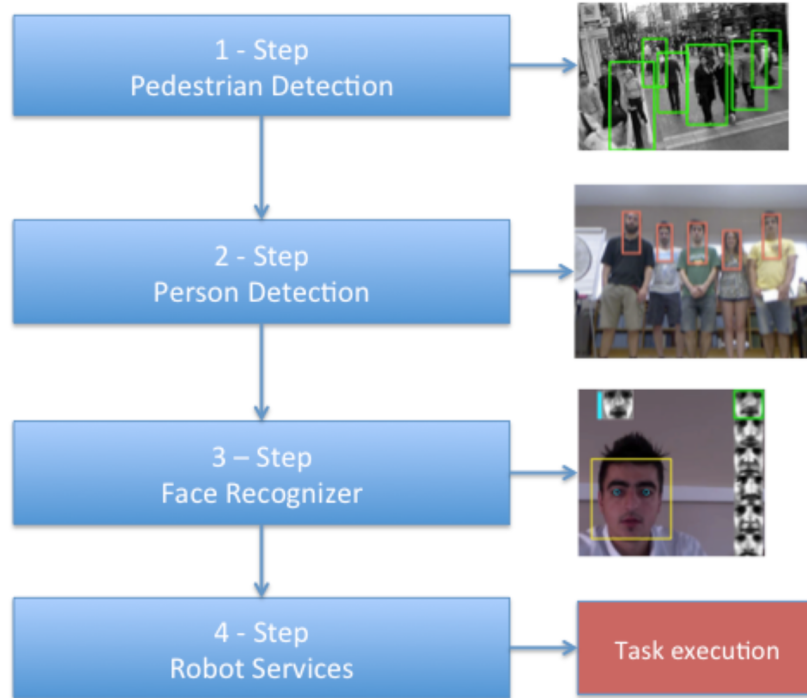


Figure 7.5: The task of our proposed framework for secure task execution.

### 7.2.1 Proposed Framework

The proposed scenario can be described as follows: A user gives instructions via an application and information from his face and t-shirt is saved. Our goal is to make a robot identify this person after a while and perform the required task (Figure 7.5). The proposed process consists of 4 distinct steps:

1. First the robot scans the indoor environment and detects humans (groups of people) to approach them (to come in close range e.g. 2.5 m). Then the robot gives instructions to the users to adjust their face and body orientation towards the agent.
2. Since all the users have aligned their faces and their bodies towards the robot camera, face detection is performed and their faces as long as t-shirt information is saved. Using both information acquired from the application and from the

face detection step we perform a similarity measurement algorithm and compute accuracy scores between the input image and the detected faces.

3. The aforementioned scores can serve as an indication of who are the people that have higher probability of being the correct match. The robot agent gives instructions to these persons to come at a closer distance (0.5 m) in order to make the final decision.
4. A second-level identification sub-system is then employed which identifies the person that needs to be served from the agent, with an accuracy of 93% for a maximum of 15 persons in the database.

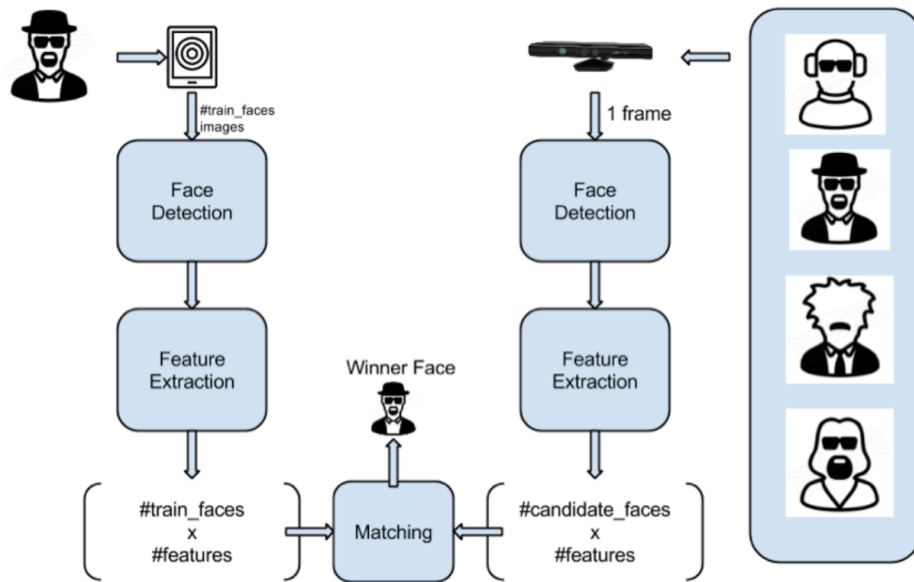


Figure 7.6: Application and Kinect feature extraction and matching for person detection.

In our study the target class is the face which was captured when the order of the service was placed. When the robot is ready to deliver the service, it scans for faces in the image plane and for each face, it performs a similarity measurement in order to



identify the person that placed the order. Having taken into consideration the nature of the problem we are trying to solve, we chose to perform feature extraction not only from the detected face, but also from the t-shirt that the customer is wearing in order to exploit color and texture information that enhance our decision (Figure 7.6). As a result, we chose to perform color histogram feature extraction and matching from the selected region (i.e. face and t-shirt). By comparing the histograms signatures of two images and matching the color content - in the RGB color space - we can recognize the customer from a group of people standing in a specific range of distances and angles. We conducted experiments with people standing in arbitrary positions and angles in front of the Kinect camera mounted on the robot and concluded that the ideal distances and angles lie between 1.5m and 2.5m and  $-20^{\circ}$  to  $+20^{\circ}$  respectively.



Figure 7.7: Face detection examples captured from the application and from Kinect camera of the robot.

In Figure 7.7 face detection examples are illustrated in images acquired from the application and the Kinect camera. In the left image, we perform color histogram feature extraction from the selected region of the customer, whereas in the right image we extract the corresponding histogram features from the face of each possible customer. We map the face of each person in the right image to a folder by using

spatial information in the image plane in order to construct the classes of the testing model of the one class classification problem.

At a next step, we perform a similarity measurement using the Euclidean distance between the values of the customer and those of the candidates. The person with the lowest distance is the one with the highest confidence of being the customer. In order to derive the decision of the proposed algorithm, we use a confusion matrix with one actual class and predicted classes, the number of which depends on the number of people in our image plane. A challenge that we faced was that the probe image and the testing images were captured from different cameras from different angles and under different conditions.

In the last step of the proposed identification framework, the person with the highest identification accuracy from the previous step, approaches the robot at a distance of approximately 0.5m, where an additional face verification technique is utilized. We measure the distance between the video captured when the order was placed and the videos in the database. The training set in this case consists of the faces of the users which are captured through a 10-second video before they enter the indoor environment.

Having acquired all the faces of the users the next step is to train the identification algorithm. For this process, we detected both the face and the eyes and obtained 10 faces per person from each video. To alleviate the vulnerability of the face identification procedure due to its sensitivity to changes in lighting conditions, face orientation and face expression, we applied the following pre-processing procedure:

- Geometrical transformation during which images are resized, rotated and translated in order for the eyes to be aligned. Information from the forehead, the chin, the ears and the background is discarded at this step.

- Histogram equalization: The brightness and the contrast of both the left- and the right-hand sides of the face are normalized independently.
- Smoothing: Image noise is reduced using a bilateral filter.
- Elliptical mask: Removes all unnecessary information from the hair and the background which was not discarded during the first step.

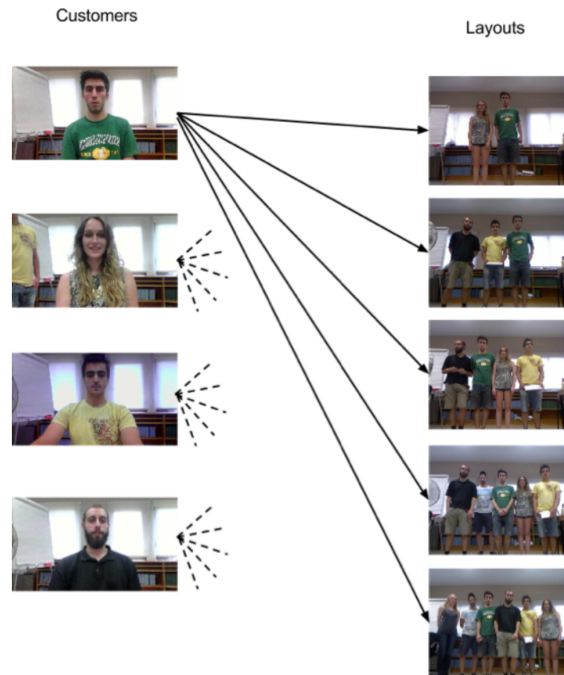


Figure 7.8: Experiments for each customer in different layout.

The above pre-processing procedure is important in our proposed framework because the captured images (before the participants enter the room and when they are in the room) may have significant differences in lighting illuminations, image resolution and color saturations. Finally the 2nd-step of the identification procedure is trained using images acquired from the videos of 20 different people. The more persons were inserted in the training set, the more training time was required. Note that a confidence level was also obtained for each face. If the confidence level was

below a threshold then the candidate face was considered as unknown and was added in the database which was then re-trained.

### 7.2.2 Experimental Evaluation

For the validation of our framework we conducted two types of experiments. For the first type we investigated the accuracy of the one-class classifier to accurately decide who is the person that requires to be served by the robot. Figure 7.8 shows the setting of the experiments we conducted that contained 4 possible customers and 5 layouts.

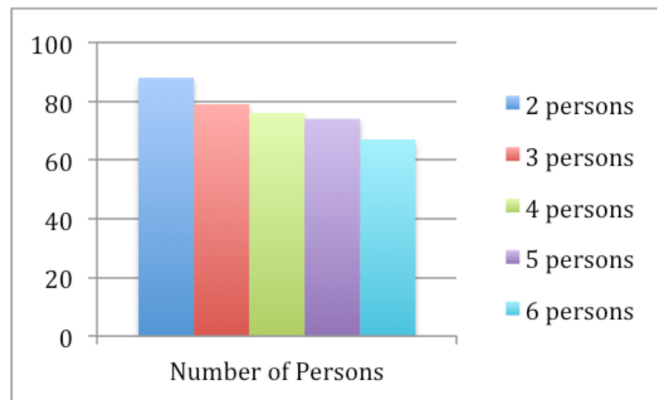


Figure 7.9: The person detection classification accuracy.

In each layout 2 to 6 people (including the customer) were standing in distances between 1.5m and 2.5m in front of the Kinect camera in 25 different combinations (i.e. people were changing positions in the setting). Our goal was to measure the customer identification accuracy in each layout in order to evaluate the performance of our algorithm. Figure 7.9 presents the performance rates of our proposed methodology for layouts with a varying number of people in them. As the number of people

increases in the image plane, the detection rate of our algorithm decreases linearly except from the 5-person layout where a small increase occurs.

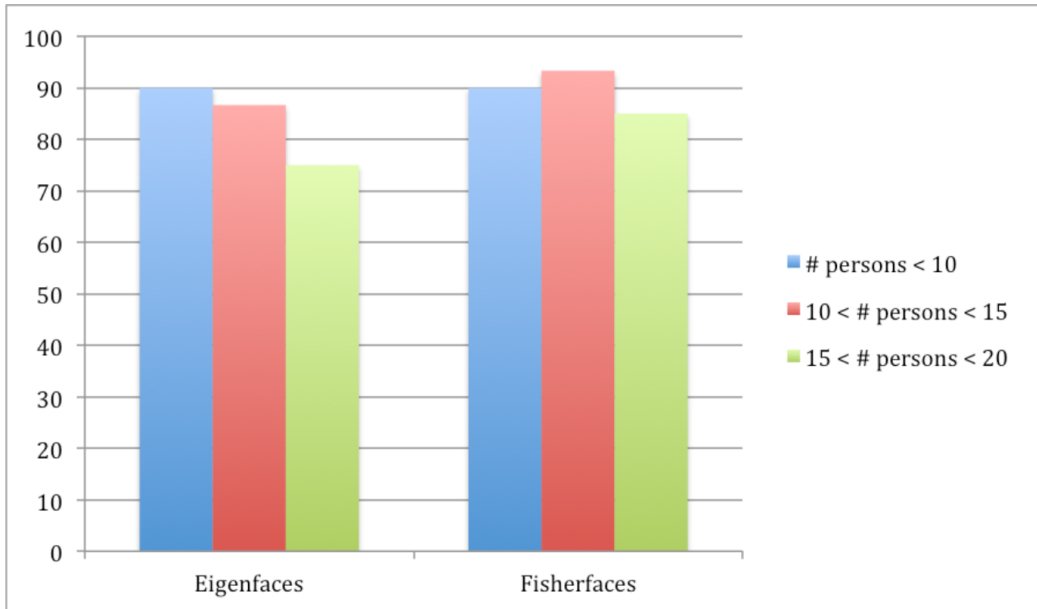


Figure 7.10: Comparison of the identification accuracy of Eigenfaces and Fisherfaces for different numbers of people.

For the second type of experiments we investigated the verification accuracy of 2 state-of-the-art algorithms. Figure 7.10 shows that the Eigenfaces approach [81] works with 90% accuracy when the number of people in the dataset is less than 10. Slightly better results are obtained using the Fisherface method of Belhumeur et al. [7]. When the number of persons in the database was 15, the accuracy was 93%. The results show that verification using Fisherfaces is higher than Eigenfaces as the number of people in the training set is increasing.

### 7.3 3D Mapping of Visual Attention for Smart Rehabilitation

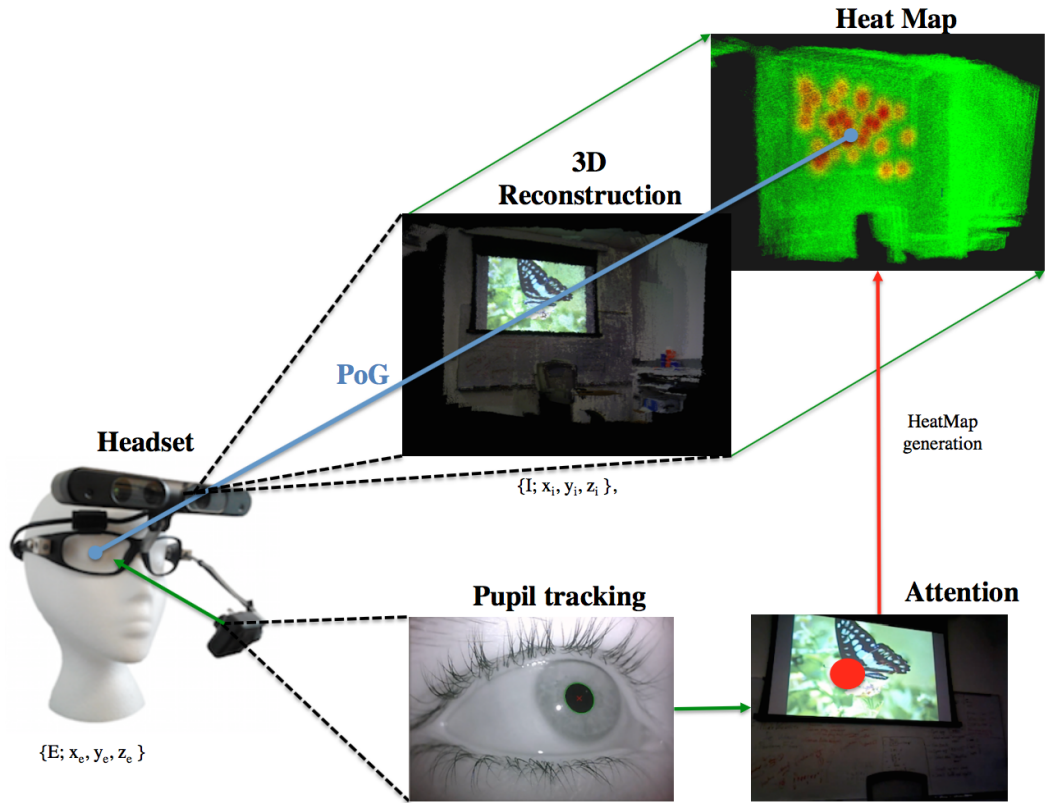


Figure 7.11: Head mounted eye-tracker and visual attention pipeline.

The estimation of human attention as input modality has been suggested as a method for an advanced human-computer interaction. With an increasing interest and development of augmented reality tools, the advent of Microsoft HoloLens glasses and increasingly affordable wearable eye-tracking devices, monitoring the human attention will soon become ubiquitous. Also, visual heat-maps have become very popular and simpler to create in the 2D space over the last few years. They are very compelling and can be effective in summarizing and communicating data. The innovation in our work is the implementation of visual 3D heat-maps of the real world combined with

advanced Computer Vision libraries [54]. Finally, we have incorporated the visual 3D heat-maps for rehabilitation purposes that deal with the loss of concentration in children with learning disabilities, or disabled patients to select items of interest for them across a room.

### 7.3.1 Visual 3D heat-maps generation

We model the application of visual heat using concepts of physical heating and cooling, which provides an intuitive means for visualizing and reasoning about visual salience over a given time window. Our approach is able to account for gaze vector error, visual fixation duration, and shifting attention through tunable variables that can be adjusted online.

#### 7.3.1.1 Computing saliency heat

In order to apply heat to a given location of interest, we first must compute the distance of the point to the gaze vector. This can be solved using the 3D point-line distance formula

$$D = \frac{|(X_0 - X_1) + (X_0 - X_2)|}{|X_2 - X_1|} \quad (7.1)$$

where  $X_1$  and  $X_2$  are two unique points lying on a line in 3D space and  $X_0$  denotes a given point of interest. We then wish to assign a saliency heat value in the range of  $[0, 1]$  with a gaussian function

$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}} + d \quad (7.2)$$

where  $a$ ,  $b$ ,  $c$ ,  $d$  set the curve's peak, horizontal offset from 0, bell width, and vertical offset from 0, respectively. Our goal is to assign a heat value from  $[0, 1]$

centered around a distance of 0 to the gaze vector, thus we set  $a = 1$ ,  $b = 0$ ,  $d = 0$ . The bell width,  $c$ , we leave as a free parameter which is set according to the error of the gaze vector. In our headset approach, we are able to estimate this parameter given the known hardware and user calibration errors. It then follows that we assign heat to a given point  $i$  using the following formula

$$H_i = e^{-\frac{(D)^2}{2c^2}} \quad (7.3)$$

where  $D$  is computed using (7.1). Combining (7.1) and (7.3) to achieve a final heat formula for a point  $i$  given the gaze vector defined by points  $X_1$  and  $X_2$  with error constant  $c$  yields

$$H_i = e^{-\left(\frac{|(X_0 - X_1) + (X_0 - X_2)|}{|X_2 - X_1|}\right)^2 / 2c^2} \quad (7.4)$$

### 7.3.1.2 Cooling salient locations over time

In real-time applications, we wish to decrease the perceived saliency of objects or points over time if the user’s attention shifts toward another location. We also want to decrease the contribution that short duration gaze vectors have on our perception of saliency, such as those measured during saccade eye motion. Intuitively, we model this “cooling” of 3D saliency map locations using basic heat transfer equations.

Recalling Newton’s Law of Cooling, the temperature of a cooling object at time  $t$  is given by the function

$$T(t) = T_e + (T_0 - T_e)e^{-kt} \quad (7.5)$$

where  $T_e$  is the temperature of the surrounding environment and  $T_0$  is the initial temperature of the object. The constant  $k$  controls the rate of cooling based on the



material properties of the given object. In our application, we define the temperature of a point with no relative saliency to be equal to zero, thus we set  $T_e$  to zero. We then rewrite Eq (7.5) as

$$T'_i = T_i e^{-kt} \quad (7.6)$$

where  $T_i$  is the initial saliency temperature of point  $i$  and  $T'_i$  is the updated temperature of the point after cooling for  $t$  seconds. Equations (7.6) and (7.4) are then used to generate dynamic 3D visual salience heat maps as shown in Figure 7.11 at the top left HeatMap image.

### 7.3.2 Hardware and Software

Information about the environment in front of the user is provided by a forward facing RGB-D camera, the Asus XtionPRO Live. This device provides a 640x480 color-image of the environment along with a 640x480 depth range image at a rate of 30 Hz. The two images are obtained from individual imaging sensors and registered by the device such that each color pixel value is assigned actual 3D coordinates in space. This provides a complete scanning solution for the environment in the form of 3D Point Cloud Library (PCL) [65], which we are using such us to implement the visual 3D heat-maps. The completed Headset is shown in Figure 7.11 at the top right image.

To capture user attention we used the popular starburst algorithm, which works by roughly estimating the pupil center, fitting an ellipse to the pupil "blob", and then refining the ellipse by considering pupil edge points, which lie on rays projected outwards from the center of the first ellipse. The 3D PoG can be obtained easily from the 2D point by looking up the 3D coordinates of the pixel in the point cloud data structure provided by the RGB-D camera.

## 7.4 Safe Human-Drone Interaction in Warehouse Environments

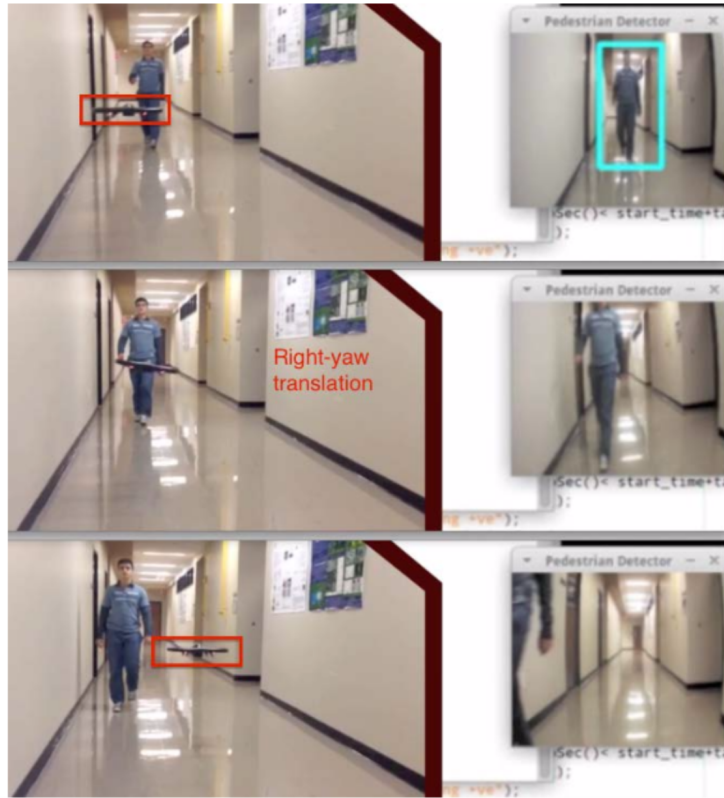


Figure 7.12: Human recognition and drone's maneuverability.

This work presents an Unmanned Aerial Vehicle (UAV), based on the AR. Drone platform, which can perform an autonomous navigation in indoor (e.g. corridor, hallway) and industrial environments (e.g. production line) [41]. It also has the ability to avoid pedestrians while they are working or walking in the vicinity of the robot. The only sensor in our system is the front camera. For the navigation part, our system relies on the vanishing point algorithm, the Hough transform for the wall detection and avoidance, and the HOG descriptors for pedestrian detection using SVM classifier. Our experiments show that our vision navigation procedures are reliable and enable the aerial vehicle to fly without humans intervention and coordinate together

in the same workspace. We are able to detect human motion with high confidence of 85% in a corridor and to confirm our algorithm in 80% successful flight experiments.

#### 7.4.1 System architecture

##### 7.4.1.1 Setup

For our experiments we used a quadrotor (Parrot AR.Drone) and a laptop MacBook Pro which is a quad core Intel i7-2410 @2GHz with 8 GB RAM memory and running Linux. For the video analysis we used OpenCV [9] which extract features from the sequence images that are captured from the frontal camera of the AR.Drone which the field of view is 92-degree and 640 x 480 pixels analysis. The high accuracy and robust recognition of the pedestrians is achieved using the OpenCV GPU module that utilize the GPU computational capabilities and is implemented using NVIDIA CUDA Runtime API. All the computational process happens in the connected laptop and the payload for the AR.Drone is very low with result the captured image to transfer quickly and the response time to be low. Finally for stability reasons we have access to the IMU data of the gyroscope to extract the yaw angular values.

##### 7.4.1.2 Visual navigation - Compass

Vision offers long-range sensing with low power and less weight, allowing smaller aerial robots to be build. By analyzing the images sequence we succeed to extract features and process them for all the task that we have done.

##### 7.4.1.3 Edge Detection

One of the first task in image processing is to determine the edges in each frame of the capture images. For this reason we use the Canny edge detector [11],

that is characterized for the low error rate, the well localized edges and the single edge point response using the *nonmaxima suppression*. Also, the effectiveness has been approved from using four filter to detect the horizontal, vertical and diagonal edges in the burring image and by combining a hysteresis thresholding to the neighbor ridges, can provide more accurate the real edges.

Because OpenCV implementation of Canny does not blur the image, we use a Gaussian blurring filter with a  $5 \times 5$  window to remove the noise and unimportant edges. From our experiments, we try to compensate the two parameters of the *high* and *low* thresholds that check the gradient value of a pixels, in a way to accept the pixels the edges of our corridors and work fine with the Hough transform algorithm when try to find the most prominent lines.

#### 7.4.1.4 Hough Transform

The detection of the perspective lines has been done using the Hough transform algorithm which use a voting scheme procedure to find the lines that lie more close to some points exported by the Canny detector. in order to accomplish that, we use the polar-coordinates where each line is expressed with a unique  $(\rho, \theta)$ , and a generic point  $(x,y)$  belongs to that line if satisfy the equation 7.7

$$\mathbf{x}\cos(\theta) + \mathbf{y}\sin(\theta) = \rho. \tag{7.7}$$

where  $\rho$  represents the distance of a line from the origin and  $\theta$  is the angle among the x-axis (figure 7.13).

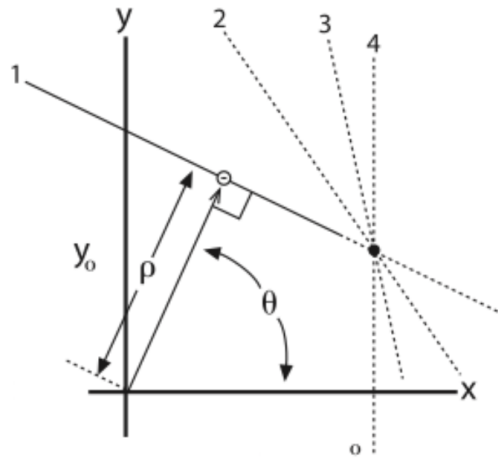


Figure 7.13: Each straight line has a unique representation in polar-coordinates  $(\rho, \theta)$ .

#### 7.4.1.5 Prune lines

From the previous algorithms we can find lines in our image and the number of these depends on some thresholds that influence the Canny edges and the Hough transform.

In figure 7.14 there is an example of the input image capture from the AR.Drone and the processed image.



Figure 7.14: Image after the Canny edge detector and have been applied the Hough transform algorithms.

Because some of these lines are outliers in our system, we try to prune them by thresholding their angles (in radians). Especially we eliminate the lines with angles  $\theta < 0.5$  for the vertical lines and the lines  $3 < \theta < 3.4$  for the horizontal.



Figure 7.15: Lines we are interested after the pruning procedure.

#### 7.4.1.6 Lines intersection

In order to find the line intersection from the lines exported by the Hough transform we use Least square problem which estimate the minimum distance of a point from a number of lines. In our environment (corridors) the lines intersect to a specific point, the vanishing point. So the way to find this point is to solve the mathematic system  $\mathbf{Ax} = \mathbf{B}$ , where the unknown parameter  $\mathbf{x}$  is the point that we find. This mathematic system can be solved for all the lines that we found in our image by the following matrices and using the SVD in equation 7.8. The solution  $\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{B}$  give us the  $(u, v)$  coordinate of the vanishing point in pixel.

$$\begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \cos(\theta_2) & \sin(\theta_2) \\ \vdots & \vdots \\ \cos(\theta_n) & \sin(\theta_n) \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_n \end{bmatrix} \quad (7.8)$$

In addition, every frame we estimate the vanishing point and we check its value with the previous frame estimations, due to outliers that occurs from the AR.Drone's abrupt maneuvers. This acts as a protect mechanism for the navigation part of our aerial platform, since the estimation of the vanishing point influence the drone's maneuvers. Also there is a boundary box in our image that the estimation of the vanishing point in that bounds consider as correct, otherwise we overlook this estimation and wait for a new one.

#### 7.4.1.7 Wall avoidance

The navigation of a micro aerial robot in indoor environments is not easy task as there are obstacles that prevent it to fly and unpredictable maneuvers can occur. Moreover, when a robot like AR.Drone that is not equipped with laser or obstacle detection sensor, can easily be trapped in an unknown environment.

The most challenging obstacle that cannot be observed are the walls. The drone should every time knows its position relative to the left and the right wall such as to avoid them while moving. This ability has been pre-estimated manually, by analyzing the lines' angular value from the Hough transform algorithm, by putting the camera to the two side-walls.

In addition, the lines' angular values limits can be estimated automatically, when the drone is positioned to the center of the corridor ( $B=D$ ), and by exporting the values of the angles  $A, B, C, D$  (Fig. 7.16). When the angle  $A$  is twice bigger than the sum of  $B$  and  $D$  then this mean that there is enough space the AR.Drone to avoid the human that are passing from the side free space without touch the walls.

The categorization and grouping of the lines can be done by measuring their angles relative to the horizontal lines. In Figure 7.16 there is an example of our experiments and from the line detection on the left and right wall we consider this



Figure 7.16: Representation of the hough lines in the left and right wall.

angular values as the bounds of the drone's side-workspace. Also, this experimental set is important for the drone, due to the air currents and aerodynamics [10] that result when in close proximity to a large obstacle. So, the drone can stabilize its position relative to the left and right wall in order to avoid collisions from the sides.

#### 7.4.1.8 Human detection

The detection of humans using cameras is a challenging task due to the wide range of body pose and other environment parameters. Dalal et.al. [18] has studied the performance the Histogram of Oriented Gradient (HOG) descriptors that works excellent relative to other feature detectors such that Viola and Jones. This features are computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy. Also, a linear SVM classifier is used in order to test the new feature detector in the MIT pedestrian database of 1800 pedestrians images of different poses and backgrounds.

That makes HOG feature very robust and highly recommended for our system in order to detect pedestrians and to protect them from the aerials vehicle's propels (Fig. 7.17).

There are different ways the aerial robot can avoid the collision with the pedestrians and the most easy way is to pass them from above, but this way is a little



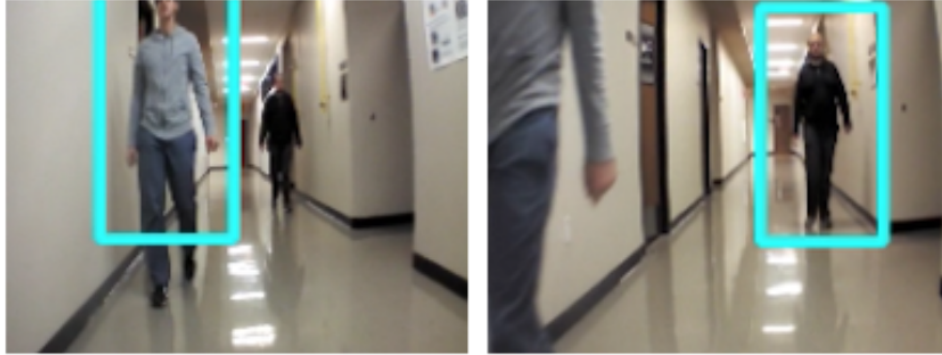


Figure 7.17: Pedestrian detection for secure avoidance proposes as the AR.Drone pass through them.

dangerous as it can collide with the human's face or with the ceiling. For that reason, we consider workspace area the boundaries of side walls, and this make the navigation more safe for both sides (human and vehicle). In Figure 7.18 the AR.Drone detects the pedestrian and estimate the most large free sideways area in order to avoid him. The wall detection are important for this task as the aerial vehicle compute the position of the pedestrian relative to the bounds of the left or the right wall, as mention in the section 3.3.

#### 7.4.2 Experiments

The navigation of the AR.Drone in indoor environments had many challenges cause the air drifts and the unbalanced yaw movements when we try to avoid obstacles, made the estimation of the vanishing point difficult as it depends from the angle of the Hough lines. The pruning procedure that has been applied on the the vertical and horizontal lines, as the AR.Drone change his yaw, rotates the capture image with bud consequences for our system. A good way to overpass it, is to compensate our captured images with the gyro measurement from the yaw coordinate of the AR.Drone



Figure 7.18: Sequence of frames that show the AR.Drone position (red rectangle) when detects a human being in front of it and try to avoid him.

and to apply a reverse rotation to the images. In figure 7.19 the red rectangle shows the upper results.

We evaluated our algorithms by letting the quadrotor fly towards the corridor to verify if it can detects and avoid the humans and the walls simultaneously. We did two different set of experiments, one with gyro compensation and one without. Also, in order to check the safety challenges of the AR.Drone for human collaborations, we did two more experiments, first with one person and second with zik-zak



Figure 7.19: Compensate the yaw angles of the AR.Drone with the captured images.

pedestrian layout such to compare and test the effectiveness of the drone's maneuvers.

	Non-gyro		Gyro-compensation	
	Total	Success	Total	Success
Run	10	3	10	8
Pedestrian crash	3		1	
Wall crash	4		1	

For our experiments, first, we flew the quadrotor 10 times on a corridor without gyro-compensation support and it managed to avoid pedestrian and walls only 3 times. Secondly we repeated 10 more flights with gyro-compensation support and the result in the above table prove that gyro-compensation is important to an aerial vehicle. The reason for these different result was quadrotor's behavior as a ground vehicle, cause the maneuvers did not influence our computer vision algorithms.

## 7.5 An Interactive Robot-based Vocational Assessment Game using Lego Assembly Paradigm

Lego construction task paradigms are utilized in order to develop logical and mathematical abilities through visuospatial memory. This study aims to assess the relationship between cognition and performance in a simulated industrial environment by employing humanoid robots to assess the stated metrics. This system proposes to develop a smart vocational assessment and intervention service system that assesses a worker's needs for training and rehabilitation in an experimental setup that simulates a factory.

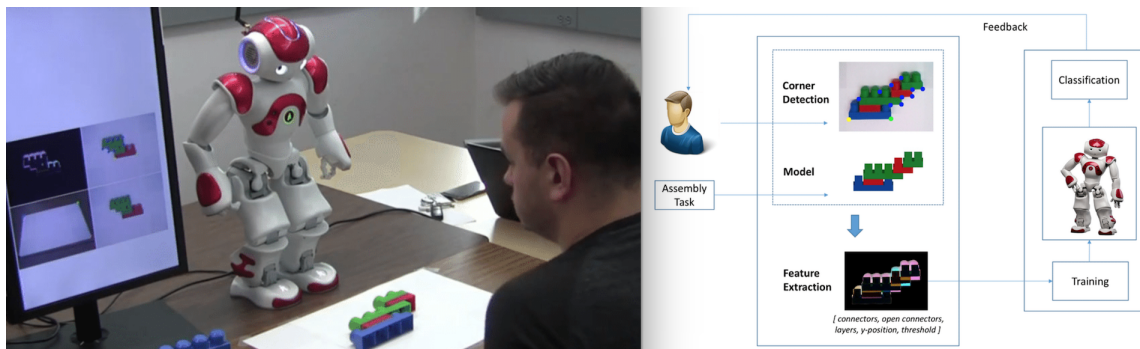


Figure 7.20: Interactive Human-Robot Learning Framework.

The proposed approach collects and analyzes multi-sensing data and recommends personalized interventions that can improve the performance of an individual worker [75]. In our implementation, Aldebaran's NAO robot gradually learns the features and thresholds needed to construct a decision tree that gradually learns the expected Lego model by interacting with the user. The results from 615 test samples show that the NAO robot is able to correctly identify the Lego blocks configuration assembled by the user with an accuracy 81% of the time. Finally, we discuss the

limitations of the proposed solution and we suggest future contributions that can overpass these limitations and boost the accuracy of our proposed solution.

### 7.5.1 Decision-Making

Decision-making is the process of identifying and selecting alternatives based on the values and preferences of the decision-Maker. The decision-making component is important for our vocational assessment system since the robot's visual features results impact the user's working memory evaluation. This subsection describes how the system chooses those features, based on probabilities calculated using robot's vision knowledge during the game.

#### 7.5.1.1 Feature selection

Since the NAO Robot does not have any machine learning library and does not contain any form of the compiler, a Decision tree classifier was trained in an external system using SciKit-Learn, converted to conditional python code, and imported back into the NAO Robot's source code. The decision tree classifier used five different features as are formulated in Equation 7.9. Figure 7.22 shows how these features are selected over the different tree's branches. A script was created that used a grid search on multiple decision tree parameters to determine the tree with the highest cross-validation score.

$$\begin{aligned}
x[0] &= \text{Total number of Lego connections} \\
x[1] &= \text{Total number of unconnected Lego connections} \\
x[2] &= \text{Number of layers of Lego blocks} \\
x[3] &= \text{Y position of the assembly on the placement board} \\
x[4] &= \text{Threshold value for normal differences}
\end{aligned}
\tag{7.9}$$

### 7.5.1.2 Interactive Machine Learning

Multiple machine learning applications involve human interaction that may incorporate input to learning processes such as correct labels, demonstrations, ranking or evaluation. These applications are using human-subjects to test and evaluate the correctness of the machine learning systems. For, the interactive machine-learning (IML) models permits the users to train, view or classify the models' outcomes and simultaneously to fine tune the wrong results [25].

### 7.5.1.3 Decision Trees

Decision trees are predictive models that correlate observations to conclusions. This machine learning technique is using a flowchart structure of branches that represent conjunctions of features and leaves that represent class labels.

### 7.5.1.4 Validation

The only way to generate training data was by interacting with our working memory assessment game. At the beginning, we managed to label 10 data samples and we used them to train our decision tree with the observed features. Then, we collected the robot's decisions while we were interacting again with the game. That step allowed us to correct the wrong robot's decisions and to use the corrected labeled

samples as training data for the decision tree generation. This iterative procedure helped us to collect a plethora of training data and increase the accuracy of the robot's decision-making system. It is worth mentioning that the threshold value (the fifth feature) is the absolute difference between our expected model and the original image.

The threshold value was originally easy to define. We could choose one value on the real number line representing this threshold as the difference between an accurate assembly and an inaccurate. As more complex assemblies were created it became obvious how the threshold value changed based on many features such as the shadow of the Y coordinate, the number of open connectors, and the number of layers. This initiated the decision to use a classifier.

As the decision tree was created over and over throughout time, various features were found to be redundant or unnecessary. Most of these features were removed, but as is indicated in the current decision tree, the first feature (the number of total connectors) is not used in this current iteration of the tree generation. It is possible that as more training data gets added, this feature will once again be used, but in the current tree it is unnecessary.

Figure 7.21 depicts the Cross-Validation history of the training data while we were collecting test data. It should be mentioned that the horizontal axis represents how many data samples we had at that point in time, and the vertical axis is the Cross-Validation score at that moment in time. The results from 615 test samples show that the NAO robot was able to correctly identify the Lego blocks configuration assembled by the user with an accuracy 81% of the time.

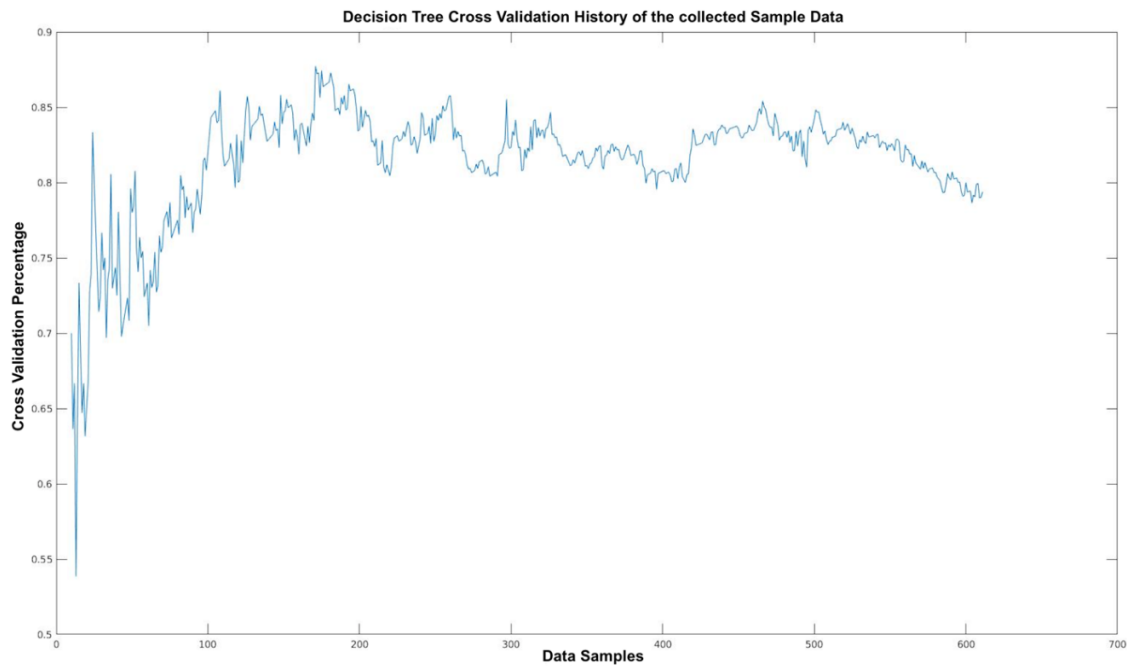


Figure 7.21: Decision Tree Cross-Validation evaluation from collected labeled training samples over time.

### 7.5.2 Experimental Results

In order to test and evaluate the accuracy of our system, we asked participants to play the memory system next to our NAO robot. The whole memory game consists of 5 Lego assembly tasks and Figure 7.22 shows only the first two tasks having incorrect and correct answers in order to evaluate the True Positive and True Negative samples. The NAO robot will not proceed to the next step if the user does not assemble the task correctly. For that particular reason detecting and verifying that the original image and the model Lego simulation if are similar, plays a crucial role in our memory game evaluation.



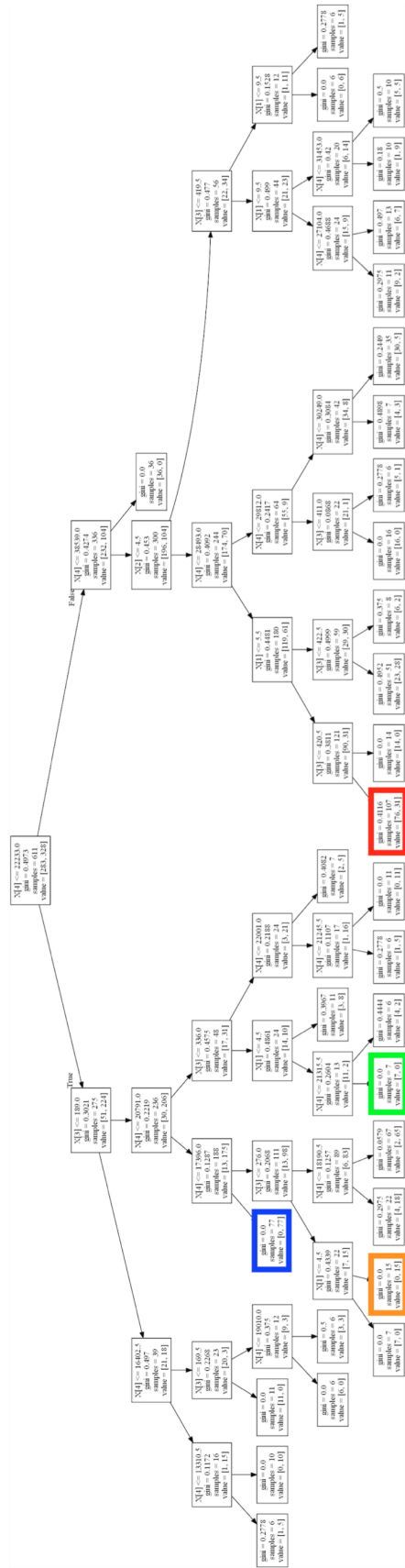


Figure 7.22: The decision tree model of the NAO robot's decision making system.

It is important to mention that Figure 7.23 colorizes four leafs with different colors (red, blue, green and orange) in the same sequence order. Each leaf represents a value of the True Positive or True Negative labeled variables with a probability distribution over the two classes. Finally, the *gini* impurity factor, which measures how often a randomly chosen sample from the classified subset, should be incorrectly labeled if it was randomly labeled accordingly to the distribution of labels in the subset, is less than 0.5 (0.4116) at the 1<sup>st</sup> incorrect sequence where the accuracy reaches 71%. An example of our memory assessment game is illustrated in this link: <https://www.youtube.com/watch?v=5-mle3REiS8><https://www.youtube.com/watch?v=5-mle3REiS8>.

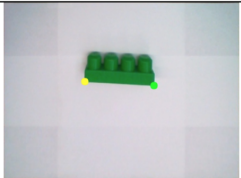


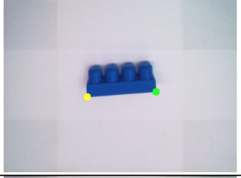

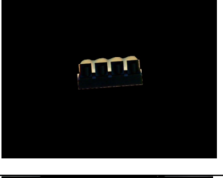
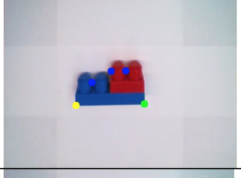
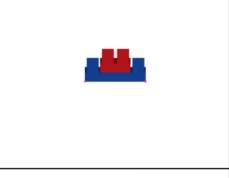

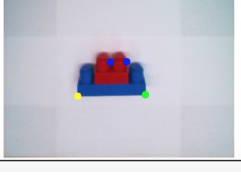
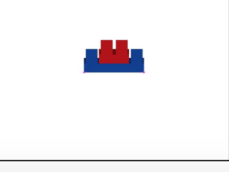
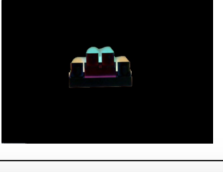
#	Task	Original Image	Model	Normal Differences	Accuracy
1	Incorrect User Input				<b>71%</b> gini = 0.4116 samples = 107 value = [76, 31]
2	Correct User Input				<b>100%</b> gini = 0.0 samples = 77 value = [0, 77]
3	Incorrect User Input				<b>100%</b> gini = 0.0 samples = 7 value = [7, 0]
4	Correct User Input				<b>100%</b> gini = 0.0 samples = 15 value = [0, 15]

Figure 7.23: Four memory-game steps showing the NAO robot's feature extraction and decision making results.

## 7.6 Using Virtual Reality to Program Robotic Manipulators

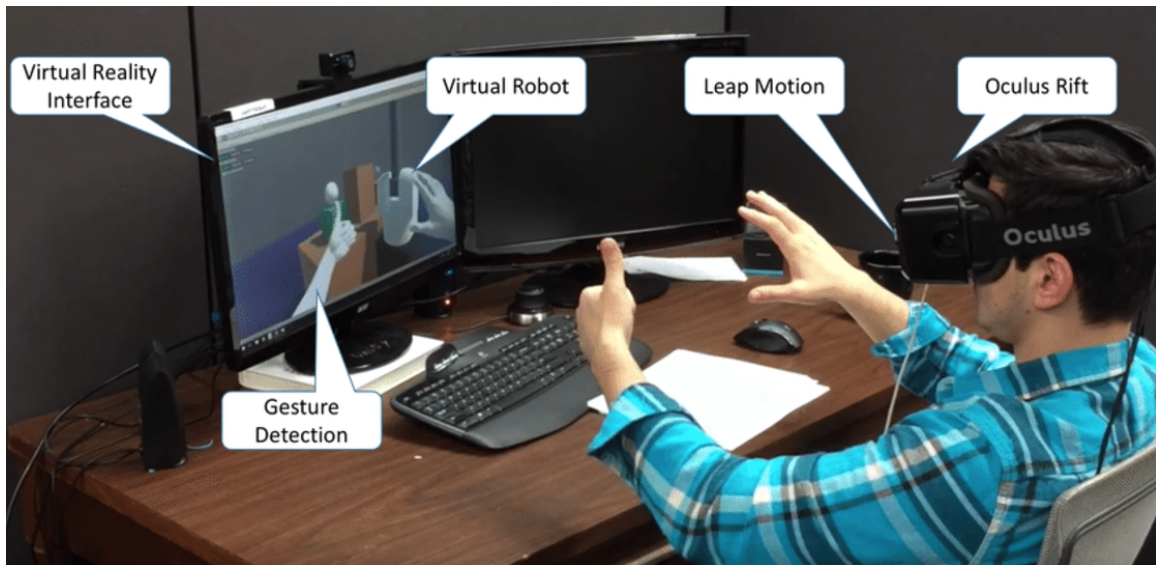


Figure 7.24: Real-time Human and Virtual Robot manipulation

This work describes a novel teleoperation interface system to program industrial robotic arms [14]. The system demonstrates the potential to create an interactive programmable interface that enables users with no prior knowledge of robotics to safely program mechanical manipulators with the use of Virtual Reality and the Leap Motion Controller. The system takes full advantage of the Leap Motion to navigate the virtual workspace of the robot that was created through the kinematic properties of the real robot (Appendix A). The implementation of the application was deemed possible by interfacing the Unity Engine with the Barrett WAM. Preliminary experimental results show the ability of the system to engage and train appropriately the user in robot programming.

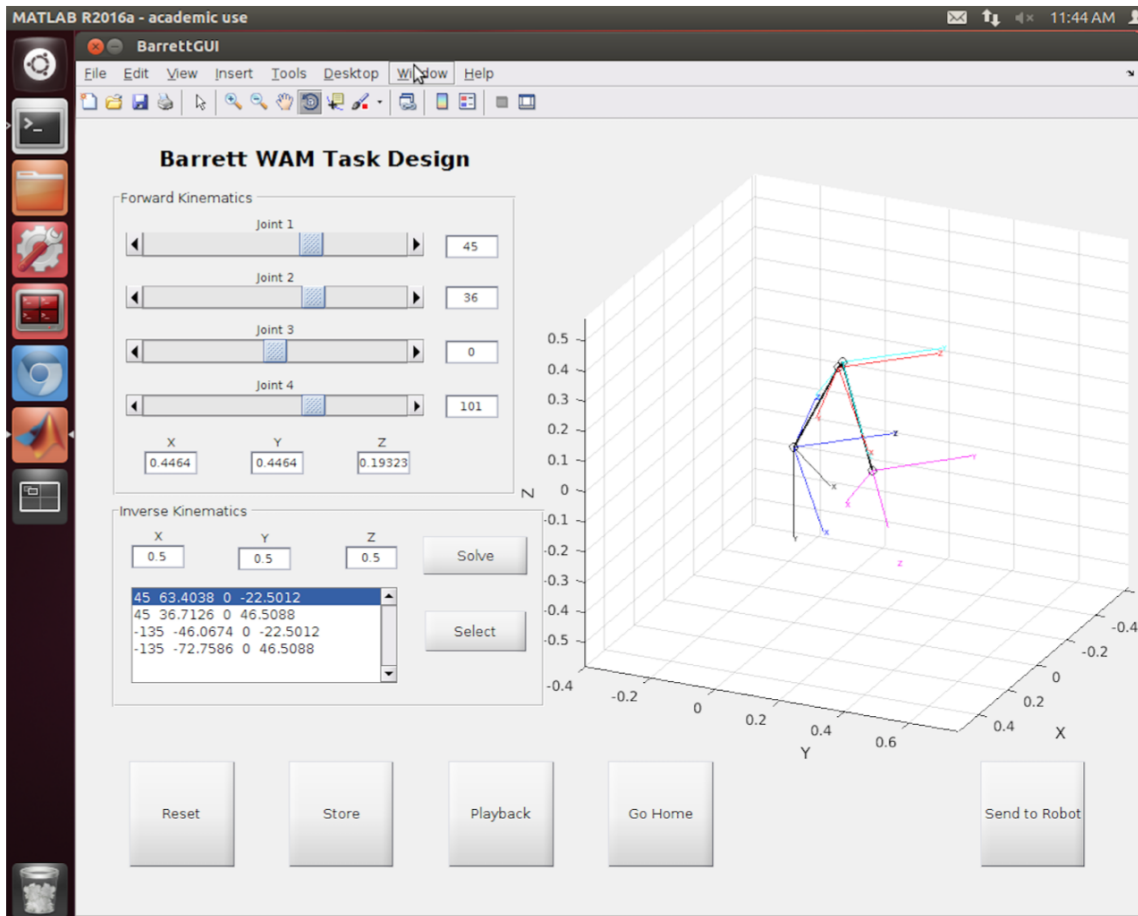


Figure 7.25: Barrett WAM task design Graphical User Interface.

### 7.6.1 Experimental Hypothesis and Case Study

To evaluate the usability and effectiveness of the proposed interface system, we created a hypothesis that compares our system with two different approaches that have already been established in the related bibliography. The comparison is meant to test the system in terms of safety and usability. To test the hypothesis we performed an experimental case study that provides measurements to support or debunk the hypothesis. Our goal will be to prove whereas that the VARM can provide a safe and intuitive interface to program robotic arms when compared to an application that

utilizes a 2D GUI (Figure 7.25) and an application that enables the user to teach the robot through direct physical interaction.

At this point, we have to mention the experimental configuration that was used to test the hypothesis and provide details about the test protocol that was followed throughout the process. The experiment involved a robotic challenge that would help determine the strength or weakness of the hypothesis. Figure 7.27 illustrates the scenario of the challenge. To elaborate more, the participants of the experiment were asked to use the three provided interface applications (Learning from Demonstration, 2D GUI, Virtual Environment) to make the robot move from the green box of Figure 7.28 to the red one without colliding with any objects in the real environment. Figure 7.26 depicts an overview of the three different interface systems.

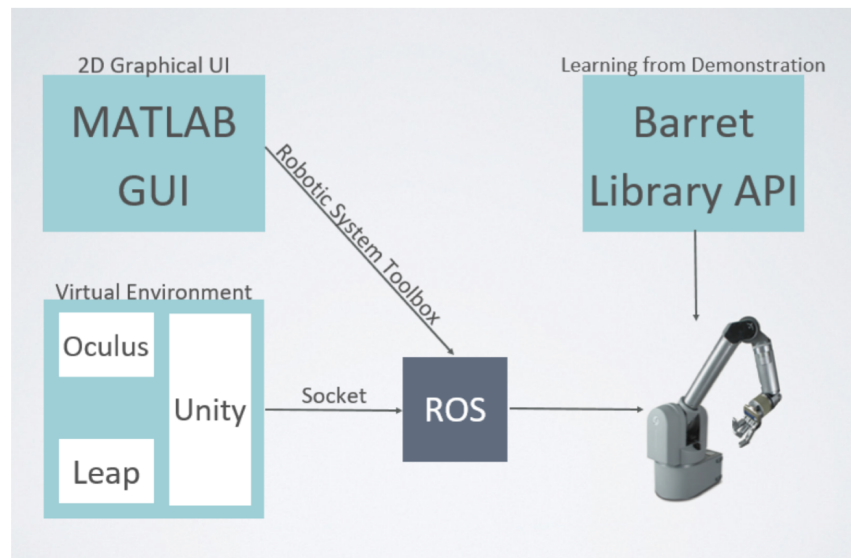


Figure 7.26: Overview of the three different interface systems

The hypothesis was tested by 11 different participants of both genders, with little to no background in the field of robotics, from the age of 20 to 25. The development team of VARM provided a quick overview of the experimental process and

explained the challenge that every participant must accomplish with all three interfaces. Furthermore, each participant was exposed to a tutorial run by a member of the team and was given a trial run to get a better understanding of every different system before they had a test run. Note that during the test run the team measured how much time it took each user to accomplish the task and how many times the robot collided with an obstacle.

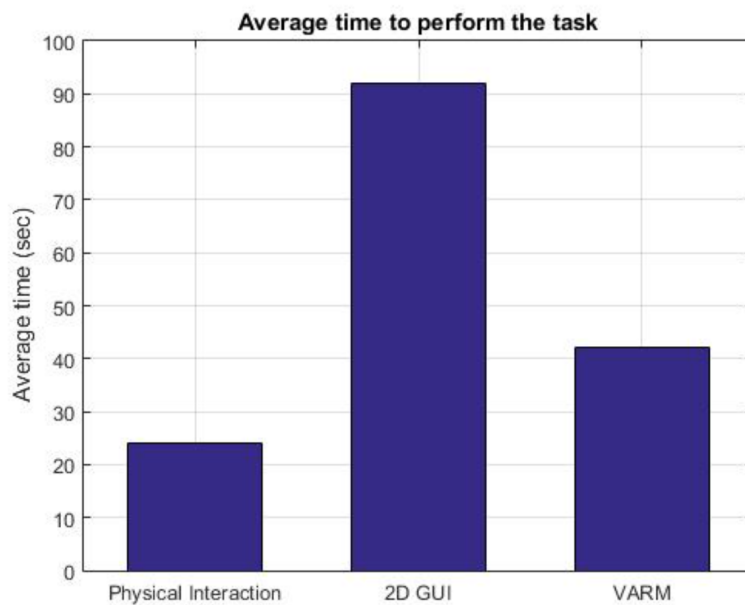


Figure 7.27: Average time to perform a task

### 7.6.2 Experimental Results

Results showed that the participants took less time to complete the task of the experiment by directly interacting with the real robot, when compared to the VARM and 2D GUI approaches. The VARM interface stood second in this comparison, while the 2D GUI was last. Figure 7.27 illustrates these results. Moreover, Figure 7.28 showcases that the same pattern was monitored from the recordings of the obstacle

collisions. To be more specific the average time to perform the task took 24 seconds by direct interaction, 92 seconds when using the 2D GUI and 42 seconds with the VARM. The average number of collisions when performing the task directly was 0, 6 when using the 2D GUI and 3 when using the VARM. Also, regardless of the measurements that were collected, it is clear that the safest approach is the 2D GUI, the most dangerous one is the direct interaction with the real robot and the VARM stands somewhere in between.

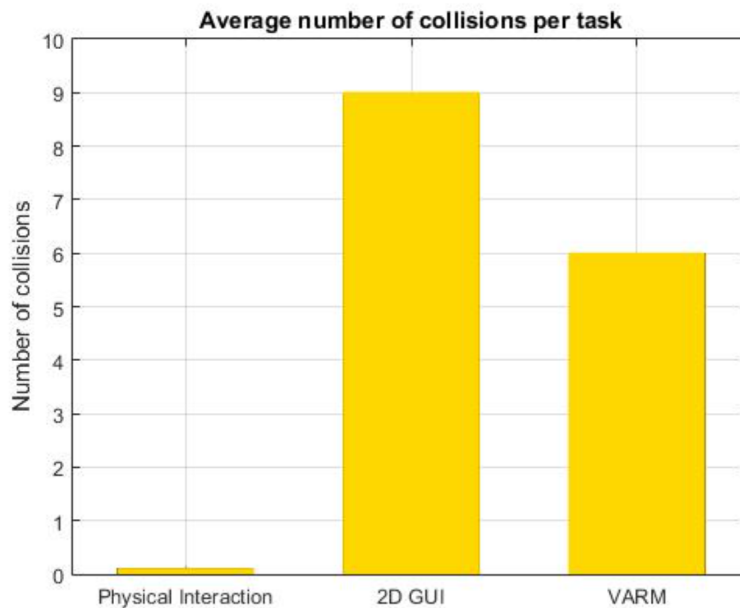


Figure 7.28: Number of collisions per task

As a final notice, from the numerical data that were collected, it was shown that the VARM interface stood in between the other two approaches in terms of both task design effectiveness and safety. That is to be expected since VARM tries to emulate a real interaction between a robot and a human, but it is lackluster in terms of freedom and believability, since no matter how realistic a virtual environment is, it still alienates the user from the real world.

## 7.7 Multimodal Analysis of Serious Games for Cognitive and Physiological Assessment

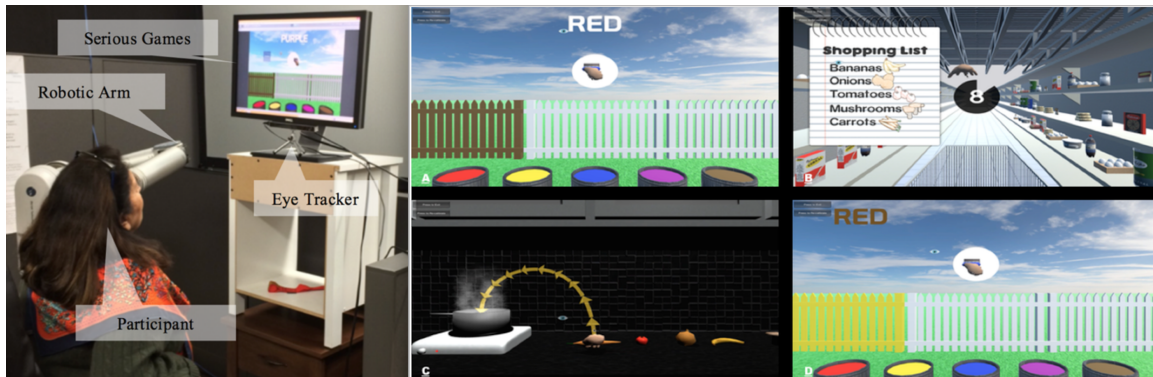


Figure 7.29: Design Serious Games for Human-Robot Assessment and Training.

Serious games usually refer to virtual games used for training, simulation, or education and can engage users in cognitive and physical tasks. The design of serious games may offer insights on users' cognitive and physical behaviors while they try to accomplish structured tasks. In this study, we concentrate on the design principles of serious games that can be used for assessment, which we employed for our own design [13]. To test our prototype, we conducted an experiment with control participants. Results from surveys, our collected game features, and sensor outputs were compared and analyzed with hypotheses based on previous research studies. Finally, we interpret the results of our experiment, and we describe issues and real life uncertainties that associate with sensor errors.

### 7.7.1 Hypotheses

For our experiments we plan to examine the serious game in a control setting of healthy individuals. These following three hypotheses are based on the assumption



that we will compare them with a future clinical population (individuals with disabilities, CP, stroke, TBI). We believe that the marginal difference would be greater in a clinical setting because of reduced inhibitory control, working memory, and issues with eye-hand coordination, that are found among individuals with disabilities.

- $H_1$ : For the comparison of L1 to L4, we expect the subject to have more mistakes and a slower move delay time for L4. The eye and move delay will be recorded in order to compare the two populations, and we anticipate that the control population would exhibit faster reaction and activity response times.
- $H_2$ : In L2, we predict that the number of iterations needed for the control population would be lower ( $\leq 2$ , no need for the visual cues) when compared to a clinical population because we expect them to exhibit stronger working memory abilities.
- $H_3$ : We hypothesize that the overall scores for L3 would be higher ( $\geq 75$ , on a scale [0 – 100]) in a control population as the participants do not suffer from any physical impairments, nor issues with hand-eye coordination.

### 7.7.2 Experimental Procedure

In order to test our hypotheses, we conducted an IRB approved experiment (*IRB No.: 2016-0121*) using human participants. After the process of informed consent, each participant received verbal instructions that described each level. Then, they were given the opportunity to ask questions or voice concerns. Next, the calibration process for the eye tracker occurred, which was followed by an interactive tutorial phase allowing each participant to play the serious games. The tutorial was a condensed version of the actual gameplay and the participants were given another opportunity to ask questions. An example of our experimental setup can be seen in Figure 7.29.

Table 7.1: Participants Demographics

<b>Participant Demographics - N = 12 Participants</b>
Age Range: 18-29
Gender: 6 Males, 6 Females
Handedness: 12 Right, 0 Left

Table 7.2: Results of the Serious Games

<b>Participant</b>	<b>Correct(L1)</b>	<b>Iterations(L2)</b>	<b>E-H C (Average)(L3)</b>	<b>Correct(L4)</b>
1	90%	2	83%	100%
2	100%	2	76%	90%
3	100%	3	78%	100%
4	100%	2	72.5%	90%

For our study, there were twelve total participants recruited. All participants were students and staff from the University of Texas at Arlington. Each participant was asked, after the experiment, to answer a questionnaire in order to gather demographic and self-report data (Table 1 shows their demographic information). The demographic information included age, gender, and hand dominance (which was useful because every participant had to interact with the robotic manipulator using their left hand). The rest of the survey was used to compare and analyze self-report responses to their physiological reactions throughout the game. For example, it collected the participants' perceptions about the individual level difficulty and their overall enjoyment.

### 7.7.3 Results and Discussion

Overall, we examined the results from four out of twelve participants in our analysis. Those four participants had either perfect or good eye tracker calibration. We have tried to re-do the eye tracker calibration for the other eight participants multiple times, but we were only able to achieve either moderate or poor calibra-

tion, which we believe it is not sufficient to produce accurate results. We should explain that this was due to the lack of data collected and hardware errors from the eye-tracking device. Moreover, we observed that the participants' facial feature deviations caused a discrepancy on the eye-tracking data. We also found that the experimental environment should be more controlled when the Eye-Tribe device is used for recording pupil dilation. The eye tracker did not consistently measure the eyes due to exaggerated movements from the participants and their distance from the device (over 60cm). In order to reduce experimental bias, we allowed a more natural setting that was unfavorable to the collection of pupillary data. In our experiment, the combination of cognitive and physical activities required autonomous control from the users. Previous work [51][52] controlled the participants' movement by limiting physical activity, which allowed for better eye pupil recordings. In the rest of this section, we will discuss results from the collected data and we will indicate some inconsistencies with our hypotheses.

The conducted experiments partially verified our first hypothesis ( $H_1$ ) as we found slightly more mistakes made in L4 instead of L1. However, move delay was not significantly different among the two levels. As seen in Figure 7.30, participants 1 and 4 showed a slower move delay for L4, but participant 2 and 3 were slower in L1. It was possible that they became more familiarized with the gameplay as they reached L4. This could have contributed to the faster move delay times found among participants 2 and 3. A possible solution could be a variation of the game-play parameters, such as color choices, playtime or game-level reordering and combination.

Following, the second hypothesis ( $H_2$ ) suggests that a control population does not need visual cues. From Table 7.2, only one participant exceeded our predicted number of iterations ( $> 2$ ) needed to collect all the items. Based on participant's survey responses, L2 was the most difficult level to accomplish due to the navigation

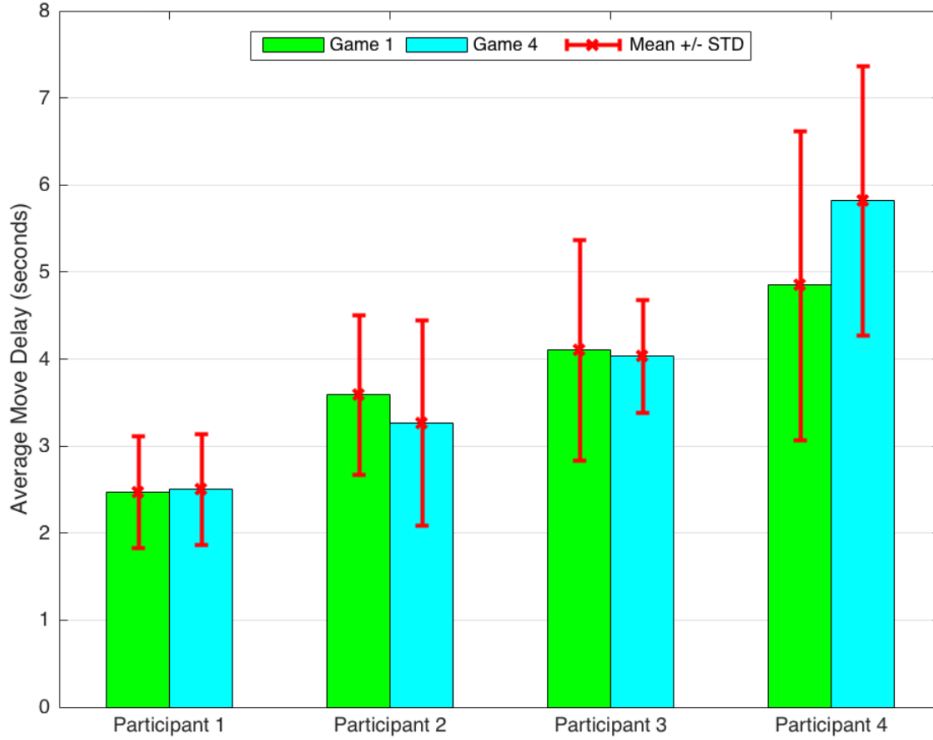


Figure 7.30: Average Move Delay for L1 and L4.

speed at the first iteration. Additionally, the participants found it challenging to recognize the items in the virtual environment as they had different perceptions for the shape and characteristics of each item. To solve this issue, we can modify their size or the game perspective view.

Lastly, for the third hypothesis ( $H_3$ ), most of the participants met our expectations ( $\geq 75$ ) as seen in Table 7.2. The outcomes for L3 show the average score for the five different 3D trajectories. Participant 4 showed a lower score, but this could be attributed to communication errors between the robotic arm and the game platform (Unity) software. A better computer gaming architecture could eliminate these errors, such as using *wxPython* implemented in *C++* instead of using Unity which is based on *C#* programming language.

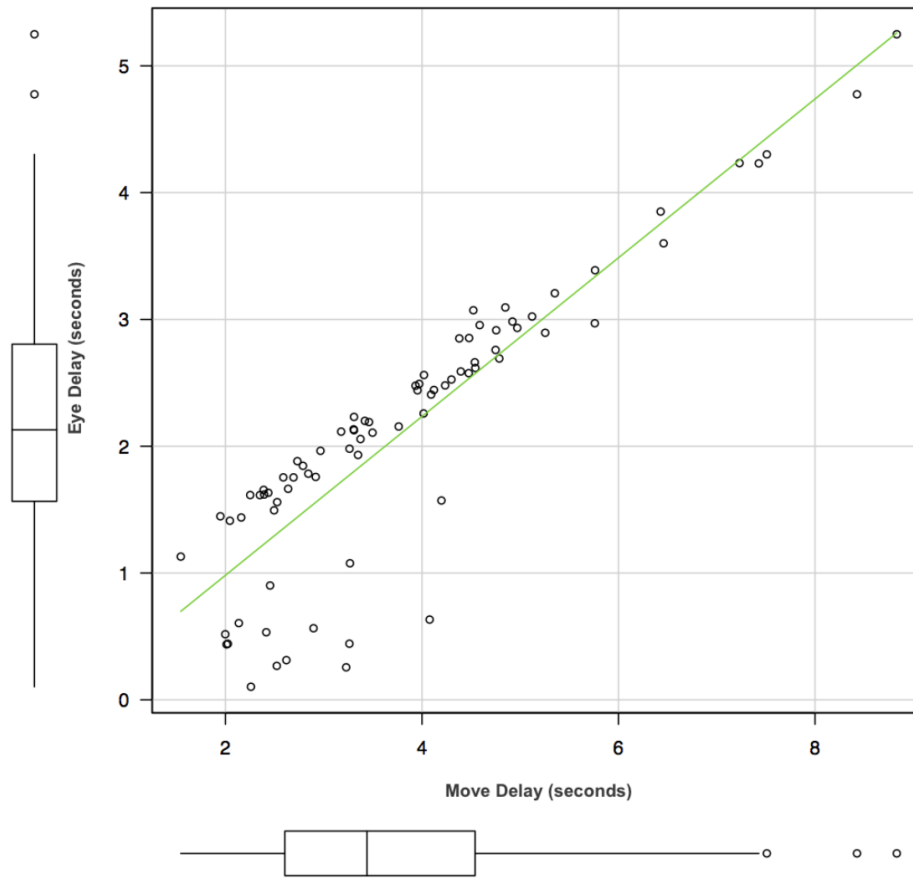


Figure 7.31: Pearson’s correlation between Move & Eye Delay.

One unexpected outcome that we found was that eye delay ( $M = 2.13$ ,  $SD = 1.08$ ) and move delay ( $M = 3.83$ ,  $SD = 1.54$ ) were positively correlated as seen in Figure 7.31. Using a Pearson’s  $r$  correlation, we found a strong correlation  $r(78) = 0.89, p \leq 0.001$ . For the four participants with working data, we discovered that as their eyes reacted slower to the stimulus, so did their physical response to choose the correct answer.

## CHAPTER 8

### Conclusions and Future Work

In this dissertation, I have presented my work in creating an Intelligent Multi-modal Upper-Limb Rehabilitation Robotic System. The main focus of this research was to investigate the development of a safe human-robot interaction assessment and training system by utilizing physiological, kinematic and dynamic modalities. This system placed the user in the robot's control loop, by feeding back his/her biomechanical, physiological and cognitive states. A proposed vision-based upper-limb monitoring system and a developed adaptive haptic guidance control mechanism involved human intentions to generate adaptive perception and behaviors for the Barrett WAM robotic arm. To facilitate this, a combined integration of computer vision, artificial intelligence, and human-robot interaction research had employed on the multisensing robotic platform.

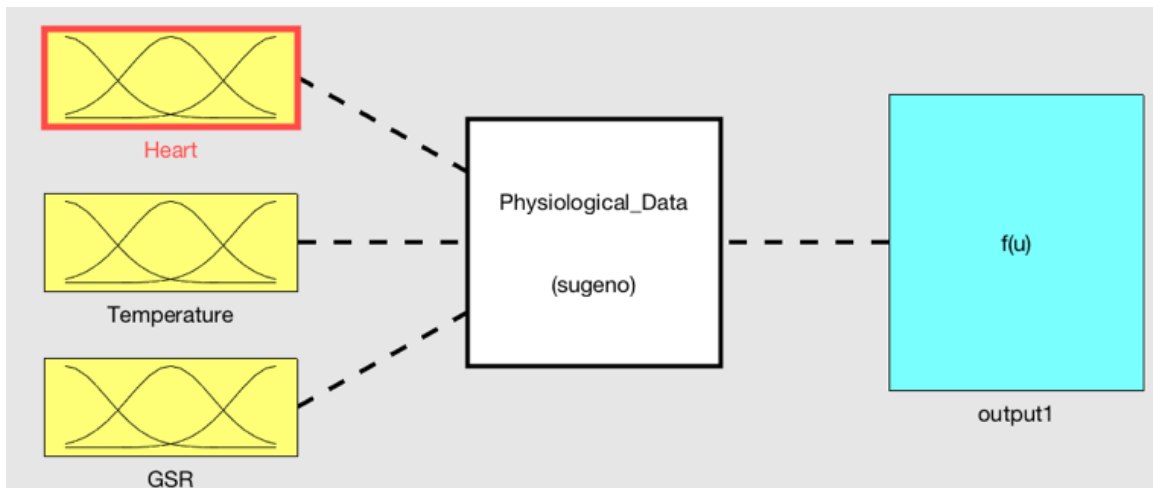


Figure 8.1: Fuzzy Inference System for user' arousal condition.

Our findings show that the computational methods can be used for a multimodal upper-limb robot-aided system. The system consisted of: (i) a virtual reality environment that assesses the user physiological and psychological stages; (ii) an interface capable to estimate patient performance utilizing motion analysis and pattern recognition methods; (iii) an unobtrusive method for reconstructing upper-limb kinematics during robot-aided tasks with end-effector machines using Microsoft Kinect's skeletal tracking is presented and experimentally validated; (iv) and an adaptive haptic guidance robotic controller is employed in order to modulate the complexity of the assigned motor tasks and increase the hand coordination abilities of the user.

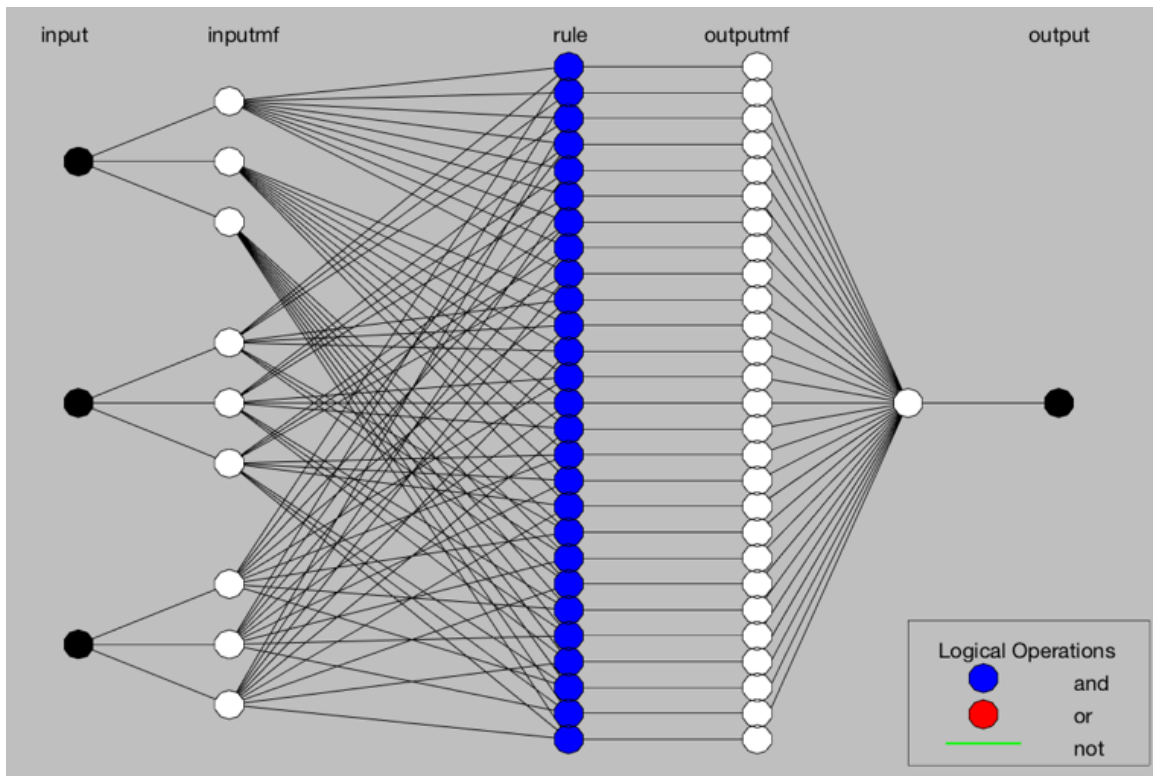


Figure 8.2: ANFIS system for membership function tuning.

In the future, we plan to model the Fuzzy Inference System according to the therapist interaction with the patient. Especially, physiological data will be incorporated as part of the user performance system which will be consisted from performance and physiological efforts. The patient's model will be generated as follow. The physiological data (Heart Rate, Skin Temperature, and GSR) in Figure 8.1 will be the input of the ANFIS [34] system in Figure 8.2. At every repetition, the therapist will ask the user about his arousal level. This adaptive neuro-fuzzy inference system will map the input characteristic to the user self-reported data. This process should be repeated for five sessions up until we have gathered user data and create a personalized model for his physiological performances. The user's performance level will be determined by the fuzzy inference system that will be designed following the task's characteristics (error and speed). Finally, it should be mentioned that the user's physiological profile data should be acquired by FDA approved biosensors as they provide ground truth data analysis



## APPENDIX A

### Forward and Inverse Kinematics of Barrett Arm

## A.1 Forward and Inverse Kinematics of the Barrett WAM Arm

In this Chapter, we provide a thorough analysis of the modeling process of the robotic arm. The inverse kinematics generation as well as the robot's design modeling was produced by my college's help, Michail Theofanidis.

### A.1.1 Forward Kinematics of the 4-DOF Barrett WAM

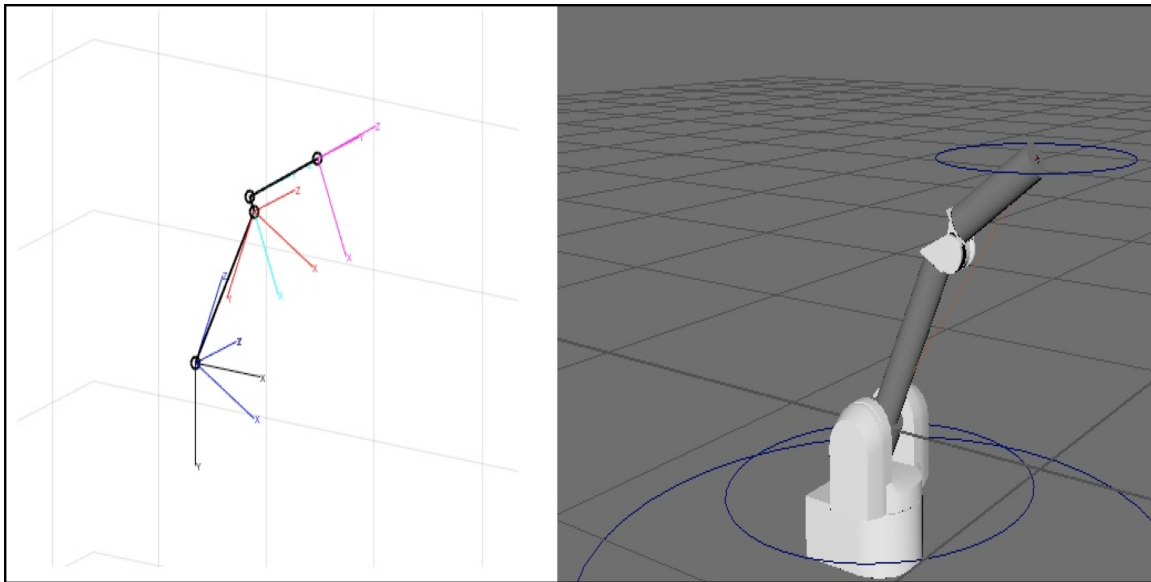


Figure A.1: Barrett WAM Arm kinematic model and the Virtual Robot in Maya.

Figure A.1 illustrates the kinematic model that was used to construct the virtual model of the robot. The frame placement of the kinematic chain is defined according to the DH parameters, which are provided in the Barrett WAM user manual.

Based on the DH table described above the homogeneous coordinate matrix of the frames can be derived according to the following matrix :

Table A.1: DH Table for 4-DOF Barrett WAM

i	$\alpha_i$	$a_i$	$d_i$	$\theta_i$
1	-90	0	0	$\theta_1$
2	90	0	0	$\theta_2$
3	-90	0.045	0.55	$\theta_3$
4	90	-0.045	0	$\theta_4$
e	0	0	0.35	0

$${}^{i-1}T_i = \begin{bmatrix} c\theta_i & -c\alpha_i s\theta_i & s\alpha_i s\theta_i & a_i c\theta_i \\ s\theta_i & c\alpha_i c\theta_i & -s\alpha_i c\theta_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.1})$$

Thus, the forward kinematics of the robot are derived by the following equation:

$${}^kT_e = {}^kT_0 * {}^0T_1 * {}^1T_2 * {}^2T_3 * {}^3T_4 * {}^4T_e \quad (\text{A.2})$$

Which determines that the resulting homogeneous transformation from the base frame of the robot to the robot's end effector frame:

$${}^kT_e = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x_e \\ r_{21} & r_{22} & r_{23} & y_e \\ r_{31} & r_{32} & r_{33} & z_e \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.3})$$

$$\left\{ \begin{array}{l}
r_{11} = c_1 c_2 c_3 c_4 - s_1 s_3 c_4 - c_1 s_2 s_4 \\
r_{12} = -c_1 c_2 s_3 - s_1 c_3 \\
r_{13} = c_1 c_2 c_3 s_4 - s_1 s_3 s_4 + c_1 s_2 c_4 \\
r_{21} = s_1 c_2 c_3 c_4 + c_1 s_3 c_4 - s_1 s_2 s_4 \\
r_{22} = -s_1 c_2 s_3 + c_1 c_3 \\
r_{23} = s_1 c_2 c_3 s_4 + c_1 s_3 s_4 + s_1 s_2 c_3 \\
r_{31} = -s_2 c_3 c_4 - c_2 s_4 \\
r_{32} = s_2 s_3 \\
r_{33} = -s_2 c_3 s_4 + c_2 c_4 \\
x_e = l_2 r_{13} + z_4 r_{11} + z_3 (c_1 c_2 c_3 - s_1 s_3) + l_1 (c_1 c_2) \\
y_e = l_2 r_{23} + z_4 r_{21} + z_3 (s_1 c_2 c_3 + c_1 s_3) + l_1 (s_1 s_2) \\
z_e = l_2 r_{33} + z_4 r_{31} + z_3 (-s_2 c_3) + l_1 (c_2)
\end{array} \right. \quad (\text{A.4})$$

### A.1.2 Inverse Kinematics of the 4-DOF Barrett WAM

In contrast to the forward kinematics problem, the goal of the inverse kinematics problem is to find a set of joint configurations given a particular end-effector position and orientation. The difficulty of the inverse kinematics problem arises from the fact that it depends on the physical configuration of the robot. Predominantly, non-redundant robotic arms can be solved analytically, while more complicated redundant manipulators require more advanced mathematical solutions such as artificial intelligent, pseudoinverse or transpose Jacobian solutions. However, these methodologies require algorithms with high time complexity costs, when compared with the analytical solutions.

The 4 DoF Barrett WAM is a kinematic redundant manipulator, which means that an analytical solution is impossible to exist. However, we solved the inverse kinematics analytically by setting the redundant third joint of the robot as a free parameter and thus effectively converting the redundant kinematic chain of the Barrett WAM robot to a non-redundant one. Note, that although this simplification allows to decouple the kinematic problem and solve it analytically, the solution of the inverse kinematics now exists for any particular x, y, z coordinate within the Barrett WAM workspace configuration, but only one fixed orientation that is defined by the value of the third joint angle  $\theta_3$ .

For the purposes of the virtual reality application, it was decided that the value of  $\theta_3$  should be zero. This alters the final position vector as follows:

$$\begin{cases} x_e = c_1(l_2c_2s_4 + l_2s_2c_4 + z_4c_2c_4 - z_4s_2s_4 + z_3c_2c_4 + l_1s_2) \\ y_e = s_1(l_2c_2s_4 + l_2s_2c_4 + z_4c_2c_4 - z_4s_2s_4 + z_3c_2c_4 + l_1s_2) \\ z_e = -l_2s_2s_4 + l_2c_2c_4 - z_4s_2c_4 - z_4c_2s_4 - z_3s_2 + l_1c_2 \end{cases} \quad (\text{A.5})$$

By taking the sum of the squares of the position vectors, the solution of  $\theta_4$  can be given from the equation:

$$A \tan \frac{\theta_4}{2} + B \tan \frac{\theta_4}{2} + C = 0 \quad (\text{A.6})$$

where,

$$\begin{cases} A = \frac{x_e^2 + y_e^2 + z_e^2 - z_3^2 - z_4^2 - l_1^2 - l_2^2 + 2(l_2l_1 + z_4z_3)}{4} \\ B = -(l_2z_3 - l_1z_4) \\ C = \frac{x_e^2 + y_e^2 + z_e^2 - z_3^2 - z_4^2 - l_1^2 - l_2^2 - 2(l_2l_1 + z_4z_3)}{4} \end{cases}$$

As a result,  $\theta_4$  has two solutions (an elbow up and elbow down solution) which are:

$$\theta_4 = 2 \arctan \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \quad (\text{A.7})$$

Moreover, to derive the solution of  $\theta_2$  we take the sum of squared the x and y position vectors. Note, that as with  $\theta_4$ ,  $\theta_2$  also has two solutions (shoulder up, shoulder down):

$$\theta_2 = 2 \arctan \frac{\pm M \sqrt{x_e^2 + y_e^2} - z_e L}{\pm L \sqrt{x_e^2 + y_e^2} + z_e M} \quad (\text{A.8})$$

where,

$$\begin{cases} M = (l_2 c_4 - z_4 s_4 - l_1) \\ L = (l_2 s_4 + z_4 c_4 + z_3) \end{cases}$$

Last but not least, the solution of  $\theta_1$  can be found from the x and y coordinates of the robot's end effector.

$$\theta_1 = \arctan \frac{y_e}{x_e} \quad (\text{A.9})$$

From the partial analytical solution described above it is important to mention that  $\theta_4$  has two independent solutions,  $\theta_1$  has an independent solution that can result in two different joint configurations and  $\theta_2$  can be computed by two different solutions that depend on  $\theta_4$ . This implies that there is a total of four different unique kinematic configurations for every Cartesian point of the robot's end effector. This phenomenon is typical for planar manipulators as the same elbow up/down position of the robot can be expressed with different joint configurations.

APPENDIX B

List of Publications

## B.1 List of Publications

1. Alexandros Lioulemes, Michail Theofanidis, Varun Kanal, Fillia Makedon, MAGNI Dynamics: A Vision-Based Kinematic and Dynamic Upper-Limb Model for Intelligent Robotic Rehabilitation, World Academy of Science, Engineering and Technology International Journal of Biomedical and Biological Engineering Vol:4, No:4, 2017.
2. Christopher Collander, Joseph Tompkins, Alexandros Lioulemes, Michail Theofanidis, Ali Sharifiara, Fillia Makedon, An Interactive Robot-based Vocational Assessment Game using Lego Assembly. International Conference on PErvasive Technologies Related to Assistive Environments, PETRA, Corfu Island Greece, July 2017.
3. Michail Theofanidis, Saif Iftekar Sayed, Alexandros Lioulemes, VARM: Using Virtual Reality to Program Robotic Manipulators, International Conference on PErvasive Technologies Related to Assistive Environments, PETRA, Corfu Island Greece, July 2017.
4. Konstantinos Tsiakas, Maher Abujelala, Alexandros Lioulemes, Fillia Makedon, An Intelligent Interactive Learning and Adaptation Framework for Robot-Based Vocational Training, in IEEE Symposium Series in Computational Intelligence, SSCI 2016.
5. Alexandros Lioulemes, Michail Theofanidis, Fillia Makedon, Quantitative Analysis of the Human Upper-Limb Kinematic Model for Robot-based Rehabilitation Applications, IEEE Conference on Automation Science and Engineering (CASE), Fort Worth TX, August 2016.
6. Alexandros Lioulemes, Adaptive User and Haptic Interfaces for Smart Assessment and Training, Doctoral Consortium, ACM Conference on Intelligent User Interfaces, Sonoma CA, March 2016 .



7. Benjamin Chebaa, Alexandros Lioulemes, Maher Abujelala, Dylan Ebert, Eric Becker, Fillia Makedon, Multimodal Analysis of Serious Game for Cognitive and Physiological Assessment, The 1st Workshop on Human Behaviour Monitoring, Interpretation and Understanding, NOTION, International Conference on PErvasive Technologies Related to Assistive Environments, PETRA, Corfu Island Greece, July 2016.
8. Alexandros Lioulemes, Michalis Papakostas, Shawn N. Gieser, Theodora Toutountzi, Maher Abujelala, Sanika Gupta, Christopher Collander, Christopher D. McMurrrough, Fillia Makedon, A Survey of Sensing Modalities for Human Activity, Behavior, and Physiological Monitoring , International Conference on PErvasive Technologies Related to Assistive Environments, PETRA, Corfu Island Greece, July 2016.
9. Michail Theofanidis, Alexandros Lioulemes, Fillia Makedon, A Motion and Force Analysis System for Human Upper-limb Exercises , International Conference on PErvasive Technologies Related to Assistive Environments, PETRA, Corfu Island Greece, July 2016.
10. Srujana Gattupalli, Alexandros Lioulemes, Shawn N. Gieser, Paul Sassaman, Vasilis Athitsos, Fillia Makedon, MAGNI: A Real-time Robot-assisted Game-based Tele-Rehabilitation System, International Conference on Human-Computer Interaction, Toronto, August 2016.
11. Alexandros Lioulemes, Paul Sassaman, Shawn N. Gieser, Vangelis Karkaletsis, Fillia Makedon, Vangelis Metsis, Self-Managed Patient-Game Interaction Using the Barrett WAM Arm for Motion Analysis , International Conference on PErvasive Technologies Related to Assistive Environments, PETRA, Corfu Island Greece, July 2015.

12. Alexandros Lioulemes, Nikos Sarafianos, Theodoros Giannakopoulos, Vangelis Karkaletsis, A Two-Step Identification Method for Human-Robot Interaction in Assistive Environments , The 5th Workshop on Robotics in Assistive Environments, International Conference on PErvasive Technologies Related to Assistive Environments, PETRA, Corfu Island Greece, July 2015.
13. Alexandros Lioulemes, Robot-Assisted Rehabilitation for Smart Assessment and Training, Healthcare Informatics (ICHI), 2015 International Conference on, 469-469.
14. Maher Abujelala, Alexandros Lioulemes, Paul Sassaman, Fillia Makedon, Robot-aided Rehabilitation using Force Analysis, International Conference on PErvasive Technologies Related to Assistive Environments, PETRA, Corfu Island Greece, July 2015.
15. Christopher D. McMurrough, Alexandros Lioulemes, Scott Phan, Fillia Makedon, 3D Mapping of Visual Attention for Smart Rehabilitation, International Conference on PErvasive Technologies Related to Assistive Environments, PETRA, Corfu Island Greece, July 2015.
16. Alexandros Lioulemes, Using Haptic and Vision sensing for Human-Robot Interaction, Doctoral Consortium, IEEE International Symposium on Haptic Audio-Visual Environments and Games, Dallas TX, October 2014.
17. Scott Phan, Alexandros Lioulemes , Cyril Lutterodt, Fillia Makedon, Vangelis Metsis, Guided Physical Therapy Through the Use of the Barrett WAM Robotic Arm , IEEE International Symposium on Haptic Audio-Visual Environments and Games, Dallas TX, October 2014
18. Alexandros Papangelis, Georgios Galatas, Konstantinos Tsiakas, Alexandros Lioulemes, Dimitrios Zikos, Fillia Makedon, Dialogue System for Ensuring Safe

Rehabilitation , International Conference on Human-Computer Interaction, Crete Greece, (6) 2014: 349-358.

19. Alexandros Lioulemes, Georgios Galatas, Vangelis Metsis, Gian luca Mariotini, Fillia Makedon, Safety Challenges in using AR.Drone to collaborate with humans in indoor environments , The 4th Workshop on Robotics in Assistive Environments, International Conference on PErvasive Technologies Related to Assistive Environments, PETRA, Rhodes Island Greece, May 2014.

## REFERENCES

- [1] Patent issued for combined cognitive and physical therapy.” computer weekly news. newsrx. retrieved january 11, 2016 from highbeam research:, 2015. [Available at: <https://www.highbeam.com/doc/1G1-416447509.html>].
- [2] Maher Abujelala, Alexandros Lioulemes, Paul Sassaman, and Fillia Makedon. Robot-aided rehabilitation using force analysis. In *Proceedings of the 8th ACM International Conference on Pervasive Technologies Related to Assistive Environments*, PETRA '15, pages 97:1–97:2, New York, NY, USA, 2015. ACM.
- [3] Francisco Javier Badesa, Ricardo Morales, Nicolas M Garcia-Aracil, Jose M Sabater, Loredana Zollo, Eugenia Papaleo, and Eugenio Guglielmelli. Dynamic adaptive system for robot-assisted motion rehabilitation. *IEEE Systems Journal*, 10(3):984–991, 2016.
- [4] Sivakumar Balasubramanian, Roberto Colombo, Irma Sterpi, Vittorio Sanguineti, and Etienne Burdet. Robotic assessment of upper limb motor function after stroke. *American Journal of Physical Medicine & Rehabilitation*, 91(11):S255–S269, 2012.
- [5] Laurent Ballaz, Maxime Raison, Christine Detrembleur, Guillaume Gaudet, and Martin Lemay. Joint torque variability and repeatability during cyclic flexion-extension of the elbow. *BMC sports science, medicine and rehabilitation*, 8(1):8, 2016.
- [6] Sai K Banala, Suni K Agrawal, and John P Scholz. Active leg exoskeleton (alex) for gait rehabilitation of motor-impaired patients. In *Rehabilitation Robotics*,

2007. *ICORR 2007. IEEE 10th International Conference on*, pages 401–407. IEEE, 2007.
- [7] Peter N. Belhumeur, João P Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):711–720, 1997.
- [8] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [9] G. Bradski. *Dr. Dobb’s Journal of Software Tools*, 2000.
- [10] Pierre-Jean Bristeau, François Callou, David Vissière, Nicolas Petit, et al. The navigation and control technology inside the ar. drone micro uav. In *18th IFAC World Congress*, pages 1477–1484, 2011.
- [11] J Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.
- [12] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [13] Benjamin Chebaa, Alexandros Lioulemes, Maher Abujelala, Dylan Ebert, Scott Phan, Eric Becker, and Fillia Makedon. Multimodal analysis of serious games for cognitive and physiological assessment. In *Proceedings of the 9th ACM International Conference on Pervasive Technologies Related to Assistive Environments, PETRA 2016, Corfu Island, Greece, June 29 - July 1, 2016*, page 30, 2016.
- [14] Chris Collander, Joseph Tompkins, Alexandros Lioulemes, Michail Theofanidis, Ali Sharifiara, and Fillia Makedon. Varm: Using virtual reality to program robotic manipulators. In *International Conference on Pervasive Technologies Related to Assistive Environments, (PETRA), Rhodes Island Greece, June 2017*.

- [15] John J Craig. *Introduction to robotics: mechanics and control*, volume 3. Pearson Prentice Hall Upper Saddle River, 2005.
- [16] Scott C Cuthbert and George J Goodheart. On the reliability and validity of manual muscle testing: a literature review. *Chiropractic & osteopathy*, 15(1):4, 2007.
- [17] Scott C Cuthbert and George J Goodheart. On the reliability and validity of manual muscle testing: a literature review. *Chiropractic & osteopathy*, 15(1):4, 2007.
- [18] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.
- [19] C De Boer, J Van der Steen, RJ Schol, and JJM Pel. Repeatability of the timing of eye–hand coordinated movements across different cognitive tasks. *Journal of neuroscience methods*, 218(1):131–138, 2013.
- [20] Alessandro De Luca and Lorenzo Ferrajoli. A modified newton-euler method for dynamic computations in robot fault detection and control. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3359–3364. IEEE, 2009.
- [21] Rudolfs Drillis, Renato Contini, and Maurice Bluestein. Body segment parameters. *Artificial limbs*, 8(1):44–66, 1964.
- [22] Matthew Dyck and Mahdi Tavakoli. Measuring the dynamic impedance of the human arm without a force sensor. In *Rehabilitation Robotics (ICORR), 2013 IEEE International Conference on*, pages 1–8. IEEE, 2013.
- [23] Coralie English, Susan Hillier, Gurpreet Kaur, and Laura Hundertmark. People with stroke spend more time in active task practice, but similar time in walking practice, when physiotherapy rehabilitation is provided in circuit classes

- compared to individual therapy sessions: an observational study. *Journal of physiotherapy*, 60(1):50–54, 2014.
- [24] Mustafa Suphi Erden and Aude Billard. End-point impedance measurements at human hand during interactive manual welding with robot. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 126–133, 2014.
- [25] Jerry Alan Fails and Dan R Olsen Jr. Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 39–45. ACM, 2003.
- [26] Lorenzo Ferrajoli and Alessandro De Luca. A modified newton-euler method for dynamic computations in robot fault detection and control. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3359–3364, 2009.
- [27] David Feygin, Madeleine Keehner, and R Tendick. Haptic guidance: Experimental evaluation of a haptic training method for a perceptual motor skill. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002. HAPTICS 2002. Proceedings. 10th Symposium on*, pages 40–47. IEEE, 2002.
- [28] Amanda C Fuzaro, Carlos T Guerreiro, Fernanda C Galetti, Renata BVM Jucá, and João E de Araujo. Modified constraint-induced movement therapy and modified forced-use therapy for stroke patients are both effective to promote balance and gait improvements. *Brazilian Journal of Physical Therapy*, 16(2):157–165, 2012.
- [29] Srujana Gattupalli, Alexandros Lioulemes, Shawn N Gieser, Paul Sassaman, Vasilis Athitsos, and Fillia Makedon. Magni: A real-time robot-aided game-based tele-rehabilitation system. In *International Conference on Universal Access in Human-Computer Interaction*, pages 344–354. Springer, 2016.

- [30] B Ghannadi, N Mehrabi, and J Mcphee. Development of a human-robot dynamic model to support model-based control design of an upper limb rehabilitation robot. pages 4–5, 2015.
- [31] Simon Hermann, Reinhard Klette, and Reinhard Klette. Multigrid analysis of curvature estimators. Technical report, Massey University, 2003.
- [32] Friedhelm Hummel, Pablo Celnik, Pascal Giraux, Agnes Floel, Wan-Hsun Wu, Christian Gerloff, and Leonardo G Cohen. Effects of non-invasive cortical stimulation on skilled motor function in chronic stroke. *Brain*, 128(3):490–499, 2005.
- [33] Rajibul Huq, Rosalie Wang, Elaine Lu, Debbie Hebert, Herve Lacheray, and Alex Mihailidis. Development of a fuzzy logic based intelligent system for autonomous guidance of post-stroke rehabilitation exercise. In *Rehabilitation Robotics (ICORR), 2013 IEEE International Conference on*, pages 1–8. IEEE, 2013.
- [34] J-SR Jang. Anfis: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, 23(3):665–685, 1993.
- [35] Jørgen Jepsen, Lise Laursen, Anders Larsen, and Carl-Göran Hagert. Manual strength testing in 14 upper limb muscles a study of inter-rater reliability. *Acta Orthopaedica Scandinavica*, 75(4):442–448, 2004.
- [36] Roland S Johansson, Göran Westling, Anders Bäckström, and J Randall Flanagan. Eye–hand coordination in object manipulation. *the Journal of Neuroscience*, 21(17):6917–6932, 2001.
- [37] Margaret Kelly-Hayes, Alexa Beiser, Carlos S Kase, Amy Scaramucci, Ralph B DAgostino, and Philip A Wolf. The influence of gender and age on disability following ischemic stroke: the framingham study. *Journal of Stroke and Cerebrovascular Diseases*, 12(3):119–126, 2003.



- [38] Julia León, Aurelio Ureña, Manuel Jorge Bolaños, Alfonso Bilbao, and Antonio Oña. A combination of physical and cognitive exercise improves reaction time in persons 61-84 years old. *Journal of aging and physical activity*, 23(1):72–77, 2015.
- [39] Chong Li, Zoltán Rusák, Imre Horváth, Linhong Ji, and Yuemin Hou. Current status of robotic stroke rehabilitation and opportunities for a cyber-physically assisted upper limb stroke rehabilitation. TMCE, 2014.
- [40] Chong Li, Zoltán Rusák, Imre Horváth, Linhong Ji, and Yuemin Hou. Current status of robotic stroke rehabilitation and opportunities for a cyber-physically assisted upper limb stroke rehabilitation. In *Proceedings of the 10th international symposium on tools and methods of competitive engineering TMCE 2014, Budapest, Hungary, May 19-23, 2014*. TMCE, 2014.
- [41] Alexandros Lioulemes, Georgios Galatas, Vangelis Metsis, Gian Luca Mariotini, and Fillia Makedon. Safety challenges in using ar.drone to collaborate with humans in indoor environments. In *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments, PETRA 2014, Island of Rhodes, Greece, May 27 - 30, 2014*, pages 33:1–33:4, 2014.
- [42] Alexandros Lioulemes, Nikolaos Sarafianos, Theodoros Giannakopoulos, and Vangelis Karkaletsis. A two-step identification method for human-robot interaction in assistive environments. In *Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments, PETRA 2015, Corfu, Greece, July 1-3, 2015*, pages 71:1–71:4, 2015.
- [43] Alexandros Lioulemes, Paul Sassaman, Shawn N. Gieser, Vangelis Karkaletsis, Fillia Makedon, and Vangelis Metsis. Self-managed patient-game interaction using the barrett wam arm for motion analysis. In *Proceedings of the 8th ACM*

*International Conference on Pervasive Technologies Related to Assistive Environments*, PETRA '15, pages 34:1–34:8, New York, NY, USA, 2015. ACM.

- [44] J Liu, JL Emken, SC Cramer, and DJ Reinkensmeyer. Learning to perform a novel movement pattern using haptic guidance: slow learning, rapid forgetting, and attractor paths. In *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on*, pages 37–40. IEEE, 2005.
- [45] Donald Lloyd-Jones, Robert Adams, Mercedes Carnethon, Giovanni De Simone, T Bruce Ferguson, Katherine Flegal, Earl Ford, Karen Furie, Alan Go, Kurt Greenlund, et al. Heart disease and stroke statistics 2009 update. *Circulation*, 119(3):e21–e181, 2009.
- [46] Wendy J Loken, Allen E Thornton, Randall L Otto, and Charles J Long. Sustained attention after severe closed head injury. *Neuropsychology*, 9(4):592, 1995.
- [47] Rui CV Loureiro, William S Harwin, Kiyoshi Nagai, and Michelle Johnson. Advances in upper limb stroke rehabilitation: a technology push. *Medical & biological engineering & computing*, 49(10):1103, 2011.
- [48] Peter Lum, David Reinkensmeyer, Richard Mahoney, William Z Rymer, and Charles Burgar. Robotic devices for movement therapy after stroke: current status and challenges to clinical acceptance. *Topics in stroke rehabilitation*, 8(4):40–53, 2002.
- [49] Emanuele Magrini, Fabrizio Flacco, and Alessandro De Luca. Control of generalized contact motion and force in physical human-robot interaction. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2298–2304. IEEE, 2015.
- [50] Regan L Mandryk and M Stella Atkins. A fuzzy physiological approach for continuously modeling emotion during interaction with play technologies. *International journal of human-computer studies*, 65(4):329–347, 2007.

- [51] Sebastiaan Mathôt, Jean-Baptiste Melmi, Lotje Van Der Linden, and Stefan Van der Stigchel. The mind-writing pupil: A human-computer interface based on decoding of covert attention through pupillometry. *PloS one*, 11(2):e0148805, 2016.
- [52] Sebastiaan Mathôt, Elle Van Heusden, and Stefan Van der Stigchel. Attending and inhibiting stimuli that match the contents of visual working memory: Evidence from eye movements and pupillometry. Technical report, PeerJ PrePrints, 2015.
- [53] Mark Matthews, Geri Gay, and Gavin Doherty. Taking part: role-play in the design of therapeutic systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 643–652. ACM, 2014.
- [54] Christopher McMurrough, Alexandros Lioulemes, Scott Phan, and Fillia Makedon. 3d mapping of visual attention for smart rehabilitation. In *Proceedings of the 8th ACM International Conference on Pervasive Technologies Related to Assistive Environments, PETRA 2015, Corfu, Greece, July 1-3, 2015*, pages 95:1–95:2, 2015.
- [55] Elaine L Miller, Laura Murray, Lorie Richards, Richard D Zorowitz, Tamilyn Bakas, Patricia Clark, Sandra A Billinger, et al. Comprehensive overview of nursing and interdisciplinary rehabilitation care of the stroke patient. *Stroke*, 41(10):2402–2448, 2010.
- [56] Kevin Murphy. Hidden markov model HMM toolbox for Matlab. *MIT License*, 1998. Software available at <http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html>.
- [57] L Omelina, Bart Jansen, Bruno Bonnechere, Serge Van Sint Jan, and Jan Cornelis. Serious games for physical rehabilitation: designing highly configurable

- and adaptable games. In *Proceedings of the 9th International Conference on Disability, Virtual Reality & Associated Technologies*, pages 195–201, 2012.
- [58] Rieko Osu and Hiroaki Gomi. Multijoint muscle regulation mechanisms examined by measured human arm stiffness and emg signals. *Journal of neurophysiology*, 81(4):1458–1468, 1999.
- [59] S. Parasuraman, Chew Yee Kee, and Arif Oyong. Human upper limb and arm kinematics for robot based rehabilitation. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, pages 845–850, 2009.
- [60] Subramaniam Parasuraman, Arif Wicaksono Oyong, and Velappa Ganapathy. Development of robot assisted stroke rehabilitation system of human upper limb. In *2009 IEEE International Conference on Automation Science and Engineering*, pages 256–261. IEEE, 2009.
- [61] Scott Phan, Alexandros Lioulemes, Cyril Lutterrodt, Fillia Makedon, and Vangelis Metsis. Guided physical therapy through the use of the barrett wam robotic arm. In *Haptic, Audio and Visual Environments and Games (HAVE), 2014 IEEE International Symposium on*, pages 24–28. IEEE, 2014.
- [62] Scott Phan, Alexandros Lioulemes, Cyril Lutterrodt, Fillia Makedon, and Vangelis Metsis. Guided physical therapy through the use of the barrett wam robotic arm. In *Haptic, Audio and Visual Environments and Games (HAVE), 2014 IEEE International Symposium on*, pages 24–28. IEEE, 2014.
- [63] PrimeSense. Primesense. [Available at: <https://en.wikipedia.org/wiki/PrimeSense>.
- [64] Michael J Quinn. *Ethics for the information age*. Pearson, 2014.
- [65] Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *Robotics and automation (ICRA), 2011 IEEE International Conference on*, pages 1–4. IEEE, 2011.

- [66] Elham Saraee and Margrit Betke. Dynamic adjustment of physical exercises based on performance using the proficio robotic arm. In *Proceedings of the 9th ACM International Conference on Pervasive Technologies Related to Assistive Environments*, page 8. ACM, 2016.
- [67] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.
- [68] Jonathan M Silver, Thomas W McAllister, and David B Arciniegas. Depression and cognitive complaints following mild traumatic brain injury. *American Journal of Psychiatry*, 2009.
- [69] Marcy L Freed MA SLP, Robert L Chatburn RRT FAARC, and Michael Christian. Electrical stimulation for swallowing disorders caused by stroke. *Respir care*, 46(5):466–474, 2001.
- [70] Meg Stanger and Susan Oresic. Rehabilitation approaches for children with cerebral palsy: overview. *Journal of child neurology*, 18(1 suppl):S79–S88, 2003.
- [71] Jennifer A Stevens and Mary Ellen Phillips Stoykov. Using motor imagery in the rehabilitation of hemiparesis. *Archives of physical medicine and rehabilitation*, 84(7):1090–1092, 2003.
- [72] Vicon Motion Capture System. Vicon motion capture system. [Available at: <http://www.vicon.com/>].
- [73] Unity Technology. Cross-platform game creation system, 2015. [Available at: <http://unity3d.com/>; accessed 19-February-2015].
- [74] Daniel Thalmann. Interactive Human Motion Control Using a Closed- form of Direct and Inverse Dynamics.

- [75] Michail Theofanidis, Saif Iftekar Sayed, Alexandros Lioulemes, and Fillia Makedon. An interactive robot-based vocational assessment game using lego assembly. In *The 1st Workshop on Assistive Robots , International Conference on Pervasive Technologies Related to Assistive Environments, (PETRA), Rhodes Island Greece, June 2017*.
- [76] Carolien J. van Anandel, Nienke Wolterbeek, Caroline A M Doorenbosch, DirkJan (H E J ) Veeger, and Jaap Harlaar. Complete 3D kinematics of upper extremity functional tasks. *Gait and Posture*, 27(1):120–127, 2008.
- [77] Qifei Wang, Gregorij Kurillo, Ferda Ofli, and Ruzena Bajcsy. Evaluation of pose tracking accuracy in the first and second generations of microsoft kinect. In *Healthcare Informatics (ICHI), 2015 International Conference on*, pages 380–389. IEEE, 2015.
- [78] Charles P Warlow, Jan Van Gijn, Martin S Dennis, Joanna M Wardlaw, John M Bamford, Graeme J Hankey, Peter AG Sandercock, Gabriel Rinkel, Peter Langhorne, Cathie Sudlow, et al. *Stroke: practical management*. John Wiley & Sons, 2011.
- [79] Helena Westerberg, H Jacobaeus, Tatja Hirvikoski, Peder Clevberger, M-L Östensson, Aniko Bartfai, and Torkel Klingberg. Computerized working memory training after stroke—a pilot study. *Brain Injury*, 21(1):21–29, 2007.
- [80] Paul Woodcraft and Despina Learmonth. Computer-aided cognitive behaviour therapy: Paul woodcraft and colleagues describe how a computer-aided intervention, beating the blues, was successfully introduced in a cognitive behaviour therapy service as an alternative to face-to-face contact. *Mental Health Practice*, 13(3):16–20, 2009.

- [81] Wenyi Zhao, Rama Chellappa, P. Jonathon Phillips, and Azriel Rosenfeld. Face recognition: A literature survey. *ACM computing surveys (CSUR)*, 35(4):399–458, 2003.
- [82] Huiyu Zhou and Huosheng Hu. Human motion tracking for rehabilitation-A survey. *Biomedical Signal Processing and Control*, 3(1):1–18, 2008.

## BIOGRAPHICAL STATEMENT

Alexandros Lioulemes was born in Gjirokaster of Albania in 1989. He moved to Greece in 1998 and stayed three years at Rizarios boarding school, in Monodendri Zagorochoreia. In 2000 he continued his studies in Ioannina and he was admitted to the Computer Science Department at the University of Ioannina in 2007. He received his B.Sc. and M.Sc. degrees in 2011 and 2013, at the same institution. During his doctoral studies, he was a research fellow at the Institute of Informatics and Telecommunications of the National Center for Scientific Research Demokritos. Finally, he received his Ph.D. in Computer Engineering in 2017 from the Computer Science and Engineering Department at the University of Texas at Arlington. His research interests include Computer and Robotic Vision, Artificial Intelligence, and Robotic Haptics for medical applications. Last but not least, he has co-authored several peer reviewed papers published in technical conferences and journals.