FACULTY OF ENGINEERING OF THE UNIVERSITY OF PORTO

Development of an intelligent robotic learning system for physiotherapy treatments

Inês Pascoal Nogueira

Master in Biomedical Engineering

Supervisors: Prof. Hélder Filipe Pinto de Oliveira, PhD,
Cláudia Daniela Costa Rocha, MSc,
Prof. Manuel Fernando Santos Silva, PhD

October 15, 2024

Development of an intelligent robotic learning system for physiotherapy treatments

Inês Pascoal Nogueira

Master in Biomedical Engineering

October 15, 2024

Abstract

Stroke is one of the leading causes of death and disability worldwide, with around 80% of disabilities occurring in the upper limbs. Physiotherapy is essential for rehabilitating these limbs; however, patients often struggle to consistently perform the necessary movements. As a result, the support and supervision of a physiotherapist are crucial, despite the growing challenges in ensuring consistent availability and capacity for such guidance. For this reason, the aim of this work is to develop an intelligent robotic system that assists in physiotherapy, ensuring that the movements are appropriate for the patient and always performed correctly. The system uses computer vision to detect the upper limbs in order to assess the patient's ability to move, and the robotic system adapts accordingly. To the best of the author's knowledge, the current literature does not have any work on systems that assess the patient's ability to move and then adjusts the exercises accordingly. Thus, the developed system addresses this gap. Following the analysis of upper limb mobility in the patient's abduction/adduction movements using the Kinect SDK, the UR5 robotic manipulator is employed to perform movements with the patient's arm passively coupled. The results obtained demonstrated that the system is easily adaptable and applicable in therapy, as motion detection was achieved with satisfactory results, with errors below 5%. Except in some cases where movement repetition may be necessary, in which a maximum absolute deviation of 8 cm was observed, the Kinect SDK is sufficiently accurate in capturing the coordinates of keypoints, enabling effective therapy sessions. The system integration was successful, enabling automation and autonomy after installation. However, maintaining a consistent setup may be challenging in rehabilitation environments. To facilitate user adaptation, a user-friendly and intuitive interface was developed, specifically designed for therapists to use. As main points for future work, it would be interesting to train a generic pose detection algorithm using a dataset specific to rehabilitation environments. Additionally, using a manipulator with a longer arm would be important to determine if the extrapolation made is indeed feasible and yields good results.

Keywords: Rehabilitation, upper limbs, keypoints detection, robotic manipulator

Resumo

O acidente vascular cerebral (AVC) é uma das maiores causas de morte e incapacidade em todo o mundo, sendo que cerca de 80% das incapacidades ocorrem em membros superiores, incluindo paralesia, perda de sensibilidade e dor. A fisioterapia desempenha um papel crucial na reabilitação de membros afetados, mas os pacientes têm frequentemente dificuldades em manter os movimentos necessários. Assim, o suporte e a supervisão do fisioterapeuta são essenciais, contudo, há uma crescente escassez de capacidade e disponibilidade para oferecer esse acompanhamento. Por este motivo, o objetivo deste trabalho é desenvolver um sistema robótico inteligente que assista na fisioterapia, procurando garantir que o movimento é adequado ao paciente e que é sempre feito corretamente. O sistema utiliza visão por computador para fazer deteção dos membros para se perceber a capacidade de movimentação do paciente e é adaptado o movimento realizado no sistema robótico. De acordo com o conhecimento da autora, não existe na literatura atual um trabalho sobre um sistema que avalie a capacidade de movimentação do paciente e de seguida adapte os exercícios ao mesmo. Assim, o sistema desenvolvido vem culmatar esta lacuna. Após a análise à mobilidade dos membros superiores do paciente no movimento de abdução/adução através dos SDK da Kinect, o manipulador robótico UR5 é empregado para realizar movimentos com o braço do paciente acoplado de forma passiva. Os resultados obtidos demonstraram que o sistema é facilmente adaptável e aplicável na terapia, uma vez que a deteção do movimento foi realizada com resultados satisfatórios, com erros inferiores a 5%. Excepto em alguns casos em que pode ser necessária a repetição do movimento, tendo-se obtendo desvios absolutos máximos de 8 cm, o SDK do Kinect mostra-se suficientemente preciso para captar as coordenadas dos keypoints, permitindo a realização eficaz das sessões de terapia. A integração do sistema foi bem sucedida, permitindo a automação e autonomia após a instalação. Contudo, é necessário que a configuração permaneça constante, o que pode ser um desafio em ambientes de reabilitação. Para facilitar a adaptação dos usuários, foi desenvolvida uma interface fácil de utilizar e intuitiva, projetada especificamente para seja utilizada pelos terapeutas. Como principais pontos para trabalhos futuros, seria interessante treinar um algoritmo genérico de detecção de pose usando um *dataset* específico para ambientes de reabilitação. Além disso, utilizar um manipulador com um braço de maior extensão seria importante para verificar se a extrapolação feita é, de fato, possível de implementar com bons resultados.

Keywords: Reabilitação, membros superiores, deteção de *keypoints*, manipulador robótico

Acknowledgements

Firstly, I would like to express my gratitude to my supervisors, Professor Hélder Oliveira, Cláudia Rocha and Professor Manuel Silva for their constant availability and indispensable guidance throughout this journey. Their knowledge and dedication played a crucial role in defining the final outcome of this work.

To the iiLab collaborators who helped me at numerous occasions, always with kindness and made me feel integrated. And to Ana Filipa Rodrigues who was available on several occasions to guide me through the work, I extend my sincere thank you.

On a more personal note, I would like to express my heartfelt gratitude to my family for their unwavering support and patience throughout this journey, especially my parents, sister and brother, whose belief in me has been a constant source of motivation. Their encouragement and guidance have been invaluable, and I am deeply thankful for all they continue to teach me.

I would also like to extend my sincere thanks to my friends, both from childhood and throughout my academic journey in physical and biomedical engineering, for their role in helping me remain grounded and focused. Your support and companionship were exceptional.

In conclusion, I extend my deep appreciation to everyone who has been a part of my journey, not only in the context of this dissertation but throughout my entire academic journey. Each person mentioned here has contributed to the successful completion of this thesis, and for that, I am truly thankful.

This work is co-financed by Component 5 - Capitalization and Business Innovation of core funding for Technology and Innovation Centres (CTI), integrated in the Resilience Dimension of the Recovery and Resilience Plan within the scope of the Recovery and Resilience Mechanism (MRR) of the European Union (EU), framed in the Next Generation EU, for the period 2021 - 2026.

"In the middle of difficulty lies opportunity."

Albert Einstein

Contents

1 Introduction **1**
- 1.1 Motivation . . . 2
- 1.2 Objectives . . . 2
- 1.3 Dissertation structure . . . 3

2 Theoretical Foundation **5**
- 2.1 Anatomical concepts . . . 5
  - 2.1.1 Anatomical plans . . . 5
  - 2.1.2 Upper limbs anatomy . . . 6
  - 2.1.3 Upper limbs movements . . . 7
- 2.2 Stroke impact analysis . . . 8
- 2.3 Detection of limbs movement . . . 9
- 2.4 Computer vision for limbs detection . . . 11
- 2.5 Robotic systems for rehabilitation . . . 12
- 2.6 Summary . . . 14

3 Literature Review **15**
- 3.1 Pose detection: approaches and technologies . . . 15
  - 3.1.1 Key findings . . . 21
- 3.2 Robotic systems . . . 21
  - 3.2.1 Key findings . . . 24
- 3.3 Computer vision in robotics aided rehabilitation . . . 24
  - 3.3.1 Key findings . . . 27
- 3.4 Summary . . . 27

4 From Concept to Execution **29**
- 4.1 Description of the proposed solution . . . 29
- 4.2 System architecture . . . 30
- 4.3 Detection of keypoints . . . 32
- 4.4 UR5 manipulator . . . 32
- 4.5 Integration . . . 34
- 4.6 Extrapolation of the project . . . 36

5 From Execution to Results **39**
- 5.1 Detection of keypoints . . . 39
  - 5.1.1 Quality analysis . . . 40
  - 5.1.2 Particularity of the individual 3 . . . 44
  - 5.1.3 Qualitative tests . . . 46

5.2 UR5 manipulator 48
5.3 Handle for UR5 49
5.4 Integration 51
5.4.1 Variables to define the trajectory 52
5.4.2 UR5 response 56
5.4.3 Graphical user interface 57
5.4.4 Integrated system 60
5.5 SWOT analysis 62

6 Conclusion and Future Work **65**
6.1 Conclusions 65
6.2 Future work 66

References **69**

List of Figures

2.1 Anatomical planes of the human body. . . . 6
2.2 Anatomy of upper limb. . . . 6
2.3 Basic movements of upper limb: (a) shoulder flexion and extension, shoulder adduction and abduction and shoulder rotation, respectively; (b) elbow flexion and extension; (c) elbow supination and pronation; (d) wrist flexion/extension and radial/ulnar deviation. . . . 7
2.4 Illustrative examples of two types of orthosis according to their structure: (a) Manipulator system; (b) Exoskeleton system. . . . 13

3.1 OpenPose algorithm. . . . 16
3.2 AlphaPose algorithm architecture. . . . 16
3.3 Cascaded Pyramid Network process. . . . 17
3.4 DeepPose algorithm. . . . 17
3.5 BlazePose algorithm: (a) Network architecture; (b) Keypoints that will be detected and its distribution; (c) Virtual keypoints represented. . . . 18
3.6 DensePose algorithm. . . . 18
3.7 Kinect system: (a) Elements in Kinect camera; (b) Triangulation of points; (c) detection process. . . . 19
3.8 Examples of robotic systems for rehabilitation purposes: (a) MIME; (b) GAVRe^2; (c) ANYexo; (d) Robert; (e) InMotion; (f) MOTORE; (g) ReoGO; (h) ArmeoPower; (i) ArmeoSpring. . . . 22
3.9 Workflow of the telerehabilitation system using the humanoid NAO. . . . 25
3.10 Overview of the system design and implementation: (a) RaspberryPi and MediaPipe based system flow chart; (b) RaspberryPi and MediaPipe based system. . . . 25
3.11 Scheme of the telehealth service. . . . 26
3.12 Overview of the Burt system design and implementation: (a) Burt system chart flow; (b) Burt system. . . . 26

4.1 System architecture block diagram. . . . 31
4.2 Referential of Kinect v2. . . . 32
4.3 Graphical representation of the increment normalised, i'. . . . 35
4.4 Attempt arm flexion/extension movement with UR5. . . . 36

5.1 Representation of upper joints. . . . 40
5.2 Camera positioning: (a) Illustrated positioning of camera; (b) Real setup made. . . . 40
5.3 Arm positioning: (a) Parallel to the camera; (b) Towards the camera; (c) Backwards the camera. . . . 41

5.4 Representation of the detection result when the movement is done in different directions: (a) Parallel to the camera; (b) Towards to the camera; (c) Opposite to the camera. . . . 41
5.5 Correct real measurement of distances between keypoints. . . . 42
5.6 Distances between the wrist and the shoulder of individual 1: (a) Left arm; (b) Right arm. . . . 44
5.7 Distances between the wrist and the shoulder of individual 2: (a) Left arm; (b) Right arm. . . . 44
5.8 Distances between the wrist and the shoulder of individual 3: (a) Left arm. (b) Right arm . . . 45
5.9 2D keypoints of an exercise done by the individual 3, with the wrist, elbow, and shoulder (both right and left) highlighted in red. . . . 45
5.10 Distances of right side of person 3: (a) Distance shoulder-elbow; (b) Distance elbow-wrist. . . . 46
5.11 Impact of background on detection: (a) White wall; (b) Darker environment; (c) Mixed background. . . . 47
5.12 Impact of additional individuals in the camera's field of view: (a) Person positioned behind the detected individual; (b) Person positioned in front of the detected individual. . . . 47
5.13 Effect of obstruction on detection: (a) Small object interference with minimal impact; (b) Full occlusion of an arm leading to decreased accuracy in detection. . 47
5.14 Spatial positioning of UR5 relative to the bench. . . . 48
5.15 Drawing of the handle created in *SolidWorks*, with units in millimetres. . . . 49
5.16 3D printed handle attached to the UR5. . . . 50
5.17 Positions for holding the handle: (a) Wrong position with the wrist flexed; (b) Wrong position with the wrist tucked in; (c) Right position, with the wrist straight. . . . 50
5.18 Setup representation of the position between the UR5, the Kinect camera and patient. 52
5.19 Position in relation to the robot: (a) When the right side is being analysed; (b) When the left side is being analysed. . . . 52
5.20 Illustration of coordinate system adjustments: (a) Modification of Kinect axes name to simplify the transformation matrix, $\mathbf{H}_{\mathbf{K}}^{\mathbf{M}}$; (b) Right hand rule for axis orientation. . . . 53
5.21 Spatial positioning of UR5 relative to the Kinect. . . . 53
5.22 Initial angle position for the left side: (a) Initial position if the angle considered was $\frac{3\pi}{2}$ *rad*, equivalent to zero degrees from the body; (b) Initial position if the angle considered was $\left(\frac{3\pi}{2} - 12 \cdot \frac{\pi}{180}\right)$ *rad*, equivalent to twelve degrees from the body. . . . 57
5.23 User interface overview: (a) First interface window; (b) Second interface window, which has the same appearance whatever side is chosen. . . . 58
5.24 User interface overview: (a) Reference exercise interface window; (b) Patient exercise interface window. . . . 58
5.25 Results interface window. . . . 59
5.26 Interface window which displays the variables to define the trajectory. . . . 60
5.27 Connections and protocols used in integrated system. . . . 60
5.28 Information transmission in integrated system. . . . 61
5.29 UML diagram of all the integrated system. . . . 62
5.30 SWOT analysis. . . . 63

List of Tables

2.1 Advantages and disadvantages of the existing detection methods. 11
2.2 Advantages and disadvantages of manipulators and exoskeleton systems. 13
2.3 Different treatment approaches depending on the interaction with the robotic system. 14

3.1 Cameras' characteristics. 20

5.1 Results obtained with Kinect SDK in the plane parallel, towards and opposite to the camera, and the real measurements. The symbol (-) indicates that the test could not be performed. 42
5.2 Comparison between the real distance between keypoints and the distance obtained from Kinect SDK detection, including standard deviation, maximum absolute deviation and error. 43
5.3 Comparison between shoulder coordinates obtained from the UR5 screen and Kinect. 54
5.4 Comparison between the target angle and the angle measured by Kinect. 56

Abbreviations

CNNs	Convolutional Neural Networks
CPN	Cascaded Pyramid Network
DNN	Deep Neural Network
DoF	Degrees of Freedom
EEG	Electroencephalogram
EMG	Electromyography
FMA	Fugl-Meyer Assessment
FPS	Frames Per Second
GAVRe^2	Game Adaptive Virtual Reality Rehabilitation
IMU	Inertial Measurement Unit
IP	Internet Protocol
IR	Infrared
LED	Light-Emitting Diodes
mAP	mean Average Precision
MIME	Mirror Image Movement Enabler
MoCA	Montreal Cognitive Assessment
MOTORE	Mobile robot for upper limb neurOrtho Rehabilitation
MS COCO	Microsoft Common Objects in COntext
NTU	Nanyang Technological University
PAFs	Part Affinity Fields
PETG	Polyethylene Terephthalate Glycol-modified
RGB	Red-Green-Blue
RGB-D	Red-Green-Blue-Depth
RMSE	Root Mean Square Error
RoI	Regions of Interest
ROS	Robot Operating System
RPN	Region Proposal Network
SIS	Stroke Impact Scale
SDK	Software Development Kit
SQL	Structured Query Language
TCP	Transmission Control Protocol
ToF	Time of Flight
UML	Unified Modeling Language
UTD-MHAD	University of Texas at Dallas Multimodal Human Action Dataset
WHO	World Heath Organization
WPF	Windows Presentation Foundation
WSL	Windows Subsystem for Linux
WSO	World Stroke Organization
XAML	eXtensible Application Markup Language

Chapter 1

Introduction

In 2022 the World Stroke Organization (WSO) and the World Heath Organization (WHO) stated that stroke is the third leading cause of disability and deaths combined worldwide, with 1 in 4 people estimated to have a stroke in their lifetime (World Health Organization (2022); World Stroke Organization (2022)). There are many disabilities that stroke can cause and these are summarised below (Kelly-Hayes et al. (1998)).

- Motor impairment - difficulty controlling certain limbs, muscle weakness, loss of coordination and decreased reflexes;
- Sensory impairment - numbness, tingling or altered sensitivity;
- Visual disability - monocular visual loss, homonymous hemianopia or cortical blindness;
- Language dysfunction - reading, writing, comprehension and fluency challenges;
- Cognitive impairment - difficulty with memory, orientation and attention;
- Emotional problems - depression, which is the most common result.

In order to minimise the impact of illnesses or situations such as stroke on people's lives, it is important to rehabilitate and then monitor patients' health, both physically and psychologically. Currently, physiotherapy is considered a primary health care profession and it has a crucial contribution to the rehabilitation of a person after a stroke. It is worth emphasising that it consists not only of helping the patient to regain movement and functionality after an injury, pain or disability, but also in prevention, where mobility and body strengthening exercises should be performed (Cromie et al. (2000)). There are several areas of physiotherapy, three important ones are described below (Shah and Desai (2022)).

1. Musculoskeletal - deals with injuries related to the muscles, bones and joints of the human muscular and skeletal system.
2. Cardio-Respiratory - deals with diseases related to the lung and circulatory system. Patients are usually in hospital recovering.

3. Neurology - related to neurological diseases such as stroke or cerebral palsy. These conditions can lead to partial paralysis of some part of the body, so the exercises are similar to those in the musculoskeletal area. The difference is the source of the problem.

Strokes occur in approximately 25% of people and it is estimated that the incidence of upper limb disability in these people can reach 80%, often with paralysis, loss of sensation and pain in the hand, arm and shoulder (Germanotta et al. (2023); Poltawski et al. (2016); Salvalaggio et al. (2023)).Rehabilitation is based on the fact that the central nervous system is plastic, which means it can adapt to new conditions. After a stroke, there is a reorganisation and with physiotherapy intervention it is possible to relearn lost movements and motor skills (Dobkin and Dorsch (2013); Mang et al. (2013)). Therefore, increasing physiotherapy time, the number of repetitions and its intensity helps to promote plasticity and improve the patient's recovery (Molteni et al. (2018)).

1.1 Motivation

The recovery of motor function is one of the main objectives for the person to have a quality of life comparable with what they had before the event, for which physiotherapy is frequently used (Germanotta et al. (2023); Langhorne et al. (2009)). There are different methods of doing physiotherapy, frequently a series of exercises are done repeatedly to work a specific muscle or body area. Although many patients struggle to maintain the correct movement due to the discomfort felt, Bielecki and Tadi (2023) and Xing et al. (2018) have shown that active therapy, that is, the patient making the effort to do the movement themselves, is more effective than passive therapy, where movement is carried out by the health professional or an external system.

In addition, not only do patients spend little time in physiotherapy, there are studies that show that active time in the therapy period is low when it is unsupervised (Kaur et al. (2012); Connell et al. (2014)). Having constant supervision increases the therapist/patient ratio, reducing the number of patients who can be in rehabilitation at the same time, which means that the total of patients treated is lower than necessary (Dorsch et al. (2019)). Since the demand is growing, once the number of stroke survivors it is also increasing, it is necessary to improve the physical therapy methods (Stinear et al. (2020)). Nowadays, robotics is being widely used to support the rehabilitation of affected limbs in order to optimise and improve patient recovery, allowing users to be more efficient even without the therapist's supervision (Colombo et al. (2005)).

1.2 Objectives

The main purpose of this work is to implement a robotic system for physiotherapy support to help the patients carry out the exercise correctly and improve their performance. Using a computer vision algorithm, the person's upper limb will be detected and tracked. And once the patient has performed the exercise chosen by the healthcare professional to the best of their ability, the therapy will begin with assistance from the robotic system to perform the movement, adjusting its

amplitude according to its initial capabilities. Over time, the exercise difficulty will be incremented in order to progressively evolve until the reference movement is reached.

It is possible to define minor tasks, which have to be achieved in order to reach the final objective and they are the following:

1. Real time upper limb detection and tracking;
2. Definition of the movements to be studied and creation of a baseline, defining a reference point for the patient's performance;
3. Analysis of the patient's movement capacity;
4. Set the robotic manipulator to make the desired movement using a connecting device between the manipulator and the patient's limb;
5. Integration of the complete system, by plugging the connecting device into the patient's limb and making the system move passively the limb to an improved position, taking into account the initial capacity.

1.3 Dissertation structure

This document is divided into four chapters. Chapter 2 refers to upper limb rehabilitation, including anatomy and a set of exercises commonly performed in rehabilitation. This is followed by a study of the various forms of limb detection and rehabilitation methods using robotic systems. Chapter 3 then reviews computer vision algorithms for detecting limbs with a camera and the robotic systems that already exist to support limb recovery. Chapter 4 presents the architecture diagram of the work to be carried out, as well as the tools and methods to be used. In Chapter 5 the results of the analysis and integration made are presented. Finally, Chapter 6 summarises the conclusions of this dissertation and outlines perspectives for future work.

Chapter 2

Theoretical Foundation

Robotics is an area that is being studied extensively and its applications in the field of patient care range from the development of robotic wheelchairs to robotic arms to assist in day-to-day activities and physiotherapy exercises (Blank et al. (2014)). When it comes to limb rehabilitation after a stroke, factors such as the patient's commitment, the correct performance of the movement and the frequency with which they do it, are crucial to improving the patient's mobility. Thus, something that is sought in this branch of robotic systems for limb physiotherapy support is how to involve and motivate the patient in the exercise and how to make the trajectory as controlled and precise as possible (Blank et al. (2014)). However, it should be noted that for this study, the cost, safety and user friendliness of the system are challenges that must always be taken into account (Laut et al. (2016)).

This chapter introduces concepts that will be important for the study that will be carried out, namely relevant movements in physiotherapy, ways of detecting upper limbs, rehabilitation approaches when involving robotic systems and what types of these systems exist.

2.1 Anatomical concepts

This section presents the anatomy of the upper limbs and the movements they allow to make. As there is often motor impairment after a stroke, it is important to know what movements the upper limbs can execute in order to know what needs to be rehabilitated.

2.1.1 Anatomical plans

The human body can be divided into three anatomical planes which are useful to facilitate the study of the anatomical structures with spatial relationship. As it can be seen in the Figure 2.1, the three planes are the transverse plane - divides the superior and inferior parts; the coronal plane - divides the anterior/ventral (the structures closer to the front of the body) and posterior/dorsal (the structures closer to the back) parts; the sagittal plane - divides the left and right parts (Hansen (2019)).

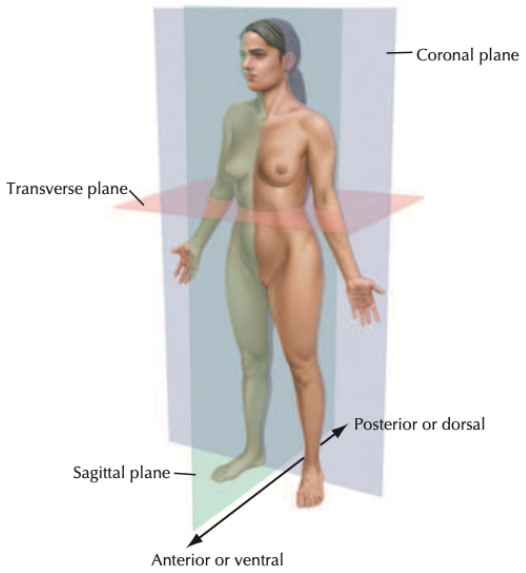


Figure 2.1: Anatomical planes of the human body. From (Hansen (2019)).

2.1.2 Upper limbs anatomy

In Figure 2.2, it is possible to see the anatomy of the upper limbs, which is a complex structure. Generally it can be divided into four parts - the shoulder, upper-arm, forearm and hand.

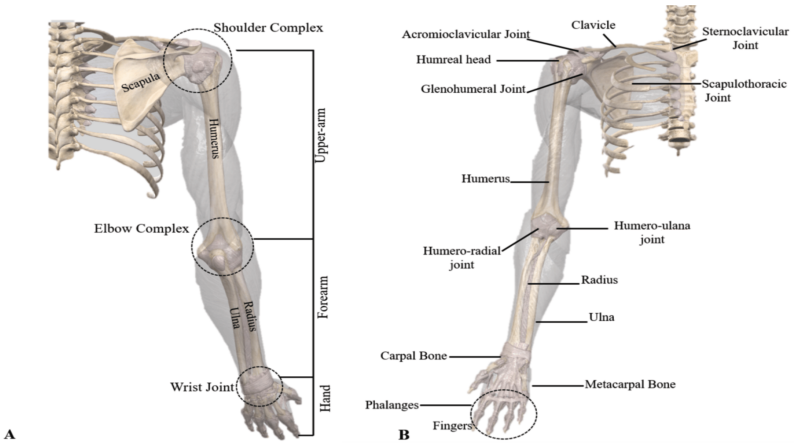


Figure 2.2: Anatomy of upper limb. From (Gull et al. (2020)).

The shoulder is constituted by three bones - humerus, shoulder scapula supported by clavicle. It is the clavicle that prevents the shoulders from moving too far forward. It also consists of four joints: the glenohumeral, acromioclavicular, scapulothoracic and sternoclavicular joint, the last of which is the connection between the shoulder complex and the axial skeleton. For movement to be effective, the shoulder must be stable, yet it must allow a wide range of movement.

The elbow complex is made up of two joints, the humero-ulna and the humero-radial joint, the last one being a ball socket joint between the humerus and the radial bones, allowing rotational

movement. The forearm bones are the ulna on the little finger side and the radius on the thumb side. It is the fact that these two bones articulate proximally to form a pivoting joint, which allows the rotation of the palm up and down without moving the arm. The radius and ulna intersect in this movement. The wrist connects the forearm and the hand and is the joint between the radius and the carpal bones of the hand (Gopura and Kiguchi (2009); Gull et al. (2020); Scanlon and Sanders (2007)).

2.1.3 Upper limbs movements

In respect of movements and degrees of freedom (DoF), the shoulder allows movement with 3 DoF - shoulder flexion/extension, shoulder abduction/adduction and internal/external rotation (Figure 2.3a); the elbow complex provides movement with 2 DoF - flexion/extension (Figure 2.3b) and supination/pronation (Figure 2.3c); the wrist allows motion with 2 DoF - flexion/extension and radial/ulnar deviation (Figure 2.3d). So, in total, the upper limbs have 7 DoF (Gopura and Kiguchi (2009); Gull et al. (2020)).

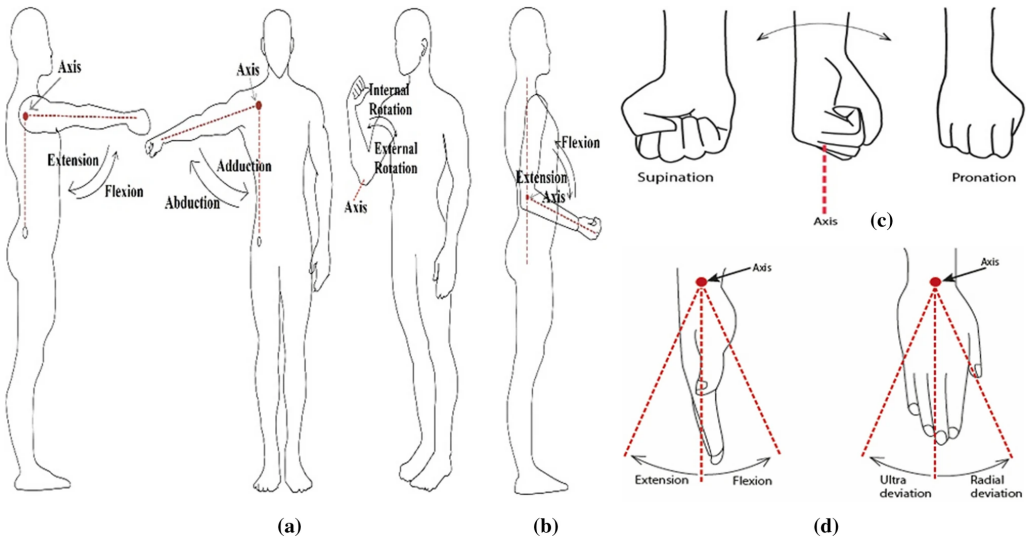


Figure 2.3: Basic movements of upper limb: (a) shoulder flexion and extension, shoulder adduction and abduction and shoulder rotation, respectively; (b) elbow flexion and extension; (c) elbow supination and pronation; (d) wrist flexion/extension and radial/ulnar deviation. From (Narayan et al. (2021)).

There are movements called active movements since they are made only with their own muscles. On the other hand, passive movements are movements where there is a support from an external person/object or another member of the person (Guzzetta et al. (2007); Qassim and Hasan (2020)).

As previously mentioned, it is crucial to do exercises for strengthening the body and restoring mobility. Therefore, simple movements such as those shown in Figure 2.3 are usually included in the physiotherapy session, along with exercises that have a specific goal to motivate the patient

(Masiero and Carraro (2008)). Doing these exercises repetitively and with progressive intensity helps to improve limb function, however, it is necessary to note that rehabilitation must be studied and adapted to each patient in order to perform the exercises that best suit their condition, as they depend on the type of stroke and the conditions/consequences that have occurred in them (You et al. (2014); Chien et al. (2020); Jan et al. (2019)).

2.2 Stroke impact analysis

Analysing the impact that a stroke has on the mobility and functionality of a person's limbs is very important in order to be able to carry out the therapy that the patient needs. For this appears the Fugl-Meyer Assessment (FMA) that is considered a model for obtaining a quantitative analysis of both the impact of the stroke and the recovery/evolution of the person after therapy (Gladstone et al. (2002)). Unlike other assessments, such as the Stroke Impact Scale (SIS) (Duncan et al. (1999)) and the Montreal Cognitive Assessment (MoCA) (Nasreddine et al. (2005)), which focus respectively on the larger impacts related to stroke and cognitive function, the FMA stands out for its specific focus on physical analysis, making it particularly suitable for guiding specific rehabilitation strategies aimed at restoring motor function.

The test consists of several analysis items, divided into 5 areas: motor function, sensory, balance, joint range of motion and joint pain (Gladstone et al. (2002)). In each area there are several tests and the health professional must score each one from 0 to 2, where 0 indicates no response, 1 indicates a partial response and 2 indicates a complete response. The final score is calculated by adding up the scores from all the tests, the maximum being 226 points: 100 for motor tests, 24 points for sensory analysis, 14 points for balance exercises, 44 points for joint range of motion and 44 points for joint pain. The higher the score, the greater the patient's motor and functional recovery. In the tests of motor function all the movements represented above, except for wrist radial/ulnar deviation, are analysed, since they are essential movements in daily life.

Often, the testimonies of patients who have had a stroke and suffer sequelae can highlight specific difficulties caused by limited movement. For example, many mention difficulties in carrying out everyday tasks, such as drying their hair or cooking, and note that recovery is a slow process. Moreover, even with physiotherapy, adaptation to everyday life is essential, such as the use of the left hand by right-handers when the right hand is affected (Poltawski et al. (2016)).

Furthermore, it is crucial to consider the evolution of the patient's disability conditions over time. Consequently, therapy must be adapted to meet the patient's needs as they change. This continuous adaptation is essential to ensure that the treatment is effective and that the patient receives the necessary support at each stage of their recovery (Raghavan (2015).

As mentioned earlier, loss of sensation and paralysis are common consequences and will greatly influence the therapy given. For this reason, objective tests are important to understand the patient's ability to move, but it is also important to take into account what the patient reports and thus adapt the rehabilitation.

2.3 Detection of limbs movement

Detecting and tracking human pose is important for several areas, such as disease recognition, film production and analysing movements in sports (Chen et al. (2018a); Wang et al. (2021)). Several methods can be used for this - they are presented in the sequel and in Table 2.1 are summarised their advantages and disadvantages.

- Inertial Measurement Unit (IMU) sensors - these sensors typically consist of an accelerometer - measures changes in linear velocity, a gyroscope - measures changes in rotation and a magnetometer - measures the strength and direction of the magnetic field. Using algorithms, the signals registered by the sensors are processed and translated into the position and orientation of an object. By positioning these sensors strategically on the human body, and with the help of algorithms, it is possible to track the various limbs, and it is a method that is widely used because it leads to very precise results (Białecka et al. (2023); Tsilomitrou et al. (2021); van der Kruk and Reijne (2018)).

- Marker-based optical motion capture system - markers are distributed throughout the body and, using a camera and an algorithm, are tracked giving the 3D coordinates of each one. This way, the position of the various limbs can be reconstructed. The markers are usually ligth-emitting diodes (LED) or circles coated with a reflective material (Zago et al. (2020); van der Kruk and Reijne (2018); Zhou and Hu (2008)).

- Markerless-based optical motion capture system - the movement and position of the person's limbs are acquired without the use of markers, through a camera and algorithms that will identify the position and detect the coordinates of the joints. This method is advantageous because it allows the patient to move freely, and the preparation for the session is quicker (Zago et al. (2020); Zhou and Hu (2008)).

- Electromyography (EMG) signal - by placing electrodes to measure the electrical potential produced by the muscles when they contract and using a deep learning algorithm, it is possible to estimate the position of the limbs. In order to obtain precise results, it requires training for each patient, as well as personalised calibration, since it will depend on each person's muscular activity (Liu et al. (2021); Trigili et al. (2019)).

- Electroencephalogram (EEG) signal - monitoring electrical activity in the brain by placing electrodes on the scalp. This method provides information on the patient's intention to move. It is low-cost, non-invasive and has high temporal resolution; however, its spatial resolution is low, which makes it inaccurate, and the signal is weak (Asanza et al. (2022); AL-Quraishi et al. (2018)).

Although the use of IMU sensors is quite common for motion analysis, they need to be precisely positioned and can condition the movements made, as is the case of systems with markers,

which, in addition to these factors, are expensive and complex. On the other hand, systems without markers are quicker, simpler to install and have no movement constraints, but require a more complex and stable algorithm to be able to detect the patient's limbs (Hernández et al. (2021)).

As mentioned earlier, a patient's motivation can significantly impact the achievement of positive results. Detecting movements through EEG and EMG can enhance patient engagement by enabling the detection of movement intentions. This way, patients are more likely to invest in their movements. However, it should be taken into account that patients with disabilities often have muscle spasms, which can lead to EMG detection being questioned. In addition, at the time they are incapacitated, the stimulated movement may align with the patient's capabilities rather than the correct movement. In this respect, EEG detection will be more stable (Blank et al. (2014)).

When evaluating the accuracy of various methods for detecting movement, a variety of results were observed, influenced by different factors. For example, in studies using IMU, as demonstrated in Tsilomitrou et al. (2021)'s work, a root mean square error (RMSE) of 0.02 m was reported in a 3D reconstruction of upper limb movement, with factors such as the calibration method having an impact on this result. On the other hand, marker-based systems, such as those employed in Merriaux et al. (2017)'s study, revealed a position error of less than 0.002 m in dynamic experiments using the Vicon system. With regard to EMG and EEG methods in the elbow flexion/extension movement, the study by Silva-Acosta et al. (2021) revealed an RMSE of 8.59º and 9.44º, respectively, and an RMSE of 9.53º when EMG and EEG are combined, with variations influenced by factors such as gender. These investigations highlight the complexity of assessing the accuracy of movement detection methods, emphasising the importance of considering various factors when interpreting their results.

In order to detect limbs without markers, cameras need to be used so that the image can be captured and then computer vision algorithms can make the desired detection. There are a few types of cameras used in pose detection, including Red-Green-Blue (RGB) and Red-Green-Blue-Depth (RGB-D) cameras. The former are often used because they are easier to find and cheaper, although pose detection in 2D is relatively easy, when moving to 3D this is no longer the case, making it a challenge in this area, and many of the methods already found are unstable. RGB-D cameras provide an RGB image and the depth information. These cameras are usually more expensive and sometimes, when used outdoors, fail due to interference from the sun (Mehta et al. (2017)).

There are several types of depth cameras: Time of Flight (ToF), stereoscopic cameras and structured light cameras. Cameras with ToF technology emit infrared (IR) light, which is reflected by the object and detected again by the camera, so they measure the time of flight of the light - the time it takes to reach the object, and back. This allows it to calculate the distance to the object. Stereoscopic cameras are made up of several lenses and determine corresponding points between the pairs of images, thus estimating the distance to the object. Structured light cameras determine the distance by comparing the emitted light pattern with the captured one, observing its deformation (Chiu et al. (2019)).

Table 2.1: Advantages and disadvantages of the existing detection methods.

Detection method	Advantages	Disadvantages
IMU sensors	Precise results	Need to be precisely positioned and can condition the movement
Marker-based systems	Precise results	Needs a precise position of the markers, can condition the movement and are expensive and complex
Markerless-based systems	No movement constraints and have a quick and simple preparation	Needs a stable and complex algorithm
EMG signals	Motivate the patient	Need a precise electrode positioning, an personalised calibration and spasms can interfere in the process
EEG signals	Motivate the patient, it is low-cost and non invasive	Need a precise electrode positioning and can stimulate the patient to do the wrong movement

Each detection method can be used depending on the objective of the study. However, markerless methods are the most advantageous if the aim is to optimise the time it takes to detect limbs and their movement. Within this form of detection, the use of computer vision has emerged as a powerful tool for analysing limb kinematics (Debnath et al. (2022)).

2.4 Computer vision for limbs detection

Pose detection is a problem that has been intensively studied in the field of computer vision since there are several areas that can benefit from pose detection and tracking (Debnath et al. (2022)). A good pose detection should be robust and stable, and should not be influenced by deformations or obstructions, by lighting or by the clothes people wear (Newell et al. (2016)).

Normally, to obtain the 3D joints' coordinates there are 2 phases to the process: the first where the pose is detected and the second where the 3D pose is reconstructed. Pose estimation can be done using various methods presented bellow (Newell et al. (2016); Shotton et al. (2013)).

- One-stage approach - locates and detects people's joints in a single step. Although it is simpler than the other approaches and does not accumulate errors, it requires a lot of training data to obtain a good pose detector and tracker.
- Top-down approach - locates the various people in the image and then detects the joints. Despite obtaining good results, the fact that it is a 2-step approach makes it slower.
- Bottom-up approach - detects all the joints in the image and only then groups them together to construct each person's pose. Errors can occur when it comes to detecting smaller people, for example, because the scales may be different.

The detection of several people in the same image is more complex than when there is only one person in the image, but the above approaches can be used in both cases (Fang et al. (2023)).

However, the last two approaches are more orientated towards the detection of multiple people (Li et al. (2018)).

There are different models that can be used for pose detection (Wei et al. (2016)). The pictorial structures model is a classic model in which the body is divided into several parts (arms, legs, among others) connected by joints, using a tree structure where each node is a part of the body. It also assumes a priori that all the parts are connected, so it makes a spatial correlation between them. This model is successful in images where people have all their limbs in view, but it has difficulty relating non-adjacent limbs (Pishchulin et al. (2013)).

The hierarchical model is based on coarse-to-fine detection, in other words, it goes from general to detailed, detecting the body first and only then moving on, for example, to the limbs, hands and face, since larger structures are often easier to recognize, also helping to detect smaller parts (Nie et al. (2019); Sun and Savarese (2011)). In the sequential prediction framework, predictions are made for each body part in each cycle, and these predictions are iteratively improved until a good degree of confidence is obtained (Ramakrishna et al. (2014)).

Many existing models use convolutional architectures, which use convolutional neural networks (CNNs) to perform pose detection. They use several stages to extract features and can learn patterns automatically. Different activation and loss functions can be used to optimise this method, but what has the greatest impact is its architecture (Khan et al. (2020); Tompson et al. (2015)). These pose detection and computer vision approaches, when integrated into advanced robotic systems, play a crucial role in limb rehabilitation (Debnath et al. (2022)).

2.5 Robotic systems for rehabilitation

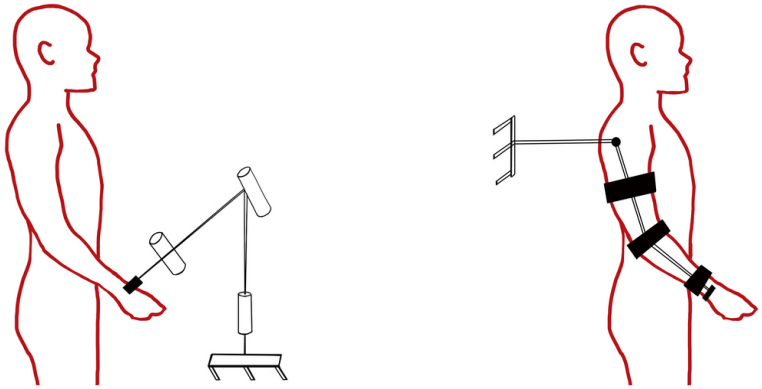
It is important to remember that not all robotic systems are suited to working in the same environment as humans. Therefore, these can be divided into two categories: industrial and collaborative systems. The former require safety systems as they are not expected to interact with a person during their task, only in preparation for it, when the system is in manual mode; they usually aim to do repetitive or heavy work. Collaborative systems, also known as cobots, are systems designed to share space and tasks with the operator, and which themselves provide security in this interaction (Villani et al. (2018)).

Orthosis and prosthesis are two types of robotic devices. The first one consisting of devices designed to assist, stabilise, repair or just improve the functional capacity of a part of a person's body, while prosthesis are artificial devices that replace a part of the body that has been lost (Gopura et al. (2016)). Orthosis include two different categories of devices - a device that aligns its joints with the user's (exoskeleton) and a manipulator where an end-effector (possibly a gripper) will be attached to a patient's limb. Both technologies are used to support upper limb rehabilitation, which makes it important to understand the differences between them. The systems are described below and in Table 2.2 the advantages and disadvantages of each are presented.

- Manipulators systems - as it can be seen in Figure 2.4a, there is no alignment between the joints of the patient and of the manipulator. The manipulator has a distal movable handle -

the end-effector that can be positioned in specific areas of the body and the patient's attached limb will follow a specific trajectory (Molteni et al. (2018); Gopura et al. (2016)).

- Exoskeleton - as it can be seen in Figure 2.4b, there is a correspondence between the human joints and those of the device, and each joint has a pre-programmed trajectory calculating the torque required for each one. They can be classified according to the number of joints controlled and can be unilateral or bilateral devices (Molteni et al. (2018); Gopura et al. (2016)).



(a) (b)

Figure 2.4: Illustrative examples of two types of orthosis according to their structure: (a) Manipulators system; (b) Exoskeleton system. From (Molteni et al. (2018)).

Table 2.2: Advantages and disadvantages of manipulators and exoskeleton systems. Adapted from (Sheng et al. (2016); Qassim and Hasan (2020)).

System	Advantages	Disadvantages
Manipulator	Adapt to different sizes and shapes of movements	Cannot provide rotational movements, as pronation and supination movements, that can be seen in the Figure 2.3c.
	The same manipulator can be used for both sides of the body.	Redundancy in trajectory
Exoskeleton	Include the joint axes of the limbs, providing a very specific movement	Since there is human symmetry, two exoskeleton are needed to rehabilitate the right and left limbs, requiring more funding

The purpose of having an automatic system is to do what a physiotherapist would normally do, which is to support the exercises and movements made by the patient, making the physiotherapy more objective and precise. Furthermore, they also have the advantage of allowing a professional to have more than one patient per session and still ensure that the exercises are correctly executed. That said, it is possible to divide and classify rehabilitation methods depending on the function the systems have in the exercise. The different treatment approaches are presented in Table 2.3.

It should be emphasised that rehabilitation with active movements requires voluntary effort on the part of the patient and consequently provides significant motor improvement (Xing et al. (2018)).

Table 2.3: Different treatment approaches depending on the interaction with the robotic system. Adapted from (Basteris et al. (2014)).

Type of therapy	Description
Passive	The limb is controlled and moved by an external element, this could be the physiotherapist, but in the case of automation, this is the automatic system's task. In both cases, it requires no volunteer effort performed by the patient. It is usually used in the early stages, as there may be no response from the limb at the beginning.
Assistive	Requires a signal from the patient to perform the movement and follows the patient's intention to move the limb, providing a force to help the movement.
Active-assistive	It is the patient who makes the movement and there will only be intervention if the patient is unable to complete the exercise. At this point, the approach is similar to passive therapy.
Active-resistive	A force is applied against the movement to further stimulate the limb. It is the patient who makes all the movement and effort, so the patient needs to be able to move the affected limb, even with little amplitude.
Active	The system is only used as a measurement sensor and does not interfere in any way with the patient's movement.
Corrective	When the patient makes a mistake in the exercise, the system stops the movement and forces the patient to adjust to the correct position.
Path guidance	The device corrects the patient when they go off the path that has been defined.
Passive mirrored	Is used when one limb is functional and the other is impaired, following a mirror concept in which both limbs make the same movement, but it is the unaffected limb that controls the passive movement of the injured limb.

2.6 Summary

In this chapter it was possible to focus on concepts that will be needed further. Including anatomical concepts, which will help to understand how the movements made are influenced. Movements such as abduction, extension and flexion of the elbow and shoulder are very important in physiotherapy, both to study the patient's progress (through the FMA) and for the therapy itself.

It is important to know how to distinguish between exoskeletons and manipulators and how they interact with the patient in rehabilitation. In addition, there are several ways of detecting the limbs, but the ones without markers are the most advantageous, since it allows to move freely and the preparation is usually quicker.

It should be realised that the accuracy of the various methods is influenced by several factors, such as the calibration method, the algorithm used and the way it is trained.

Chapter 3

Literature Review

As mentioned earlier, pose detection is an area that has been studied extensively due to its application in different areas. Libraries for pose detection have been developed for 20 years (Chung et al. (2022)). Many algorithms are well known for this purpose, including OpenPose (Cao et al. (2017)), PoseNet (Smilkov et al. (2019)), DeepPose (Toshev and Szegedy (2014)) and others. Often the algorithms developed are based on older structures.

All the projects face challenges, especially when multiple people appear in the plan. The fact that it is not known how many people will appear in the plan, in which zone or scale, as well as partial occlusions are factors that will influence the results and all the work aims to create or improve the algorithm so that it is robust to these challenges.

Automatic support systems have been developed for a long time, but it was in the 1990s that developments in the field of robotics emerged and made significant progress (Gopura et al. (2016)). The first systems that appeared worked only in a horizontal plane, with just 2 DoF and used end-effector manipulators for assistance. Exoskeletons appeared a little later with the aim of reducing the ambiguity of movements (Blank et al. (2014)).

This chapter reviews the literature on the computer vision methods used to detect upper limbs and robotic systems that have been developed in order to assist the physiotherapy.

3.1 Pose detection: approaches and technologies

OpenPose, the first open-source system to detect the 2D pose of multiple people in real time, was developed by Cao et al. (2017). It is a high-quality, accurate bottom-up system that localises 135 keypoints, with a mean Average Precision (mAP) of 75.6%. This algorithm is based on convolutional neural networks (CNN), and the pose detection process begins with a network to identify the areas of the body - arms, legs, head and others. The input are the images or videos. Confidence maps are generated in which each one shows the areas with the highest and lowest probability of containing a certain part of the body. Simultaneously, the connection between two parts are determined using the so-called part affinity fields (PAFs), which provide a 2D vector

associated with each pixel indicating the position and orientation of each limb. This results in the final keypoints of the human skeleton. The process is illustrated in Figure 3.1 below.

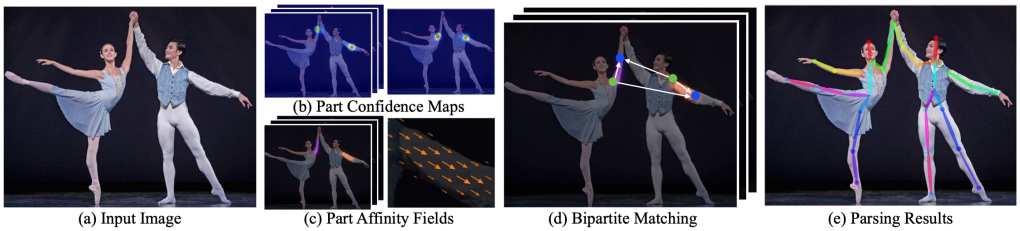


Figure 3.1: OpenPose algorithm. From (Cao et al. (2017)).

AlphaPose, created and presented by Fang et al. (2023), like OpenPose, takes images or videos as inputs. Initially, it uses an object detection model to generate bounding boxes around each person detected in the image. These bounding boxes are then cropped to create sub-images of the original image, each containing a single person. Subsequently, a neural network is applied to each sub-image to predict the positions of the keypoints of the person, producing a heatmap for each keypoint. In other words, areas with higher or lower confidence of containing a keypoint are identified. The keypoints are determined by finding the points of highest intensity in the heatmaps. Finally, there is a post-processing step where the positions are refined. This process is illustrated in Figure 3.2. It is important to note that each block presented is independent, allowing for acceleration in the application. The FIFO queue means First In First Out, which means that each process stores the calculated results of the current module, and the following modules retrieve the results directly from the queue. Moreover, it uses a pose distance metric to compare the similarity of poses and effectively eliminate redundant poses.

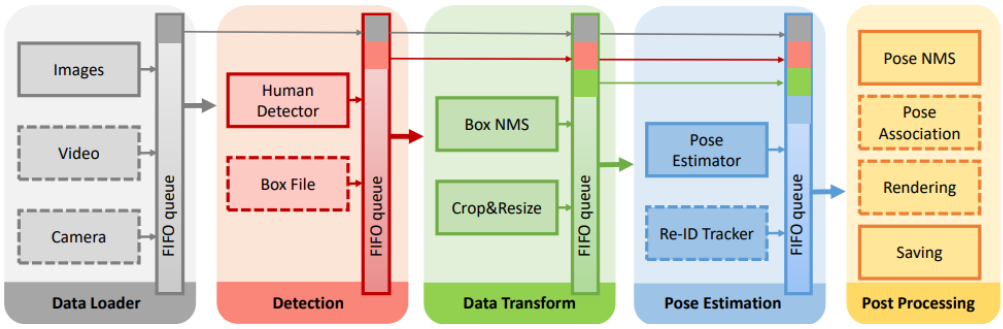


Figure 3.2: AlphaPose algorithm architecture. From (Fang et al. (2023)).

PoseNet was developed by a team at Google (Smilkov et al. (2019)), and only uses a network called MobileNet (Howard et al. (2017)), which was developed for mobile devices and is lighter. Firstly, relevant features are extracted from the image and then the positions of the joints are estimated, producing heatmaps. In addition, offset maps are produced to predict the lines connecting the keypoints. Using both types of maps, the algorithm estimates the final position of the person's limbs, providing coordinates for each keypoint.

The Cascaded Pyramid Network (CPN) was presented by Chen et al. (2018b) also based on CNNs, more specifically, on two sub-networks - GlobalNet and RefineNet. Figure 3.3 presents the procedure that will be explained as follows. GlobalNet uses convolutional layers to extract features from images at different levels of resolution and convolutional filters are applied to generate heatmaps which, as in the previous cases, provide information on the probability of containing a keypoint. In this case, the heatmaps produced in higher layers will have more spatial resolution but will have less information; the reverse occurs in deeper layers. Therefore, in order to maintain a balance between the two pieces of information, 1x1 convolutional are used before the element summation process in the upsampling process, where the upsampling serves to increase the spatial resolution of the features and the 1x1 convolutional aim to adjust the semantic information. This process is able to detect keypoints such as the eyes, however, for areas called hard keypoints (such as keypoints that are occluded), RefineNet was required. At this stage, there will be a concatenation of all the pyramid features and more bottleneck blocks stacked into deeper layers, in order to obtain a more correct result for the hard keypoints, achieving a mAP of 72.1%.

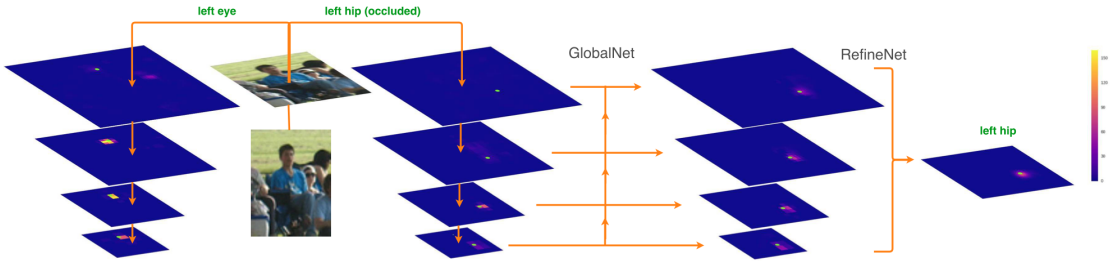
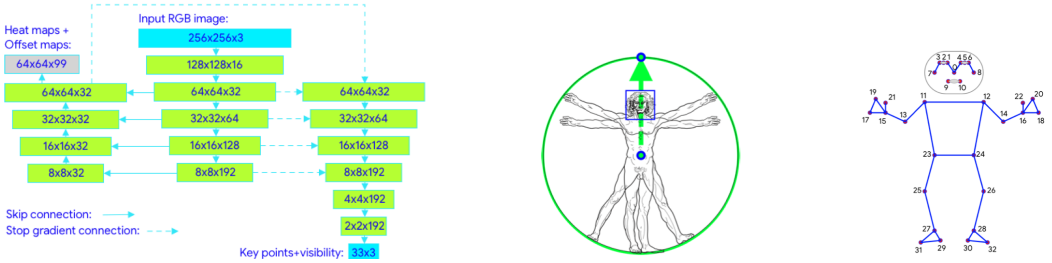


Figure 3.3: Cascaded Pyramid Network process. From (Chen et al. (2018b)).

DeepPose uses a deep neural network (DNN) which learns the features and then, through a regression layer, directly predicts the coordinates (x, y) of the keypoints. Toshev and Szegedy (2014) proposed a cascade of DNN-based pose predictors, increasing localisation accuracy. The localisation of each joint is regressed, taking the full image as the input for each one, with the advantage of capturing the entire context and the fact that it is simpler to formulate, however, sub-images with better resolution are formed where the localisation is improved. Figure 3.4 shows what occurs in this algorithm, for which the authors obtained a mAP of 69%.

Figure 3.4: DeepPose algorithm. From (Toshev and Szegedy (2014)).

BlazePose was developed more recently by a team from Google and is integrated into the well-known MediaPipe (Bazarevsky et al. (2020)). The network architecture is presented in Figure 3.5a. Firstly, a face detector is used so that it is fast to know if there is a person in the plane or not, and it's taken for granted that whenever someone is in the plane of the image, their face will be too. In addition, virtual keypoints will be detected, which are the midpoint of the hips and shoulders, the angle of inclination of the line connecting the two midpoints and the radius of a circle that will circumscribe the person, making the detection more robust and consistent (Figure 3.5b). This algorithm will detect 33 keypoints on the human body, as illustrated in Figure 3.5c. It only detects points considered essential to be able to estimate the position, size and rotation of the areas of interest. It is a two-stage algorithm, first identifying zones of interest - through heatmaps and offsets maps - and then determining the location of the keypoints - using a regression encoder. This encoder receives the information on zones of interest and position refinements from the first stage and produces the 2D coordinates of the keypoints, with a mAP of 84.5%.



(a) (b) (c)

Figure 3.5: BlazePose algorithm: (a) Network architecture; (b) Keypoints that will be detected and its distribution; (c) Virtual keypoints represented. From (Bazarevsky et al. (2020)).

DensePose is another widely used algorithm (Güler et al. (2018)) with an mAP of 71.1% and its aim is to find the dense correspondence by partitioning the surface, mapping all the pixels in an image to create a 3D surface. To do this, as shown in Figure 3.6, each pixel must be classified as belonging to one of the surfaces (of the various body parts) or to the background. Segmentation masks are formed for each part of the body, with 25 regions, including one for the background. Next, there is a cascade of proposals for regions of interest (RoI), extracting the most specific features of each region and a regression system will determine the 2D coordinates of each pixel and map them onto a 3D surface.

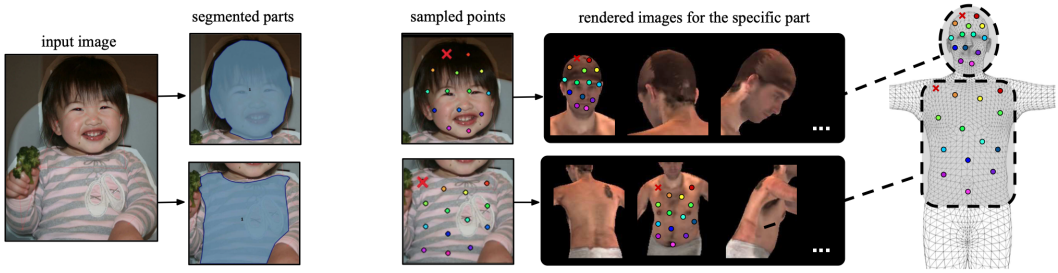
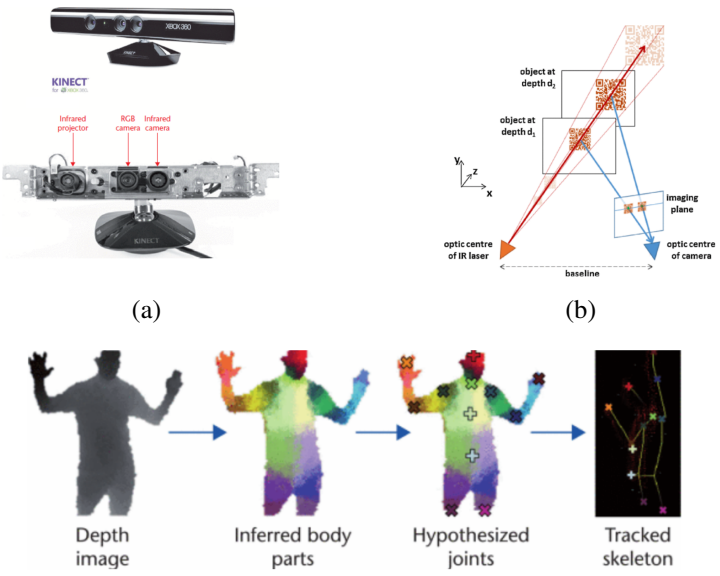


Figure 3.6: DensePose algorithm. From (Güler et al. (2018)).

Besides detecting the pose using the algorithms mentioned above, RGB-D camera systems can be used where the sensors themselves and the associated libraries often include integrated algorithms to perform keypoint tracking and skeleton detection in real time. These algorithms are designed to analyse the depth data and RGB images captured by the sensors, automatically identifying the key joints of the human body. A possibility is to use the Kinect v1 camera, which uses the Kinect Software Development Kit (SDK), developed by Microsoft (Zhang (2012); Han et al. (2013)). A SDK is a set of tools, libraries, documentation and code examples provided by a software manufacturer that gives developers everything needed to create software that integrates with a particular technology (Developers (2014)).

Figure 3.7a shows the layout of the elements of this camera. The depth sensor is the combination of the infrared (IR) projector and the IR camera. The IR projector emits a pattern of IR light points in the environment that is projected onto the entire viewing area of the camera and the IR camera will detect the IR image that corresponds to the reflection of the emitted rays on objects. As illustrated in Figure 3.7b, knowing the distance between the projector and the camera, and the pattern emitted by the IR detector, it is possible to triangulate the points, matching each point in the emitted pattern with the point in the IR image. It should be noted that the resulting depth map is in grey scale, where the lighter pixels mean that the object is further away from the camera. Black pixels mean that it was not possible to detect it correctly.

This system was also developed to enable tracking of the human skeleton as shown in Figure 3.7c, starting with a pixel-by-pixel classification of each part of the body, then the most likely places to have a joint, using the mean shift algorithm to identify areas with a higher probability density on the map. Finally, the hypothetical joints are mapped and the skeleton is adjusted, taking into account temporal continuity and prior knowledge of the training data.



(a) (b)

(c)

Figure 3.7: Kinect system: (a) Elements in Kinect camera; (b) Triangulation of points; (c) Detection process. From (Zhang (2012); Han et al. (2013)).

Chiu et al. (2019) and Mejia-Trujillo et al. (2019) compare the Microsoft Kinect v2 camera with Intel RealSense D435. The Kinect v2 camera uses time-of-flight technology to determine the distance to the object. Thus, light is emitted and detected again, and by knowing the time it took to be detected it is possible to calculate the distance to the object, since the speed of light is constant. RealSense D435 is a stereoscopic camera, which means it uses two lenses and makes a point-by-point relationship between the image and calculates the distance to the object. Ultimately, both authors concluded that the Kinect v2 camera was more accurate at detecting 3D points. In Table 3.1 are presented the characteristics of all three cameras discussed - Microsoft Kinect v1, Microsoft Kinect v2 and Intel RealSense D435. The major difference is the technology used in each.

Table 3.1: Cameras' characteristics. Adapted from Mejia-Trujillo et al. (2019)

Characteristic	Kinect v1	Kinect v2	RealSense D435
Technology	Structured Light	Time of Flight	Stereoscopic
Resolution	640×480	1920×1080	1920×1080
Depth resolution	320×240	512×424	848×480
Frames per second (FPS)	30	30	30
Frames per second (FPS) Depth	30	30	90
Depth Range (m) (for skeleton detection)	0.8 - 4.0	0.5 - 4.5	0.2 - 10
Field of View (Depth)	57°×43°	70°×60°	85.2°×58°

It is important to note that although the depth range of the RealSense D435 camera is 10 metres, Mejia-Trujillo et al. (2019) were only able to track the pose up to 4.5 metres, using the Nuitrack SDK. All the RGB-D cameras presented use a different technology to detect depth, with the ToF technology measuring depth directly while the other two technologies triangulate it. The Kinect SDK is able to detect ten joints of the upper limbs if the person's lower limbs are not visible, allowing the detection of the upper extremities if a person is sitting, for example (Han et al. (2013)).

That said, it is important to understand whether these methods are accurate for use in a rehabilitation environment. For this purpose, Hernández et al. (2021) carried out a comparative analysis of some existing methods for detecting the human skeleton. The authors compared the OpenPose method and Detrecton2 with the results obtained using two Kinect cameras, which are used as groundtruth. The first algorithm has already been explained; the second can be described as a combination of DensePose, also explained above, and Mask R-CNN, an algorithm that detects and segments objects. The study compared the angles between the shoulder and the hand, and between the hand and the trunk, in exercises such as elbow and shoulder flexion/extension, and concluded that OpenPose is an algorithm with greater precision, with an RMSE of 7.9º, and is fast enough to be used in a therapy context.

3.1.1 Key findings

Based on the methods discussed for human pose detection, it is possible to draw significant conclusions about their characteristics and applicability. OpenPose and AlphaPose are both a robust and precise choice for pose detection. Meanwhile, PoseNet, opting for the lightness of the MobileNet network, is an efficient option, especially on mobile devices.

Cascaded Pyramid Network (CPN) and DeepPose, both based on CNNs, demonstrate approaches to improve spatial resolution and keypoint localisation accuracy. BlazePose, incorporating facial detection and virtual keypoints, presents a more recent and robust solution for detecting 33 keypoints. DensePose stands out by searching for dense correspondence and mapping pixels on a 3D surface. In relation to RGB-D cameras and their SDK, the comparative analysis between Kinect v1, Kinect v2, and RealSense D435 reveals that Kinect v2 demonstrates greater accuracy in detecting 3D points.

3.2 Robotic systems

Using robotic systems to assist in physiotherapy is something that is being extensively studied because of the impact they can have. There are different systems that can already be used, but they all have the same objective, which is to support the patient during physiotherapy. Currently, many solutions combine robotic systems with virtual reality and video-gaming to stimulate patients. Instead of making free movements, there is a game with a goal of challenging the patient to different movements. In this case, the robot will serve to replicate the user's movements, providing feedback on force, angles and other parameters of the patient's movement throughout the exercise (Rodríguez-Hernández et al. (2023); Morone et al. (2014)). Examples of these systems that have been developed are described below and shown in Figure 3.8.

There are studies showing an improvement in rehabilitation with the use of these systems. Indeed, Lum et al. (2002) compare the evolution of conventional therapy with therapy using a system called Mirror Image Movement Enabler (MIME) which has 6 DoF, allowing 3D movement control where the patient has a handle for holding and straps to secure the limb. It has 4 modes of operation - passive, active-assistive, active-resistive and passive-mirrored, and the movement (forces and torques) was controlled with a 6-axis sensor - the system can be seen in Figure 3.8a. In this study, two groups were compared and given the same rehabilitation time, one conventional and the other with the robotic system - 24 sessions of 1 hour each. The results show that there was a significant improvement in patients who had access to therapy with the robotic system, with improvements such as reduced disability, improved strength and increased range of motion.

There are some studies of systems that are presented in more detail to understand how the control is made and their specifications. More recently, Chen et al. (2016) developed an exoskeleton system called NTUH-ARM with 7 DoF. It can work in 3 modes - active, passive and assisted-as-needed, has two force/torque sensors and each joint has a potentiometer and an encoder to measure its position.

Figure 3.8: Examples of robotic systems for rehabilitation purposes: (a) MIME (Lum et al. (2002)); (b) GAVRe^2 (Lai et al. (2018)); (c) ANYexo (Zimmermann et al. (2019)); (d) Robert (Life Science Robotics (2020)); (e) InMotion 2.0 (BIONIK (2023)); (f) MOTORE (HUMANWARE (2023)); (g) ReoGO (SAMCON (2018)); (h) ArmeoPower (Hocoma (2023a)); (i) ArmeoSpring (Hocoma (2023b)).

Lai et al. (2018) used Game Adaptive Virtual Reality Rehabilitation (GAVRe^2), which uses the robotic system UR3 (Figure 3.8b) from Universal Robots as a support and virtual reality to motivate and involve the patient in the therapy, in order to help rehabilitate the upper limbs. It also uses force and torque sensors for patient-manipulator and manipulator-environment interactions.

Kyrkjebø et al. (2018) used Universal Robots' UR5 robotic system with 6 DoF and studied its safety and functionality. It uses three high-level control strategies - buoyancy mode, resistance mode and random mode to carry out these tests. In the first strategy the therapy is assistive, in the second it is active-resistive and the last is a challenge, where the system will make a force in random directions throughout the movement. In all modes, the results obtained show that this system is reliable for helping to treat patients.

In Figure 3.8c is shown a system called ANYexo developed by Zimmermann et al. (2019). An exoskeleton system for the arm and shoulder, which differs from the others in the range of movements it allows, resembling daily activities, more specifically movements that are closer to the body. It has two IMU sensors to measure forces and torques.

Oliveira et al. (2022) aimed to develop a framework to control the KUKA LBR iiwa robotic

system for upper limb rehabilitation using a game-engine unit. They conducted tests to determine the reliability of applying these systems in the medical environment. The device used is a 7 DoF robot and it proved to be viable for this task, having been emphasised that it is one of the safest systems on the market, due to having force sensors in each joint that allow the applied force to be known. The end-effector has a holding system with buttons that allow the operator to control the resistance offered by the system, as well as a button that allows the user to record the positions in which the system should move every 50 ms to repeat the exercise. It was concluded that these systems can be a great help in limb rehabilitation.

A system also used and that is shown in Figure 3.8d is the Robert from Life Science Robotics (Life Science Robotics (2020)). It is a robotic system with 7 joints, used to support lower limb rehabilitation and allows for passive or active-resistive therapy, with 10 levels of resistance available. Chan (2022) studied the monetary issue of obtaining such a system, which is very important since the cost of these systems is often a factor in not purchasing them. However, this study concludes that it is monetarily favourable to have such a system to support physiotherapy compared to therapists.

Furthermore, there are some systems already commercially available, such as InMotion 2.0, MOTORE, ReoGO, ArmeoPower and ArmeoSpring (Calabrò et al. (2021); Lee et al. (2020)). InMotion 2.0, as it can be seen in Figure 3.8e, is a robot with 2 DoF to support shoulder and elbow physiotherapy in the horizontal plane, developed by Bionik Laboratories (BIONIK (2023)). Collecting kinematic data using hand trajectory sensors, it employs active-assistive therapy to provide support whenever necessary to complete the exercise (Duret et al. (2019)).

Mobile robot for upper limb neurOrtho Rehabilitation (MOTORE), created by *humanware*, is a robotic system designed to support the recovery of an upper limb after a stroke. It has no links that can adapt the type of assistance to the patient's needs. Nevertheless, it can act as a support, help complete the exercises and even oppose movement. Thus, the device's function can be chosen according to the professional's understanding (HUMANWARE (2023)). The system is presented in Figure 3.8f.

Motorika has developed ReoGO, a robotic system that enables passive, active-assistive or active-resistive therapy, monitors movement and provides biofeedback, allowing movement in two or three dimensions. As illustrated in Figure 3.8g, the patient's affected limb is connected to the system with the help of elastic bands and using the stimulation on the screen the patient moves the affected arm (Taravati et al. (2022)).

Both ArmeoPower and ArmeoSpring, shown in Figures 3.8h and 3.8i, are exoskeleton systems for upper limb rehabilitation and they were developed by Hocoma AG. The ArmeoSpring allows 3D therapy with 6 DoF and is based on self-initiated movement, with an extensive library of games designed to stimulate the patient's movement, with specific objectives (Hocoma (2023b)). The ArmeoPower controls the exercise and supports the patient whenever it detects that they are unable to complete the movement. It provides support in an extensive 3D workspace, and allows rehabilitation of the arm and hand simultaneously (Hocoma (2023a)).

3.2.1 Key findings

The use of robotic systems in physical rehabilitation has been proven effective in several studies. From the MIME system to the integration of UR3 with virtual reality, these systems provide notable improvements in strength, range of motion and reduction of disabilities when compared to conventional therapy methods.

A growing tendency is to integrate these robotic systems with virtual reality environments and therapeutic games. This approach not only involves patients more actively in their rehabilitation process, but also provides valuable real time feedback on performance during exercises, making therapy more interactive and motivating.

The diversity of systems commercially available, such as ANYexo, ReoGO, ArmeoPower and ArmeoSpring, offers therapeutic options adaptable to patients' individual needs. From passive support to active resistance, these systems provide a wide range of therapeutic possibilities, enabling a personalised approach to upper and lower limb rehabilitation.

Despite the costs associated with robotic systems, studies such as the one that analyses the Robert system indicate a financial justification compared to the benefits obtained in rehabilitation. In addition, the generalised applicability of these systems in upper and lower limbs, combined with the ability to adapt to the patient's needs, highlights their relevance in advancing the quality of physiotherapy and promoting physical recovery.

3.3 Computer vision in robotics aided rehabilitation

There are other studies that combine computer vision algorithms with robotics in order to provide assistance in physiotherapy. The works carried out by Assad-Uz-Zaman et al. (2021) and Kapula et al. (2023) are the first step towards having robotic systems capable of replicating the movements made by people. These studies aim to enable a robotic system to move according to the pose of an individual, representing an advance in the interaction between humans and robots. In the first case, the aim is for each patient to have access to a representation of the exercise to be performed without having to go to a rehabilitation centre. To do this, as can be seen in Figure 3.9, the therapist demonstrates the exercises to be performed in front of a Kinect camera that detects the limbs and the NAO system performs the same exercises as the therapist. Similarly, in Kapula et al. (2023) a Pi camera is used to capture images in real time, which are then passed on to the RaspberryPi and the person's keypoints are tracked using the MediaPipe method. If there is movement, and based on the distance between the keypoints, the Raspberry Pi gives commands to the Arduino to control the servomotors that control the joints in the system. The flow chart of this system is presented in Figure 3.10a and the system in Figure 3.10b.

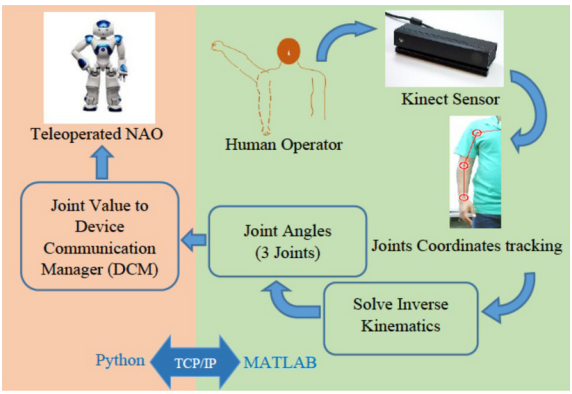
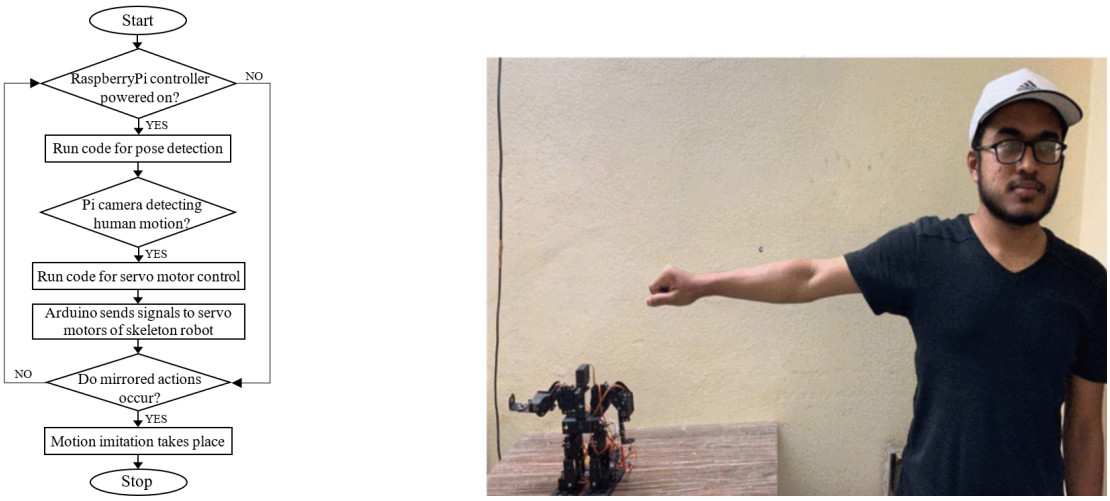


Figure 3.9: Workflow of the telerehabilitation system using the humanoid NAO. From (Assad-Uz-Zaman et al. (2021)).



(a) (b)

Figure 3.10: Overview of the system design and implementation: (a) RaspberryPi and MediaPipe based system flow chart; (b) RaspberryPi and MediaPipe based system. From (Kapula et al. (2023)).

Something that is also very important is analysing and evaluating the movement made by the patients, in order to understand if they are doing the exercise correctly. With this in mind, Chua et al. (2021) developed a telehealth service that will monitor the movements made by the patient using only a camera and the PoseNet algorithm - a simple scheme of this system is presented in Figure 3.11. This includes a statistical analysis in which the patient's angular movements are calculated and, at the end of the movement, a graph is drawn up with this information. In this way, therapists and doctors can analyse the patient's performance and progress. In a similar way, Gu et al. (2019) developed a system called ExerciseCheck where a reference exercise is established for each patient, defined with the therapist as an exercise that the patient can perform. At home, the patient has a Kinect camera in front of which the therapy session will take place and the movement made will be analysed, providing information on position and speed, and consequently feedback

to both the patient and the therapist. This study compared the use of a depth camera (Kinect) and computer vision algorithms using a RGB camera and concluded that the former method was not as accurate as using the depth cameras.

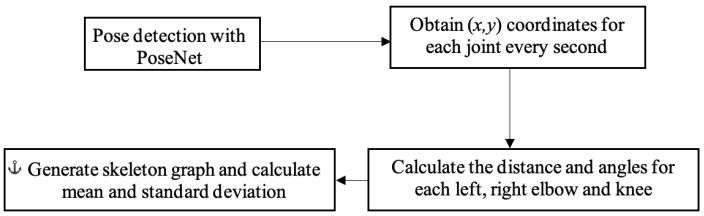
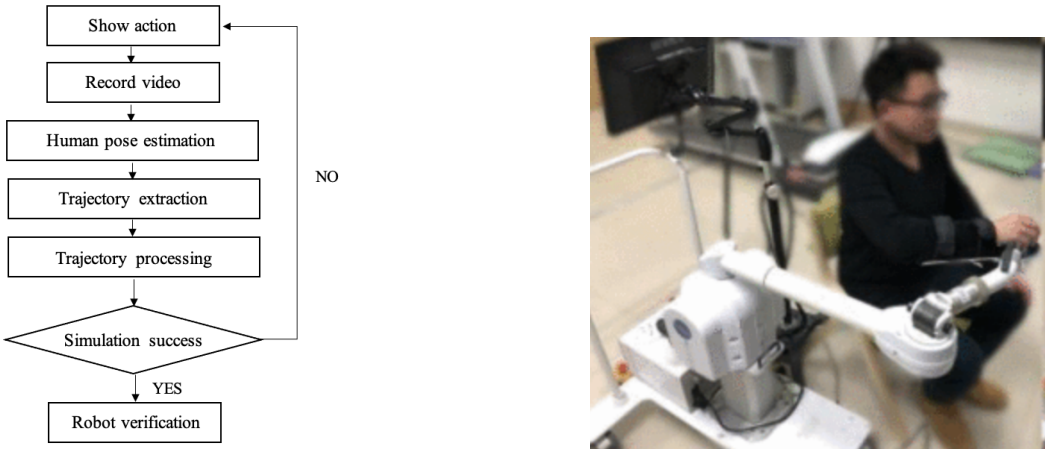


Figure 3.11: Scheme of the telehealth service. Adapted from (Chua et al. (2021)).

More complexly, Tao et al. (2020), by extracting information from the therapist's movement, controlled the robotic system called Burt that conducts the patient movement. Specifically, the therapist performs the movement in front of a Kinect camera and through the OpenPose algorithm the exercise is tracked and then the movement is performed by the robotic system - a manipulator with 3 DoF. The patient grabs the ball at the end of the system and it performs the movement passively. Thus, it was concluded that this is a simple and reliable method for rehabilitation using a robotic system. The flow chart is presented in Figure 3.12a and the system in Figure 3.12b.



(a) (b)

Figure 3.12: Overview of the Burt system design and implementation: (a) Burt system chart flow; (b) Burt system (Adapted from Tao et al. (2020)).

Rijanto et al. (2019) used computer vision algorithms to detect the position of the wrist during rehabilitation. The camera is on the manipulator itself, and the person sits on a table, takes a horizontal gripper and performs the therapy exercises. The aim of this work was to study the accuracy of the position results obtained with the algorithm, which were not excellent, depending for example on the area of the table on which they were working.

3.3.1 Key findings

The ability to replicate human movements, as demonstrated by Assad-Uz-Zaman et al. (2021) and Kapula et al. (2023), suggests the possibility of making visually representative therapies available to patients remotely, emphasising the potential of robotic systems to replicate human movements accurately. The detailed analysis of patient movements carried out by telemedicine services such as those proposed by Chua et al. (2021) and Gu et al. (2019) highlights the importance of monitoring performance in real time, providing valuable feedback for both patients and healthcare professionals.

While these innovations hold significant promise, their full clinical implementation requires continuous evaluation. Comprehensive validation of the efficacy and safety of these approaches in specific clinical contexts is essential to ensure that they effectively contribute to therapeutic goals. Close monitoring and adaptation as necessary are key to optimising the effectiveness of these systems, ensuring that they are aligned with clinical standards and patient needs. In order to ensure that the integration of computer vision into robotic systems in physiotherapy reaches the highest standards of quality, safety and clinical efficacy, there must be close collaboration between healthcare professionals, researchers and engineers.

3.4 Summary

In this chapter, it was possible to learn about some of the work that has already been done, both in the area of pose detection and in the area of robotics for physiotherapy assistance. Understand how it has evolved, is something very important and relevant to the work that is going to be done, in order to be able to realise the flaws and gaps that exist.

It is feasible to notice that the algorithms discussed are prepared to detect the whole pose in real time. However, changes can be made to obtain information on only the upper limbs, ensuring that if the physiotherapy is conducted with the patient seated, this does not prevents detection.

When comparing the use of RGB cameras and algorithms with the use of RGB-D cameras, it was possible to realise that the former already have associated libraries. Their use simplifies the development of applications involving gesture-based interactions, games, motion recognition and other applications that require pose detection, eliminating the need to implement key point tracking algorithms from scratch. In addition, the Kinect SDK allows the detection of only upper limbs, which is an advantage in the context of this project. Moreover, cameras like the Kinect appear in some studies as a groundtruth for comparing different methods, showing that the results can be trusted.

Systems such as InMotion or ArmeoSpring already exist commercially and can support therapy. It can be seen that the aim of the studies is to implement physiotherapy supported by robotic systems, allowing it to be more objective. However, no system implemented in physiotherapy has been found that has a component for assessing the patient's movement through computer vision algorithms and then applying it to the robotic system in order to stimulate a difficult movement for the patient.

Chapter 4

From Concept to Execution

This chapter presents the proposed solution, starting with the requirements necessary for its implementation. Next, the process of detecting and tracking the patient's limbs using Kinect v2 and its SDK will be detailed.

The manipulation with the UR5 robotic arm will also be discussed, including the important considerations and the essential parameters for defining movements. In addition, the development of the coupling part with the manipulator will be described in detail. Additionally, the integration of the system will be outlined, covering the variables required to enable complete automation of the process. Finally, the chapter will explore potential extensions and generalisations of this project to accommodate various movements.

4.1 Description of the proposed solution

The aim of this thesis is to implement a system that, using a RGB-D camera, is able to detect the movement made by the patient, then quantify its mobility and, finally, have a robotic system assisting the patient to perform the same movement in an improved way. For this purpose, Universal Robots' UR5 manipulator was used, with a 6 DoF, range of 850 mm and a payload of 5 kg (Universal Robots (2023a)). The case study was shoulder adduction/abduction, as they are simple but very important exercises.

The Kinect v2 camera was used with the Kinect SDK, which provides an interface for interacting with the camera. The SDK's input is the camera output - RGB images and depth data and the result are the 3D coordinates of each keypoint. The Kinect SDK uses the C# language. Once the upper limbs have been detected, the patient's movements must be analysed to understand their range of motion, which is done using the same programming language, C#, to ensure consistency in this part of the process. To do this, the opening angle of the arm must be calculated for the shoulder abduction/adduction movement. Subsequently, an algorithm was developed which progressively adapts and amplifies the range of movement, through an increment.

After this, an array with the final movement coordinates and orientations must be sent to the robotic system. The patient's limb should be connected to the end-effector, passively following the movement produced by the system.

To conclude, the proposed project can be divided into two parts: the first concerns the detection of the patient's upper limbs and the second involves the interaction between the patient and the robotic system. In the first part, it is necessary to ensure that the computer vision system is able to detect the limbs in a stable and robust manner.

The aim is to detect the upper limbs of just one person - the patient - with just one camera, in real time, to be robust when other people or objects are in the camera plane and even when there is partial occlusion. The camera must be facing the patient, parallel to the patient and as centred as possible. It is also important to ensure that the patient's arm, when extended in any direction, never reaches the limits of the camera's plane. In the second part of the project, focused in the patient-robot interaction, the robotic arm must be easily adaptable to different types of people and must guarantee patient' safety throughout its use. Additionally, it must be able to follow the required trajectory uniformly and stably.

4.2 System architecture

Figure 4.1 shows the architectural block diagram of the project, where the main components are a Kinect v2 camera, which detects and tracks the upper limbs in real time through the Kinect SDK, and the UR5 robotic system, which assists the patient in moving closer to the reference movement.

In detail, the therapist must establish the reference position that he/she considers that the patient should be able to reach at the end of the rehabilitation process, captured by the Kinect v2 camera. The 3D coordinates of the keypoints are then detected using the Kinect SDK and, after 15 seconds, saved. The patient performs the exercise suggested by the therapist to the best of their ability and the coordinates saved are the ones from the moment when the patient achieved their best performance.

The angles β and α, corresponding to the movements made by the therapist and the patient, respectively, are calculated, and an increment, i, is applied to help the patient approximate the therapist's movement. The expression of the increment will be explained in Section 4.5. The 3D pose is sent to the UR5 system, which guides the patient's arm through the motion, repeating it five times. This process is done at the start of each session to gradually align the patient's movement with the therapist's. By the end of the rehabilitation, the goal is for the patient to achieve the target angle independently.

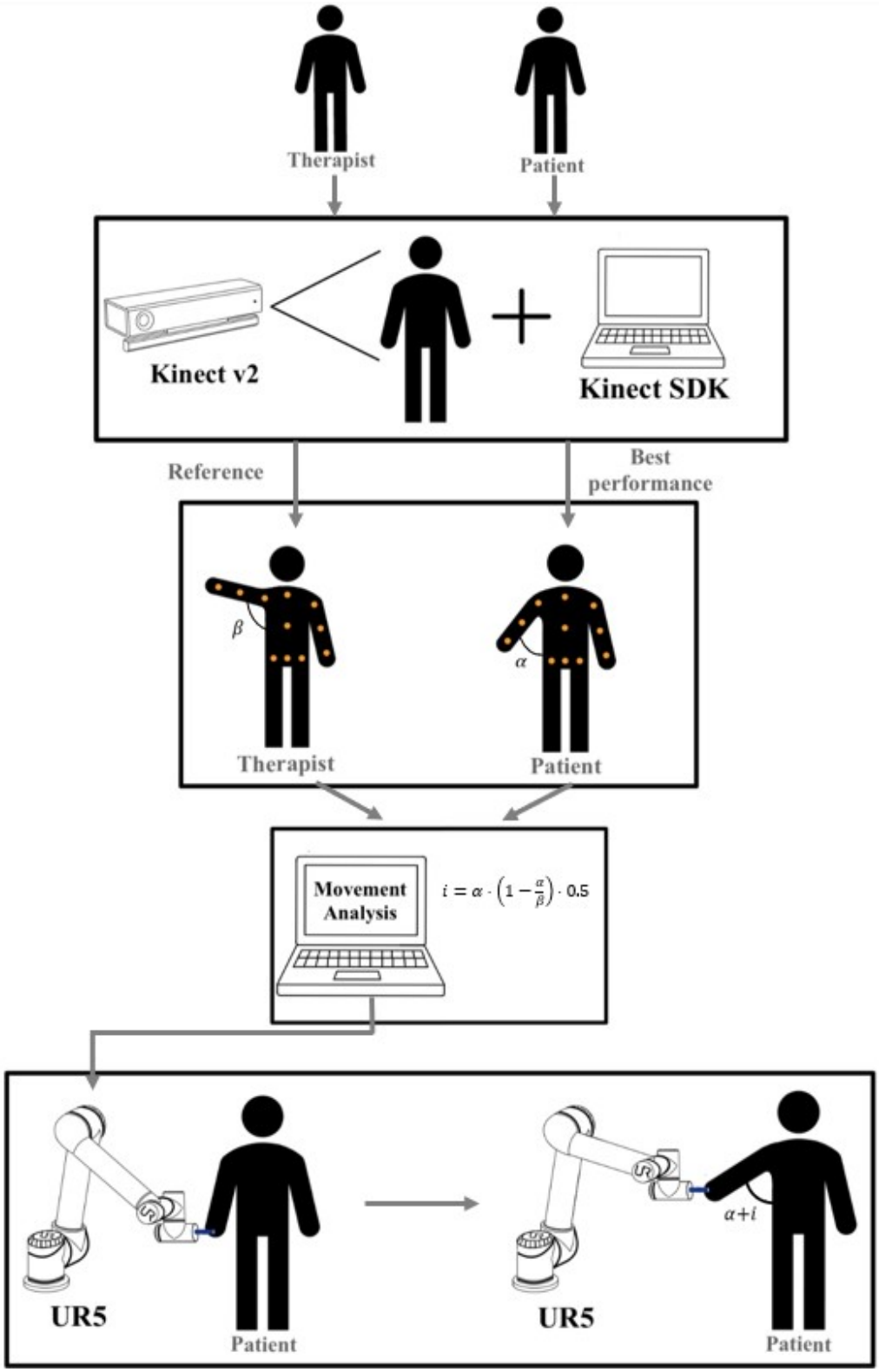


Figure 4.1: System architecture block diagram.

4.3 Detection of keypoints

As explained in Section 3.1, the Kinect SDK, designed for use on Windows operating systems connecting via USB 3.0, is a set of tools that allow developers to make more direct use of a particular device, in this case the Kinect v2 camera. Specifically, it provides the ability to obtain depth images and RGB images. Furthermore, it offers an interface for accessing and processing this data in real time. It allows the coordinates of the human skeleton's keypoints to be detected, which is what was used for this project (Developers (2014)). The SDK works by isolating areas with depth characteristics similar to the human body, and subsequently identifying the joints based on these characteristics. It supports the detection of up to six people, with detailed 25 joint tracking for two individuals. In this project, however, it is necessary that only one person is detected.

As stated before, the Kinect v2 was used as the camera to capture the images, so understanding its reference point is crucial for interpreting the results. As illustrated in Figure 4.2, from the perspective facing the camera, the xx axis is oriented from left to right, the yy axis is from bottom to top and the zz axis points directly out of the camera towards the viewer. Firstly, the methods used and the changes made to each one to evaluate the accuracy and effectiveness of the detection will be explained.

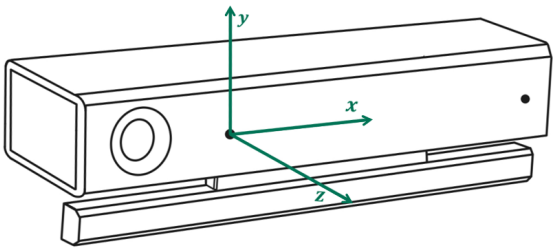


Figure 4.2: Referential of Kinect v2. Adapted from (Albert et al. (2020)).

In order to analyse the robustness of the detection, using the Kinect SDK, the goal was to analyse the accuracy of the detection in different planes and to understand how the methods reacted when there were two or more people in the camera's view, when the person was partially obstructed by other people or objects, when there were variations in background.

4.4 UR5 manipulator

Regarding the UR5 manipulator, it is important to note that it was used in collaborative mode, making it suitable for use around people and ensuring their safety. The robot is equipped with sensors that monitor the current; if these values exceed predefined limits, the robot automatically stops. Besides, the manipulator has a emergency button that can be pushed at any time needed. In both cases, if the UR5 stops automatically or forced, it is necessary to reinitialise the system to be able to use it again.

To use the UR5 manipulator automatically, a set of software frameworks for developing applications with robots called the Robot Operating System (ROS) is normally used, on Ubuntu. The

main advantage of this framework is that it enables a direct and simple connection to the robot, without the need to develop software from scratch. When the goal is to do a complex algorithm with many functions, it is proper to use it.

It is necessary to ensure that the manipulator is connected to the computer via Ethernet and both were on the same Internet Protocol (IP) subnet. The connection between the robot and the computer is made via sockets using the Transmission Control Protocol (TCP) through a specific port. This setup allows direct communication between the two systems, enabling the exchange of information and commands. In this case, the UR5 robot acts as the server, and the computer is the client, with data sent as a set of bytes containing the necessary instructions.

For controlling the UR5, URScript is used (Universal Robots A/S (2021a)). URScript is a language developed by Universal Robots for managing their manipulators, enabling direct command of the robot's movements. The *moveL* command directs the robot to move the tool of UR5 in a straight line to a specific point, based on given 3D coordinates and orientation, speed, acceleration and a blend radius. It ensures that the the tool of manipulator follows a precise linear trajectory.

To have the UR5 follow a specific trajectory, as a circumference, the *moveL* command can be used in a series of short linear segments, effectively simulating the arc and providing a close approximation to the desired circular path. This involved defining the centre, c_x, c_y, c_z, the radius of trajectory, R, a variable angle, θ, from 0 to 2π, and the number of trajectory points, num_traj. This trajectory was thought because of its proximity to the movements typically made in physiotherapy sessions, since any movement with the arm can be approximated to an arc of a circle.

The operation of the manipulator is influenced by the three parameters defined in *moveL* command - speed (v), acceleration (a), blend radius (r) - along with an additional one due to circular trajectory - the number of trajectory points (num_traj). These values must be adjusted to ensure that the movement is executed effectively and mimics the natural movement of the arm, providing reliable and comfortable rehabilitation. Each of these parameters is discussed in detail below.

- **Speed** (v): Speed defines how quickly the manipulator moves along the trajectory. The value of v should be sufficiently high to prevent patient fatigue, but not so high as to cause discomfort or risk of injury. Furthermore, since the UR5 is used in collaborative mode, its speed is limited to 250 mm/s (Universal Robots A/S (2021b)).

- **Acceleration** (a): Acceleration controls the rate of change of speed during the movement. It is essential to ensure that both acceleration and deceleration are gradual to avoid abrupt movements that could compromise the patient's safety.

- **Blend radius** (r): The blend radius defines the radius of a curve around the points through which the manipulator must pass through. When the manipulator enters this area defined by the radius, it considers the point as reached and begins transitioning to the next point. This parameter is crucial for smoothing the movement and making it more similar to the circular path. However, excessively large values of r can compromise the precision of the movement.

- **Number of trajectory points** (*num_traj*): This parameter determines the number of discrete points into which the trajectory is divided. A very small number of points can result in a segmented and less precise trajectory, while a very large number of points can cause the movement to become excessively slow, regardless of the speed and acceleration settings.

The UR5 manipulator has a mounting plate that can be coupled with several end-effector devices. For proper attachment, the end-effectors must conform to the specific design of the plate, which includes aligning with four screw holes. Therefore a part was designed in *SolidWorks*, which will be called the handle. In this case, it should be something simple that the patient can hold onto to make the movement with the passive arm coupled to the manipulator, but which, in the event of extreme pain, the patient can release, to abandon the movement. It must also allow the patient's hand to adapt throughout the movement.

4.5 Integration

As mentioned above, the movement of the arm is considered to follow approximately an arc of a circle with its centre at the shoulder, with the radius being the size of the arm and the angle varying depending on the movement made. It should be noted that this movement is intended to be made by the robotic arm, so all the information obtained must be taken into account and must be in accordance with the referential of the robot itself. Therefore, the following is an explanation of how each of these variables was obtained.

1. Centre of trajectory

   In order to calculate the centre of the circle, it is necessary to determine the coordinates of the shoulder in space in relation to the UR5's reference system. Since it is possible to obtain the 3D coordinates of the shoulder's keypoint in relation to the Kinect's reference frame, what can be done is to determine the transformation matrix, $\mathbf{H}_{\mathbf{K}}^{\mathbf{M}}$, which allows to pass the shoulder's coordinates from the Kinect reference frame (K) to the manipulator's reference frame (M). To do this, it is necessary to understand how the two references are positioned relative to each other. The transformation matrix will have the structure presented in Equation 4.1.

$$\mathbf{H}_{\mathbf{K}}^{\mathbf{M}} = \begin{pmatrix} & \mathbf{R}_{\mathbf{K}}^{\mathbf{M}} & & \mathbf{d}_{\mathbf{K}}^{\mathbf{M}} \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.1}$$

   From the $\mathbf{H}_{\mathbf{K}}^{\mathbf{M}}$ matrix described, the $\mathbf{R}_{\mathbf{K}}^{\mathbf{M}}$ is the rotation matrix, as the name suggests, it takes into account how the axes are rotated relative to each other, which is represented using Equation 4.2. The column vector $\mathbf{d}_{\mathbf{K}}^{\mathbf{M}}$ is the distance in x, y and z between the two references.

$$\mathbf{R}_{\mathbf{K}}^{\mathbf{M}} = \begin{pmatrix} x_K \cdot x_M & y_K \cdot x_M & z_K \cdot x_M \\ x_K \cdot y_M & y_K \cdot y_M & z_K \cdot y_M \\ x_K \cdot z_M & y_K \cdot z_M & z_K \cdot z_M \end{pmatrix} \tag{4.2}$$

2. Radius of trajectory

   With regard to the value of the trajectory radius, which is the size of the patient's arm, this measurement can be obtained by calculating the Euclidean distance between the keypoint of the shoulder and the hand. The hand's keypoint is a better choice than the wrist's, because the handle is supposed to be located in the palm of the hand. Between the hand and the wrist there is a difference of approximately 5 to 8 cm, which greatly influences the obtained result. As there are errors in detection and measurement, in cases where the result of the distance between the shoulder and the wrist is less than the real one, this would result in a radius value quite different from the intended one, causing the movement to be carried out with the arm quite bent.

3. Angle of trajectory

   The angles β and α, corresponding to the amplitude of the movements made by the therapist and the patient, respectively, are obtained and an increment, i, is calculated using Equation 4.3. Figure 4.3 presents the graph of the normalised increment, i', illustrating its variation for any given angle. The angle, which is defined as the maximum angle of movement that the robot will make, is $\alpha + i$.

$$i = \alpha \cdot \left(1 - \frac{\alpha}{\beta}\right) \cdot 0.5 \tag{4.3}$$

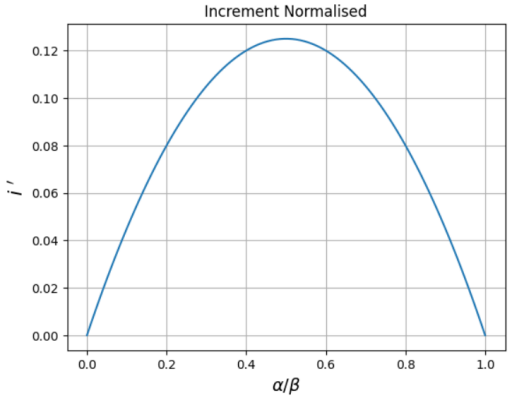


Figure 4.3: Graphical representation of the increment normalised, i'.

This type of increment was thought because there are certain limitations. The maximum angle obtained could not be higher than the angle that the therapist defines as the goal, which means that whatever amplitude the patient can achieve on their own, the final movement must tend towards the imposed angle. In addition, an exaggerated variation was not

intended. With this type of increment, there is a maximum variation of 12.5%, which in a 180º range is equivalent to 22.5º. Regarding the shape of the graph, the assumption was as follows: if a person is further away from reaching the goal, that is, if they have more difficulty moving, the increment should be smaller than for a person who already has more movement capacity, since this will be a person who has been more affected by the stroke.

It should be emphasised that the position that the therapist defines as the goal should be the one that, in their experience, the patient should reach at the end of all the sessions, so in the ideal case, this position is fixed for each patient. Although it can be changed if the therapist sees fit, if it remains fixed, this type of increment will promote the patient's evolution.

4.6 Extrapolation of the project

To extend the project's applicability, firstly the system was intended to be tested with two different movements: abduction/adduction and flexion/extension. However, as illustrated in Figure 4.4, the UR5 manipulator, as it was positioned, turned out to be insufficiently long to perform the flexion/extension movement. Having said that, if the base of the robot was closer to the side of the bench, it might be possible to perform other movements.

Figure 4.4: Attempt arm flexion/extension movement with UR5.

Despite the physical limitations of this manipulator, it is possible to extrapolate the design to various ranges of movement. This evaluation is essential for assessing the system's potential for various therapeutic movements. The most challenging scenario analysed was when the patient was facing the camera and performing shoulder flexion/extension, as it was not possible to detect the keypoints in this position. A potential solution would be for the person to turn slightly to the side being analysed so that detection is possible.

For more complex movements, such as raising the arm at a 45° angle from the parallel plane, a more sophisticated analysis is necessary. This is because the movement occurs in a plane where all coordinates — x, y, and z — vary. It is crucial to note that this generalisation primarily applies to movements within vertical planes. Movements in horizontal planes present additional challenges for Kinect detection due to the limb being oriented parallel to the camera.

Therefore, for the project to be generalised to any movement in the vertical plane, some additional considerations are necessary. First, it is essential to determine the plane in which the movement occurs. To achieve this, the angle between the patient's arm and the xz plane needs to be calculated.

Given the patient's arm vector $\mathbf{v} = (x, y, z)$, it is possible to project this vector onto the xy plane, obtaining $\mathbf{v_{proj}} = (x, y, 0)$. Then, it is possible to compute the angle, γ, between the vector $\mathbf{v_{proj}}$ and the xz plane using Equation 4.4, where $\left\|\mathbf{v_{proj}}\right\| = \sqrt{x^2 + y^2}$, represents the norm of the projected vector.

$$\gamma = arcos\left(\frac{x}{\left\|\mathbf{v_{proj}}\right\|}\right) \tag{4.4}$$

Knowing this angle, only a rotation matrix is needed. Since the plane of movement is rotated around the z axis, the corresponding rotation matrix for the angle γ is given by Equation 4.5.

$$\mathbf{R} = \begin{pmatrix} cos(\gamma) & 0 & sin(\gamma) \\ -sin(\gamma) & 0 & cos(\gamma) \\ 0 & 0 & 1 \end{pmatrix} \tag{4.5}$$

The movement in the xz plane is defined by the expressions presented below - Equations 4.6, 4.7 and 4.8.

$$x = c_x + R \cdot cos(\theta) \tag{4.6}$$

$$y = c_y \tag{4.7}$$

$$z = c_z + R \cdot sin(\theta) \tag{4.8}$$

Thus, by applying the rotation matrix to these expressions, the movement equations can be adapted to any plane. The transformed coordinates (x', y', z') can be computed by the Expression 4.9.

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \mathbf{R} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} c_x \cdot cos(\gamma) + Rcos(\theta)cos(\gamma) + c_z sin(\gamma) + Rsin(\theta)sin(\gamma) \\ -c_x \cdot sin(\gamma) - Rcos(\theta)sin(\gamma) + c_z cos(\gamma) + Rsin(\theta)cos(\gamma) \\ c_z + Rsin(\theta) \end{pmatrix} \tag{4.9}$$

Movements involving the hand are not included in this algorithm due to the lower precision associated with detecting hand movements. However, with an algorithm specifically designed for hand rehabilitation, such movements could also be incorporated.

To expand the system's applicability to various body movements, both upper and lower limb, it would be ideal to develop an interface allowing the therapist to select the desired movement. Based on this selection, the system would automatically identify the relevant keypoints and adjust the movement plane accordingly. Currently, this approach is feasible for simpler movements, such as those depicted in Figure 2.3.

Chapter 5

From Execution to Results

This chapter will explain the tests carried out and examine the results obtained from detecting keypoints on the upper limbs of three individuals with different structures, ages and levels of mobility, using the Kinect SDK. It will also address the impact of factors such as background, the presence of multiple people in the camera's field of view, and partial obstruction of the skeleton.

The tests made with the UR5 robotic arm will also be discussed and the handle for it will be presented and discussed in detail. In terms of integration, the chapter will evaluate the accuracy of the transformation matrix used to convert coordinates from the Kinect camera reference to the UR5 manipulator reference. The final results will be discussed by comparing the target angle for positioning the patient's arm with the angle measured by the Kinect at that moment. Finally, a SWOT analysis of the project will be presented.

The entire project was carried out on a Windows 10 computer equipped with an Intel(R) Core(TM) i7-8565U CPU @ 1.80 GHz, 16.0 GB of RAM and a NVIDIA GeoForce GTX 1050 with Max-Q Design graphics card.

5.1 Detection of keypoints

In order to analyse the robustness of the detection, using the Kinect SDK, various tests were carried out. Qualitatively, the goal was to understand how the methods reacted when there were two or more people in the camera's view, when the person was partially obstructed by other people or objects, and when there were variations in background. To analyse the accuracy, quantitative tests were conducted. Real distances between keypoints were compared with those calculated from real time coordinates.

The base code of the Kinect SDK enabled the detection and display of human skeleton joint keypoints in the interface mentioned above. However, modifications were made to focus on detecting and storing only the desired joints. Firstly, the displayed keypoints were restricted to the joints of the upper limbs and the base of the hip. The base of the hip was considered only for the purpose of better visual localisation of the body. The upper limb joints considered are illustrated in Figure 5.1.

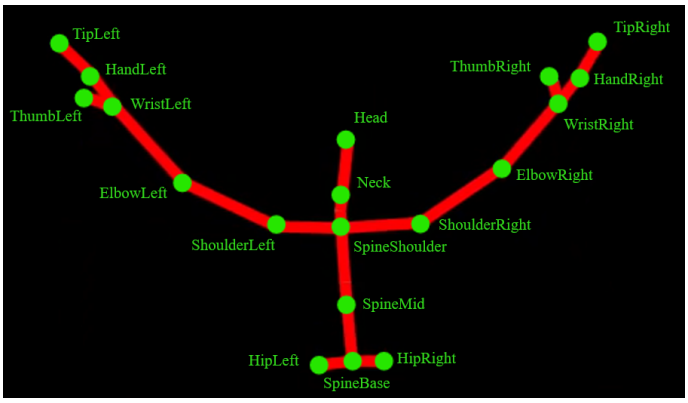
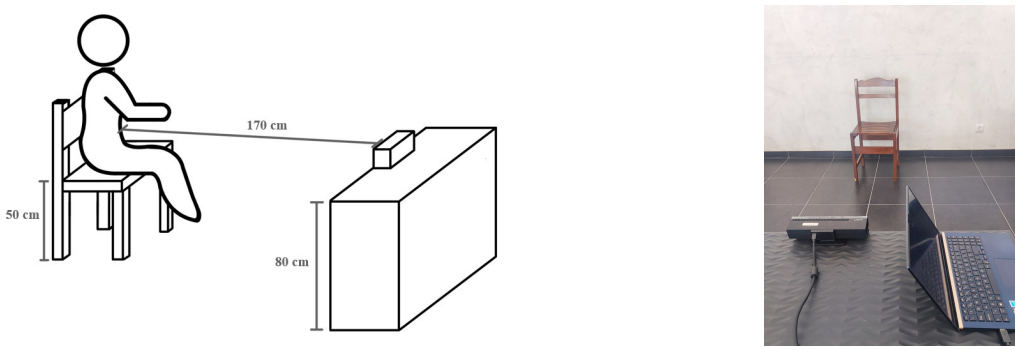


Figure 5.1: Representation of upper joints.

As the goal is to detect only one person, a function was added to ensure that only a single individual is tracked in the camera plane. The objective is for the same person to remain tracked, even if another individual enters the frame. The system maintains the detection on the first person detected and only changes the tracking if the original one leaves the plane. This is possible because the Kinect SDK has a parameter called *TrackingID*. Each detected body is assigned a unique *TrackingID*, allowing the system to focus on tracking only the body with the *TrackingID* of the first person detected.

5.1.1 Quality analysis

In order to assess detection accuracy, the real distances between certain keypoints on a person were measured and then compared with the Euclidean distances calculated from the coordinates obtained through real time human skeleton detection. To have a sample considered valid, 20 samples were taken with three individuals when they were sitting framed in the plane and at a distance of 1.70 m from the camera, as illustrated in Figure 5.2a and in the real image of the setup made in Figure 5.2b. Of the 20 samples, ten were with the limbs at 90º to the camera at different angles to the torso - Figure 5.3a; five were with the arms towards the camera but slightly open - Figure 5.3b; and five with the arms in the opposite direction to the camera - Figure 5.3c.



(a) (b)

Figure 5.2: Camera positioning: (a) Illustrated positioning of camera; (b) Real setup made.

(a) (b) (c)

Figure 5.3: Arm positioning: (a) Parallel to the camera; (b) Towards the camera; (c) Backwards the camera.

The tests carried out, both towards or away from the camera, aimed to assess the accuracy of depth measurement and to determine whether movements outside the plane parallel to the camera (equivalent to the coronal plane when the person is facing it) could be considered. The subjects chosen for these tests were a 23 year old woman, a 15 year old boy, and a 93 year old woman. This selection was made to include individuals of different statures and mobility levels. Although mobility does not affect the coordinate capture, as the project is intended for people with reduced mobility, it is crucial to understand if any factors need to be considered when analysing the data.

In Figure 5.4a is shown one representation of the detection result for movement in the parallel plane to the camera, Figure 5.4b presents the detection when the movement is done towards the camera and Figure 5.4c shows the detection result when the movement is done opposite to the camera.

(a) (b) (c)

Figure 5.4: Representation of the detection result when the movement is done in different directions: (a) Parallel to the camera; (b) Towards to the camera; (c) Opposite to the camera.

As shown, the detection appears to be satisfactory in all directions. It can also be concluded that hair does not interfere with the detection process. Moreover, since the algorithm is designed to detect fingertips, the thumb, and the palm even when the hand is closed, as seen in the last figure, these points are estimated and positioned according to where they would likely be.

As previously mentioned, the coordinates of the skeleton of the three individuals were recorded with the upper limbs in different positions. Subsequently, distances between specific keypoints were calculated: wrist to elbow, wrist to shoulder, elbow to shoulder, right shoulder to left shoulder, left hip to right hip, and centre of hip to centre of shoulder. Arm measurements were taken on both the left and right sides, with real values determined using a tape measure.

As illustrated in Figure 5.5, when measuring the real values, attention had to be paid to the fact that the coordinates of the keypoints displayed by the algorithm are not the values of the extremities of the limbs. For example, the distance between shoulder keypoints does not correspond to the distance between the ends of the limbs, which would result in much shorter measurements. To ensure that real measurements could be more accurately compared with those obtained from the algorithm, keypoint markers were placed on the individuals. This allowed for a closer alignment between the actual measurements and the data provided by the algorithm.

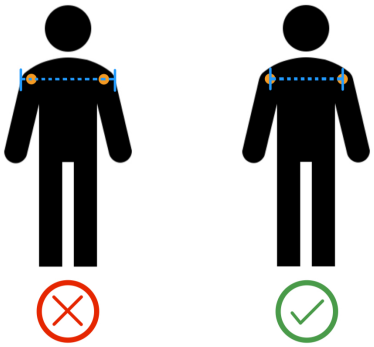


Figure 5.5: Correct real measurement of distances between keypoints.

Table 5.1 displays the average values obtained with the limbs parallel to the camera, facing the camera, and opposite to the camera for the three individuals tested. Additionally, the table includes the average of all obtained values and the real measured values. It is important to note that for the older individual, measurements with the limbs oriented away from the camera could not be obtained due to restricted mobility.

Table 5.1: Results obtained with Kinect SDK in the plane parallel, towards and opposite to the camera, and the real measurements. The symbol (-) indicates that the test could not be performed.

Individual	Measure		Distance				
			Parallel (cm)	Towards (cm)	Opposite (cm)	Mean (cm)	Real Measurement (cm)
Individual 1	Left	Shoulder - Wrist	50.5	50.7	50.3	50.5	50.0
		Shoulder - Elbow	26.3	26.9	24.2	25.9	25.0
		Elbow - Wrist	25.0	23.9	24.4	24.6	25.0
	Right	Shoulder - Wrist	49.8	50.2	50.6	50.1	50.0
		Shoulder - Elbow	25.2	24.4	27.4	25.6	26.0
		Elbow - Wrist	24.8	26.0	23.4	24.7	25.0
	Spine Base - Spine Shoulder		41.5	42.0	41.7	41.7	40.0
Individual 2	Left	Shoulder - Wrist	46.4	46.0	51.2	47.5	46.5
		Shoulder - Elbow	24.4	23.6	25.2	24.4	24.0
		Elbow - Wrist	22.5	23.0	25.9	23.4	23.0
	Right	Shoulder - Wrist	47.1	48.9	50.5	48.4	48.0
		Shoulder - Elbow	24.1	24.9	24.7	24.4	24.5
		Elbow - Wrist	23.3	24.4	26.0	24.2	24.0
	Spine Base - Spine Shoulder		41.1	37.8	39.0	40.5	40.0
Individual 3	Left	Shoulder - Wrist	42.1	40.7	-	41.7	43.0
		Shoulder - Elbow	22.3	22.1	-	22.3	23.0
		Elbow - Wrist	19.7	19.9	-	19.8	20.0
	Right	Shoulder - Wrist	39.9	39.9	-	39.9	42.0
		Shoulder - Elbow	21.6	22.9	-	22.0	22.0
		Elbow - Wrist	19.8	19.1	-	19.6	20.0
	Spine Base - Spine Shoulder		38.3	39.0	-	38.5	40.0

At first, the distances obtained show some discrepancies compared to the real values, with differences of a few centimetres. This is expected given that keypoint detection is not perfectly stable. The analysis carried out was a descriptive analysis, using parameters such as the mean, standard deviation and percentage error to understand how the data was presented and to draw conclusions about its accuracy and overall quality in real time. However, it is also important to consider the maximum absolute deviation obtained from the tests. This measure is crucial because it is necessary to guarantee that in any case the error will never be large enough to influence the rest of the process when integration is conducted. These results, presented in Table 5.2, reveals that the highest error is of 5% for individual 3 in the distance between the shoulder and right wrist, which will be examined in more detail later.

Table 5.2: Comparison between the real distance between keypoints and the distance obtained from Kinect SDK detection, including standard deviation, maximum absolute deviation and error.

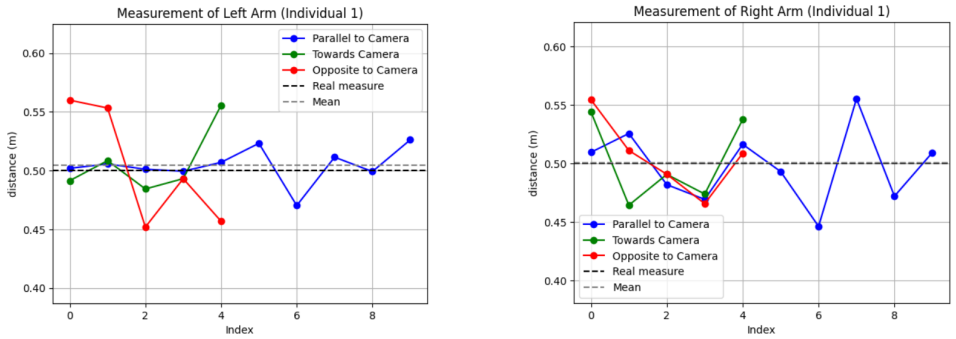
Individual	Measure		Standard deviation (cm)	Maximum absolute deviation (cm)	Error
Individual 1	Left	Shoulder - Wrist	2.9	6.0	0.9%
		Shoulder - Elbow	2.7	5.5	3.7%
		Elbow - Wrist	2.7	6.1	1.8%
	Right	Shoulder - Wrist	3.2	5.5	0.2%
		Shoulder - Elbow	2.6	4.6	1.7%
		Elbow - Wrist	2.5	4.2	1.0%
	Spine Base - Spine Shoulder		1.6	3.4	1.6%
Individual 2	Left	Shoulder - Wrist	3.7	**8.5**	2.1%
		Shoulder - Elbow	2.1	3.9	1.6%
		Elbow - Wrist	2.6	4.5	1.9%
	Right	Shoulder - Wrist	2.8	5.0	0.8%
		Shoulder - Elbow	1.6	3.7	1.8%
		Elbow - Wrist	2.3	4.2	1.0%
	Spine Base - Spine Shoulder		**4.6**	7.8	1.1%
Individual 3	Left	Shoulder - Wrist	1.5	4.4	3.1%
		Shoulder - Elbow	2.1	6.0	3.2%
		Elbow - Wrist	1.4	3.6	1.0 %
	Right	Shoulder - Wrist	2.4	7.9	**5.0** %
		Shoulder - Elbow	2.3	5.3	0.1%
		Elbow - Wrist	1.4	2.3	2.0%
	Spine Base - Spine Shoulder		3.1	7.2	3.7%

It can be observed that the largest deviation is 8.5 cm. While this is somewhat expected due to the occasional instability in detection, such a value might be too high for practical integration, potentially affecting the overall rehabilitation process. The analysis indicates that the standard deviation and percentage error suggest the results are generally precise and accurate, with relative errors below 5.0% and standard deviations mostly around 2.0 and 3.0 cm, though there is an exception with a high value of 4.6 cm. Considering the maximum absolute deviation, it is evident that there may be instances where the obtained values are less accurate. The impact of this variability will be discussed further below.

The influence of the measurement plane (parallel or not to the camera) on the distances was analysed. In this analysis, graphs of the calculated distances were generated. Each graph dis-

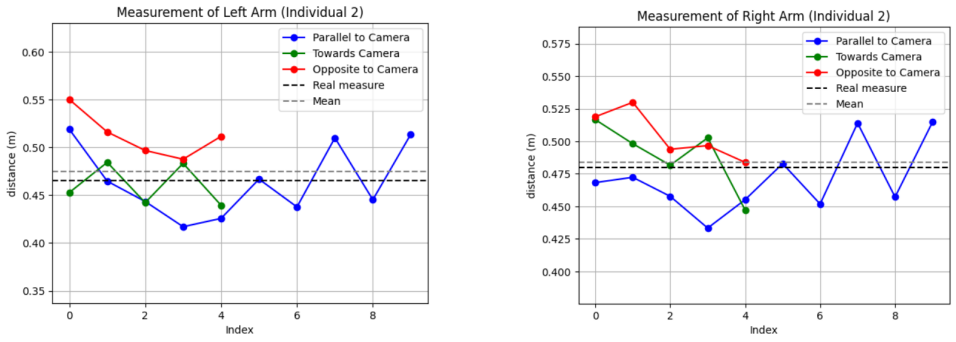
plays a point for every calculated distance, clearly distinguishing between the different types of movements, where blue points represent distances obtained when the movement was performed in a parallel plane, the green points indicate movements made towards camera and the red points denote movements made opposite to camera. Additionally, the graphs feature two dashed lines: a grey dashed line represents the average of all the calculated distances; a black dashed line indicates the real distance. These visual elements help to compare the calculated distances with the actual measured distance and assess the consistency and accuracy of the measurements across different movements.

As shown in the graphs, of the left and right arm distances for individual 1 (Figures 5.6a and 5.6b, respectively) and individual 2 (Figures 5.7a and 5.7b, respectively), there is no clear tendency in these measurements, with values both above and below the actual value in all planes. This pattern is consistent across all cases.



(a) (b)

Figure 5.6: Distances between the wrist and the shoulder of individual 1: (a) Left arm; (b) Right arm.



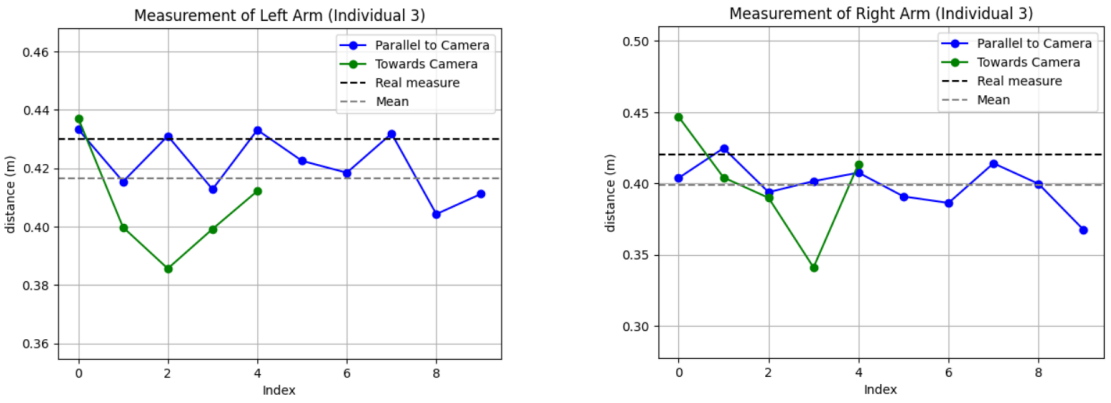
(a) (b)

Figure 5.7: Distances between the wrist and the shoulder of individual 2: (a) Left arm; (b) Right arm.

5.1.2 Particularity of the individual 3

In Figure 5.8, graphs illustrate the distances measured between the shoulder and wrist of the left (Figure 5.8a) and the right side (Figure 5.8b) for person 3 in all positions, representing the cases

with the greatest error. It can be concluded that the distances measured between the wrist and shoulder for the older individual are shorter than the actual measurements, with a more pronounced discrepancy on the right side. This variation is attributed to the individual performing all movements with her arm slightly tucked in.



(a) (b)

Figure 5.8: Distances between the wrist and the shoulder of individual 3: (a) Left arm. (b) Right arm

As shown in Figure 5.9, which displays the detection of keypoints with the relevant points highlighted in red (wrist, elbow, and shoulder, both right and left), the individual was unable to fully extend her arm despite the effort. Since the distance between keypoints is calculated as a vector, the computed value is likely lower than the actual distance, leading to a higher error. This discrepancy is evident in the graphs, particularly in Figure 5.9, which clearly shows that the right arm is more retracted compared to the left. Indeed, the distance calculated for the left arm in Figure 5.8b is closer to the actual distance than that for the right arm, as shown in Figure 5.8a.

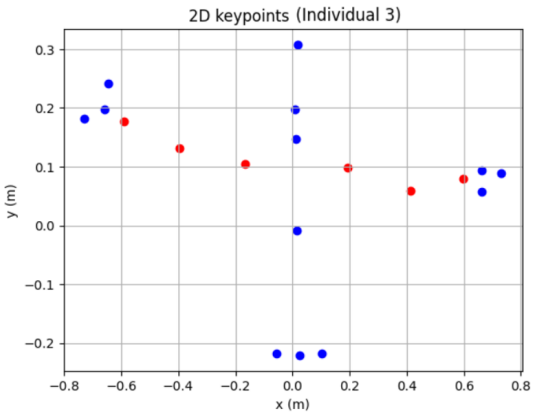
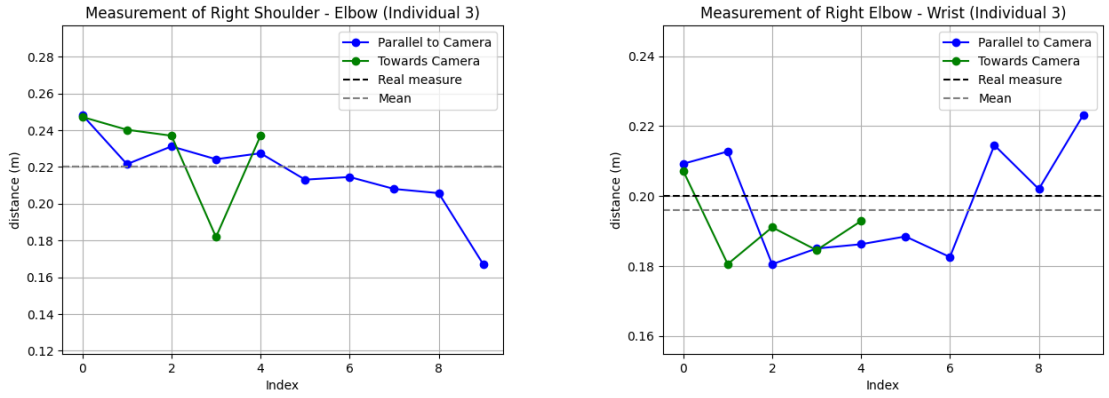


Figure 5.9: 2D keypoints of an exercise done by the individual 3, with the wrist, elbow, and shoulder (both right and left) highlighted in red.

In Figure 5.10 are presented the calculated distances between the right shoulder and elbow (Figure 5.10a) and the distances between the right elbow and wrist (Figure 5.10b). It is evident that these distances are closer to the real values compared to the previously observed measurements. When the averages of these two are summed, a value of 41.6 cm is obtained, resulting in an error of 0.8% compared to the real value of 42.0 cm, in contrast to the error of 5.0% obtained from the average shoulder-wrist distance of 39.9 cm.



(a) (b)

Figure 5.10: Distances of right side of person 3: (a) Distance shoulder-elbow; (b) Distance elbow-wrist.

Only the graphs for the right side are shown, as this side exhibited the largest discrepancy between the real and measured values. However, a similar phenomenon was observed on the left side, though the difference was smaller, with an error of 2.2% when summing the two distances, compared to an error of 3.1% with direct measurements.

The data for individuals 1 and 2 were more stable, with percentage errors in most cases ranging between 1.0% and 2.0%. This is probably due to their younger age, which allowed them to maintain the requested position more steadily with their arms more fully extended. However, all obtained values were generally favourable since, even for individual 3, the errors were consistently less than or equal to 5.0%, which is considered acceptable.

Errors and deviations from the real value are to be expected since any algorithm will always have some uncertainty associated with it. To improve accuracy, one could use a pose detection algorithm that has already been trained on a large dataset and then fine-tune it on a dataset specific to rehabilitation images. This would make the algorithm more tailored to the environment in which it will be used, making it more robust and reliable.

5.1.3 Qualitative tests

Changing the background, from a white wall to various environments, as can be seen in Figure 5.11, did not impact the detection. In all cases the detection was done correctly. This is important for integration purposes, as it indicates that placing the UR5 behind the person will not interfere with the detection process.

(a) (b) (c)

Figure 5.11: Impact of background on detection: (a) White wall; (b) Darker environment; (c) Mixed background.

As for the presence of multiple people within the camera's field of view, the implemented method proved highly effective, with no interference or disruption in detection, as seen in Figure 5.12a, except if the person is obstructed, as shown in Figure 5.12b. In such cases, the detection is affected by the obstruction itself, not by the presence of another person. When another individual passed directly in front of the detected person, completely obscuring them, the detection may either remain on the original person or shift, depending on the speed of the interruption.

(a) (b)

Figure 5.12: Impact of additional individuals in the camera's field of view: (a) Person positioned behind the detected individual; (b) Person positioned in front of the detected individual.

This behaviour is consistent with the occlusion of the subject: minor interferences, such as small objects, as presented in Figure 5.13a, do not significantly affect detection. However, if an arm, for instance, is fully occluded, as shown in Figure 5.13b, the system fails when attempting to predict its position. The more the limb is obscured, the less accurate the detection becomes, which is expected and predictable.

(a) (b)

Figure 5.13: Effect of obstruction on detection: (a) Small object interference with minimal impact; (b) Full occlusion of an arm leading to decreased accuracy in detection.

5.2 UR5 manipulator

According to Universal Robots, the current recommended setup for using the UR5 manipulator involves Ubuntu 20.04 with ROS Noetic (Universal Robots A/S (2020)). Since a Windows computer was needed to run the Kinect SDK, a connection was established through the Windows Subsystem for Linux (WSL) with Ubuntu 20.04. WSL version 1 was chosen because it allows direct access to the computer's hardware, such as Ethernet ports, which is necessary for communicating with the UR5 robot.

The code for sending commands to the UR5 was written in C++. To begin, a single command, using *moveL*, was tested to send the tool of the manipulator to specific coordinates, to also understand how it reacts to the defined speed and acceleration. Next, an attempt was made to have the tip of the UR5 follow a specific trajectory, a circumference.

To define the movements, the assembly of the UR5 on the bench had to be taken into account. Since the origin of the reference point is at the base of the robot, which is placed on the plane of the bench, the axes were aligned with it, as shown in Figure 5.14, with the *xx* axis parallel to the edge of the bench in a right-left direction, the *yy* axis at 90° to the same reference in a back-front direction and the *zz* axis pointing upwards.

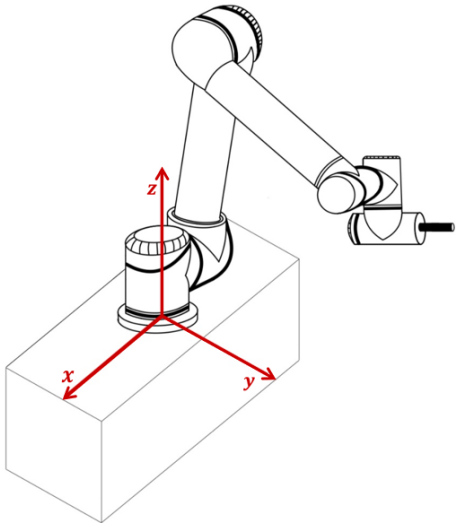


Figure 5.14: Spatial positioning of UR5 relative to the bench.

For the tests, the chosen plane for the movement was the *xz* plane. Consequently, the value of the *y* coordinate was kept constant (Equation 5.2) while the *x* and *z* coordinates varied according to the Equations 5.1 and 5.3, respectively.

$$x = c_x + R \cdot cos\theta \tag{5.1}$$

$$y = c_y \tag{5.2}$$

$$z = c_z + R \cdot sin\theta \tag{5.3}$$

Having made a complete circumference, the angle of movement was restricted, as the aim was to simulate the movement that will be made by the arm. So the circular movement was restricted between the angle of $\frac{3\pi}{2}$ rad and $\frac{3\pi}{4}$ rad. The choice of these angles was arbitrary, which was considered reasonable, since the ultimate aim was to vary the movement between the angle of $\frac{3\pi}{2}$ rad and the angle $\alpha + i$ rad mentioned in the Section 4.2.

5.3 Handle for UR5

The handle was designed taking into account the dimensions presented in the official Universal Robots document (Universal Robots (2023b)). The drawing of the part is shown below, in Figure 5.15 with the units being millimetres. The aim was for it to be secure and stable when attached to the robotic system. As the movement is curved and the handle will not rotate during the movement, it is the patient's hand that will rotate on the handle, so it must be smooth and shaped to allow this rotation. For this reason, the handle was designed in the shape of a cylinder.

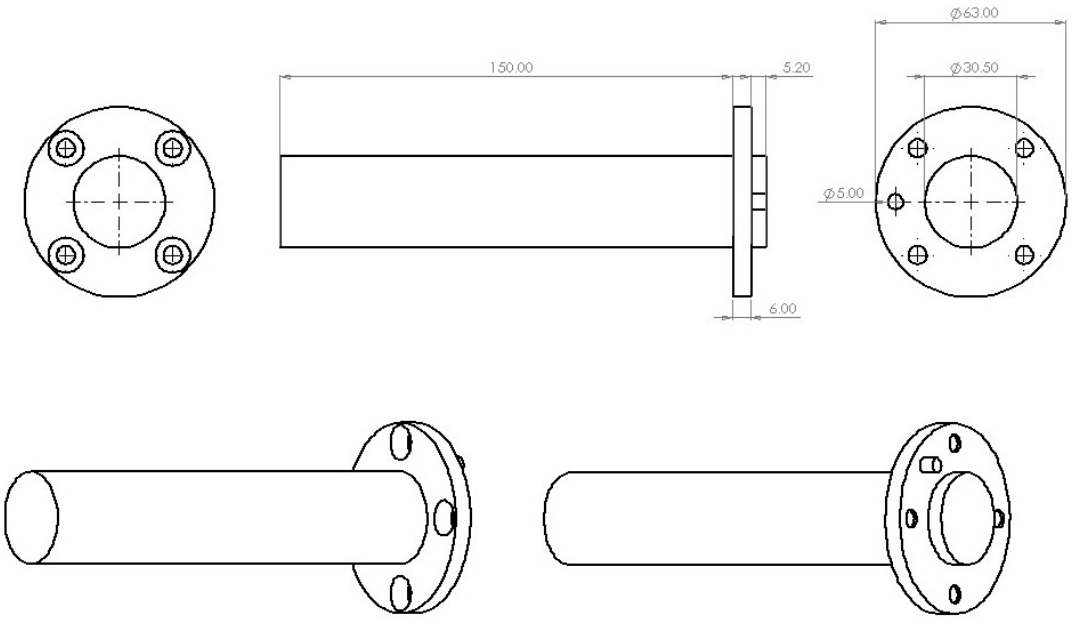


Figure 5.15: Drawing of the handle created in *SolidWorks*, with units in millimetres.

The length of 15 cm was chosen for the cylinder because it was considered that the handle should be comfortable, not just the size of the patient's hand. However, the most important factor is the uncertainty associated with determining the position in which the handle should be, something that will be explained when the integration is carried out.

With regard to the thickness of the piece to be gripped by the patient, it was taken into account that a cylinder with a length of 15 cm and a small radius would be fragile and could break. Although the aim is not to apply force to the handle, there will always be some pressure associated with the movement, since the patient's arm is in dead weight. However, the radius must also not be too large for it to be comfortable in the hand and the screws must be able to be tightened. The

thickness chosen was 30.50 mm which is the diameter of a cavity in the UR5 tool, so that the screws could be tightened easily and it was large enough not to break.

The part was 3D printed with a Raise3D Pro2 Plus (work volume of 305×305×605 mm) using Polyethylene Terephthalate Glycol-modified (PETG) filament. This additive manufacturing process creates objects layer by layer and it had to be taken into account that there is always an uncertainty, so 1 mm was removed in all directions (diameter and thickness) from the surfaces that would fit into the UR5 tip. The screws used were M6 and cavities were designed for the screw head itself, leaving a flat surface.

After 3D printing, the final piece was attached to the manipulator mounting plate with the screws, as shown in the Figure 5.16. The handle was very firm and no alterations were necessary. This handle does have one drawback: it lacks a fixed position for the hand grip. While the intention is to allow some flexibility due to the uncertainty of the detection algorithm, the handle can be grabbed in various ways. The extremes of this adjustment are illustrated in Figures 5.17a and 5.17b. It is important for the therapist to demonstrate the correct grip to the patient, as shown in Figure 5.17c, where the wrist is positioned straight.

Figure 5.16: 3D printed handle attached to the UR5.

(a) (b) (c)

Figure 5.17: Positions for holding the handle: (a) Wrong position with the wrist flexed; (b) Wrong position with the wrist tucked in; (c) Right position, with the wrist straight.

5.4 Integration

As observed with the older individual, during the tests, she was unable to fully extend her arm. Consequently, the calculated length of the arm, based on the distance between the shoulder and wrist keypoints, is shorter than the actual arm length. The value sent to the robotic system should not simply be the sum of these distances, as this does not reflect the person's natural position. Therefore, if the goal is to improve the person's mobility in the abduction/adduction movement, the exercise should begin with the arm positioned comfortably for them. Similarly, if the objective is to enhance the ability to fully extend the arm, the same design approach can be applied, focusing on elbow extension/flexion movements to facilitate this improvement. In summary, the movement should always be performed with the arm in a position that is comfortable for the patient.

The integration of all components was done with the aim of having a system that works in real time and is suitable and easy to use during therapy. When it came time to integrate the system, it became clear that while it was possible to operate using WSL, these connections introduced complexity, requiring data exchange between Windows and WSL, and between WSL and the UR5. This added complexity and made the system less user friendly, which contradicts the project's goal of creating an intuitive interface. As a result, efforts were made to connect the UR5 robot directly via Windows. While WSL provided a workaround, it introduced layers of communication that complicated the system.

Since the implementation on the robot is straightforward, involving only the transmission of an array with the information to move and considering that the robot's buffering system stores and executes commands in the order they are received, a more direct communication approach was feasible. As URScript can be used with the C# language, the code was directly translated into it, and communication with the robot was accomplished using TCP, without using ROS. This direct approach leverages the robot's ability to handle and execute commands sequentially, simplifying the integration process.

As represented in Figure 5.18, in the setup, the Kinect camera should be centred with the UR5 to ensure consistent body detection, and positioned approximately 2.5 m from the robot's base, as there must be sufficient space for the person to sit between the two elements. The patient must face the camera with the UR5 robotic arm positioned behind them. Due to the limited reach of the robotic arm, some aspects need to be taken into consideration when performing with it. In particular, the height of the seat the patient is in. In cases where the patient's arm is longer, if the seat is lower, in order to get the arm to the starting position, it could hit the bench. If it is higher, they may not be able to bring their arm to angles close to 180º.

In this case, the seat height was set at 50 cm because the person who conducted the tests had an arm length of 58 cm, and this height allowed to perform a 180° movement, with the robot's range being pushed to its limits in extreme positions; however, this value will depend on the height of the bench on which the manipulator is located. In addition, the patient should always be positioned approximately 30 cm from the bench, to ensure that the movement, which is made in the xz plane with the y coordinate constant, has the y value already outside the bench.

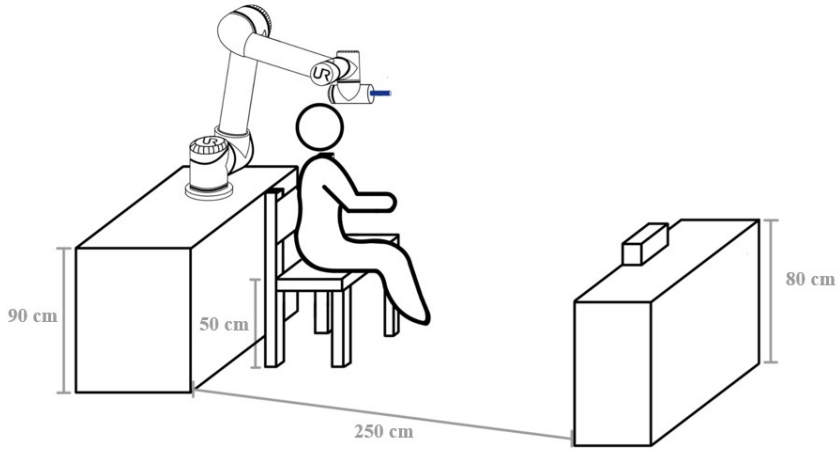


Figure 5.18: Setup representation of the position between the UR5, the Kinect camera and patient.

Moreover, when analysing the left side, the patients should sit with the entire manipulator on their left side, as shown in Figure 5.19a. If the side to be analysed is the right, then the manipulator should be on the right, as presented in Figure 5.19b, otherwise it may not be able to follow the entire movement due to its length. However, the distance to the robot was not taken into account in an accurate way, partly because it is difficult to determine precisely and because, by ensuring that it is on the right side, it was always able to follow the movement. This factor was highlighted by the fact that people tend to centre themselves with the camera and the manipulator, something that will not happen in this case.

(a) (b)

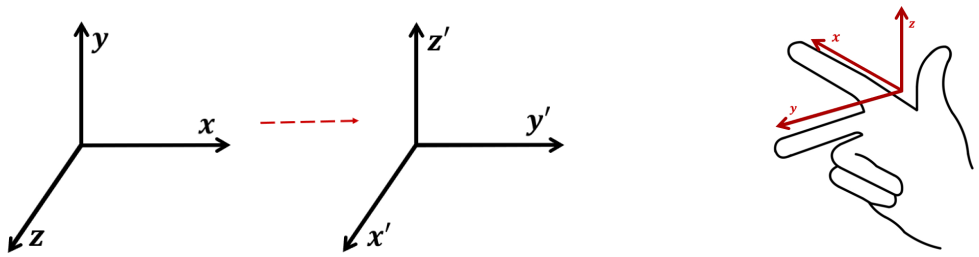
Figure 5.19: Position in relation to the robot: (a) When the right side is being analysed; (b) When the left side is being analysed.

5.4.1 Variables to define the trajectory

Regarding the variables to be calculated to define the trajectory to be done by UR5 manipulator, these include the centre of movement - 3D coordinates of the shoulder keypoint; the radius - distance between the shoulder and hand; and the angle of movement. All these values depend on the

accuracy of the Kinect camera, which, as previously discussed, can deviate by a few centimetres.

The shoulder coordinates must be obtained in the UR5 reference frame, as mentioned in the previous chapter. Therefore, it is necessary to calculate the transformation matrix from the Kinect reference frame to the manipulator reference frame, $\mathbf{H}_{\mathbf{K}}^{\mathbf{M}}$, composed of a rotation matrix $\mathbf{R}_{\mathbf{K}}^{\mathbf{M}}$ and a displacement vector $\mathbf{d}_{\mathbf{K}}^{\mathbf{M}}$. Considering the setup shown in Figure 5.18, the axis names of the Kinect can be changed to simplify this matrix as much as possible, as shown in Figure 5.20a, always taking into account the right hand rule, shown in Figure 5.20b.



(a) (b)

Figure 5.20: Illustration of coordinate system adjustments: (a) Modification of Kinect axes name to simplify the transformation matrix, $\mathbf{H}_{\mathbf{K}}^{\mathbf{M}}$; (b) Right hand rule for axis orientation.

Considering these changes, the relative position between the axes of the Kinect camera and the UR5 manipulator is shown in Figure 5.21. With this representation, the rotation matrix, illustrated in Equation 4.2, was easily determined, with its final form presented in Equation 5.4, and the vector $\mathbf{d}_{\mathbf{K}}^{\mathbf{M}} = \begin{pmatrix} x_K^M & y_K^M & z_K^M \end{pmatrix}$, with values illustrated in Figure 5.21.

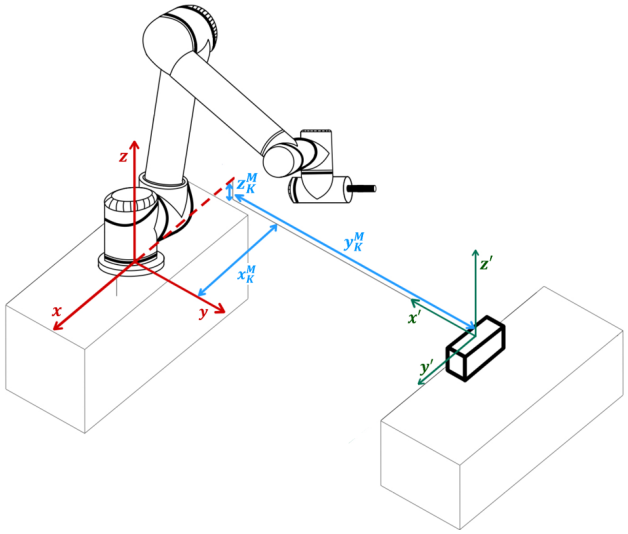


Figure 5.21: Spatial positioning of UR5 relative to the Kinect.

$$\mathbf{R}_{\mathbf{K}}^{\mathbf{M}} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5.4}$$

Regarding the $\mathbf{d_K^M}$ vector, it was tried to centre the Kinect so that all positions can be detected. Then, the relative position of the Kinect to the manipulator would be measured; however, this measurement would not be realistic for several reasons. Firstly because it is difficult to determine precisely where the centre of each reference point is and secondly because such measurements are not made on any surface, which always leads to higher errors.

Despite these limitations, measurements were made using a tape measure to estimate the parameters of the $\mathbf{d_K^M}$ vector. Afterward, the tool of the manipulator was positioned at the patient's shoulder, and its coordinates were read from the UR5's screen. Following this, the shoulder was detected using the Kinect, and the transformation matrix adjusted until the values displayed on the UR5 screen match those obtained through the matrix.

Once the displacement vector, $\mathbf{d_K^M}$, was determined through trial and error for a given point, additional tests were conducted by varying the shoulder's position. These tests aim to verify if the correct centre values were obtained while keeping the Kinect's position and the transformation matrix constant. The tool of the manipulator was then aligned with the shoulder's position so that its coordinates matched those obtained via the transformation matrix.

It is important to note that the coordinates detected by the Kinect are relative to the front of the shoulder, so the tool was also positioned accordingly. Although the coordinate values of UR5 are fairly precise, the alignment of the tool relative to the shoulder is approximate. Despite a specific uncertainty value is not assigned, it is estimated to be on the order of centimetres in all directions, including the *yy* direction, due to the lack of guarantee that the tool is positioned as close to the shoulder as required.

Several tests were conducted and, for each tool position and alignment with the shoulder, the shoulder's position was determined five times using the transformation matrix, to reduce the inherent errors in Kinect measurements. The Table 5.3 presents the average values obtained and the corresponding error. The first row of the table lists the values used to adjust the transformation matrix. As the quality analysis has been completed, only the maximum absolute deviation is presented, as this measure is most relevant for the functioning of the integrated system.

Table 5.3: Comparison between shoulder coordinates obtained from the UR5 screen and Kinect.

Test Number	Coordinate	UR5' screen (m)	Mean (m)	Maximum absolute deviation (m)
0	x	-0.030	-0.03	0.01
	y	0.682	0.69	0.03
	z	0.141	0.12	0.02
1	x	-0.232	-0.20	0.04
	y	0.690	0.69	0.02
	z	0.141	0.15	0.01
2	x	-0.340	-0.33	0.03
	y	0.510	0.52	0.02
	z	0.141	0.12	0.04
3	x	-0.100	-0.08	0.04
	y	0.551	0.58	**0.05**
	z	0.187	0.18	0.03

The tests conducted revealed that the obtained values were fairly consistent, with a maximum deviation of 5 cm. This level of deviation aligns with expectations based on the quality analysis

results. Thus, the method demonstrated its viability for use in this project. Additionally, these tests reaffirm the detection capabilities and accuracy of the Kinect camera and its SDK. Consequently, the matrix utilised in these tests, which was maintained constant throughout the remaining experiments due to the unchanged setup, is as in Expression 5.5.

$$\mathbf{H}_{\mathbf{K}}^{\mathbf{M}} = \begin{pmatrix} 0 & 1 & 0 & -0.10 \\ -1 & 0 & 0 & 2.34 \\ 0 & 0 & 1 & 0.02 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.5}$$

It is important to note that the values shown in the matrix should not be overly emphasised, as each setup will require adjustment of this matrix. However, confidence in determining the matrix has been established. For practical therapy implementation, the setup should remain fixed to ensure readiness for patient use. Recalibrating the matrix for each therapy session is neither practical nor feasible. Therefore, the system should be installed and calibrated in a manner that remains constant, by fixing the UR5 and the Kinect v2 position. In this context, all values can be determined automatically without requiring additional measurements by the therapist. Reviewing each of these values:

- The shoulder coordinates are derived by applying the transformation matrix to the keypoint coordinates obtained by the Kinect.
- The radius, representing the arm's length, is calculated using the Euclidean distance between the shoulder and hand keypoints.
- The angle of movement is determined by adding the increment from Equation 4.3 to the angle between the shoulder - hand vector and the middle shoulder - middle base vector in the maximum position reachable by the patient.

Once the variables had been defined, the code was designed to make the UR5 move and the side to be analysed also had to be taken into account. Considering the UR5's reference point, if the left side is being analysed, the angle will decrease, as it is intended to move clockwise. If the intention is to analyse the right side, the angle will increase so that the movement is anti-clockwise.

These calculations are dependent on the patient's position, physical structure and motor capacity. It is also important to emphasise that, as discussed earlier, inaccuracies in keypoint detection can adversely affect the rehabilitation process. A more noticeable example of this issue is the radius value; if there is a discrepancy of 8 cm, as observed in the analysis, this difference will be quite apparent during therapy. Such inconsistencies are unpredictable and may necessitate the patient repeating the movement analysis.

Given that the 3D coordinates of the shoulder keypoint determined by the Kinect already have an associated uncertainty, this value is even more affected by this change of the referential, and it is possible that this value differs from reality by a few centimetres. It is therefore important that the handle of this project is something that the patient can release, and the handle itself is 15 cm long,

allowing for a greater margin of error in determining the depth with the Kinect. In addition, 7 cm, which is approximately half the length of the handle, will be subtracted from the y coordinate of the centre calculated by the transformation matrix. This adjustment will align the handle with the person's shoulder. The handle's length of 15 cm provides a margin for mitigating any detection errors.

5.4.2 UR5 response

After calculating the variables and determining the trajectory according to Equations 5.1, 5.2, 5.3, it was observed that the manipulator ended the movement at an angle significantly different from the target angle. Inadequate selection of the parameters previously mentioned - velocity (v), acceleration (a), blend radius (r), and the number of points on the trajectory (num_traj), is likely the cause of the observed error.

It was determined that the high value of r was primarily responsible for the deviation in the final angle. Consequently, the radius of the approach curve was reduced and the number of trajectory points was increased. These adjustments, made through trial and error, resulted in a smoother trajectory with improved precision, more closely mimicking the natural movement of the human arm.

The final parameters — $a = 0.02$ ms^{-2}, $v = 0.25$ ms^{-1}, $r = 0.02$ m, and $num_traj = 50$ — demonstrated promising results for the manipulator, reducing the initial inaccuracy in the final angle. However, additional final tests are necessary to fully validate these adjustments.

The final tests involved positioning the patient's arm at the desired angle, $\alpha + i$, using the manipulator. The Kinect SDK was then employed to measure the angle of the patient's arm in this final position and compare it with the target angle, $\alpha + i$. This comparison assessed the accuracy of the manipulator's positioning. To account for the Kinect's inherent error, the angle measurement was repeated five times to minimise measurement error. The results are presented in Table 5.4. Non-absolute deviations were considered in the analysis because if the error is attributed to the r parameter, it is anticipated that the deviation will be negative. This means the angle achieved by the robot would be slightly less than the intended angle.

Table 5.4: Comparison between the target angle and the angle measured by Kinect.

Test Number	Reference amplitude	Amplitude achieved	Increment value	Amplitude of therapy	Angle by Kinect SDK	Signed deviation
0	90.8°	81.6°	4.2°	85.7°	85.1°	- 0.6°
1	102.8°	81.6°	8.4°	90.0°	90.8°	0.8°
2	120.6°	36.3°	12.7°	49.0°	48.1°	- 0.9°
3	120.6°	55.5°	15.0°	70.4°	68.9°	-1.5°
4	120.6°	74.4°	14.3°	88.6°	87.1°	- 1.5°
5	120.6°	102.8°	7.6°	110.4°	108.6°	- 1.8°
6	146.0°	70.8°	18.2°	89.0°	87.6°	- 1.3°
7	146.0°	82.7°	17.9°	100.6°	99.4°	- 1.3°

Based on the presented results it can be concluded that they were satisfactory, with the largest absolute deviation between the target amplitude and the Kinect measured amplitude being 1.8°. Notably, in only one instance was the deviation positive, indicating that in most cases the measured angle was less than the target angle. Although it is important to remember that the Kinect measurement includes some error, the results might be influenced by the r parameter. In practice, a therapist would find it very difficult to discern whether a patient's arm is more or less 2° than intended. Therefore, while the target angle is important, a deviation of 2° is not problematic. Additionally, reducing the parameter r made the movement less smooth, so even if the error primarily stems from this adjustment, as the impact of it on therapy is minimal, the parameter was maintained.

The initial trajectory angle was originally set to vary from $\frac{3\pi}{2}$ rad, as it is close to the arm's resting position alongside the body, making it a reasonable starting approximation. However, if the trajectory begins exactly at this angle, the manipulator will bring its tool over the patient's body, as shown in Figure 5.22a. To address this, the arm was placed at rest, and the angle in this position was measured five times using the Kinect. These measurements showed that, if the side being analysed is the left, the initial angle should be approximately $12 \cdot \frac{\pi}{180}$ rad smaller than the $\frac{3\pi}{2}$ rad previously considered, as presented in Figure 5.22b. If the side being analysed is the right, the initial angle would be $12 \cdot \frac{\pi}{180}$ rad greater than the $\frac{3\pi}{2}$.

(a) (b)

Figure 5.22: Initial angle position for the left side: (a) Initial position if the angle considered was $\frac{3\pi}{2}$ *rad*, equivalent to zero degrees from the body; (b) Initial position if the angle considered was $\left(\frac{3\pi}{2} - 12 \cdot \frac{\pi}{180}\right)$ *rad*, equivalent to twelve degrees from the body.

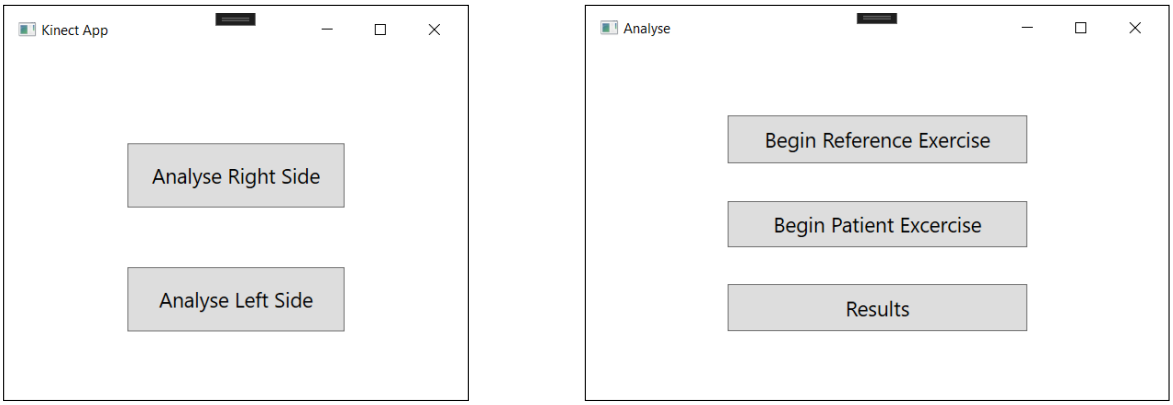
It can therefore be concluded that this method meets the expected criteria and achieves satisfactory results for the project. Despite minor deviations, this approach has the potential to be a valuable tool in the rehabilitation of individuals with motor difficulties.

5.4.3 Graphical user interface

To integrate the complete system and make it work in real time in a more user-friendly and interactive way, an interface was created using Windows Presentation Foundation (WPF), as it was already used in the camera's SDK. Each page of the interface is defined in an eXtensible Application Markup Language (XAML) file where it is possible to define buttons, interactive text boxes

or just information, graphics, among other things. Each of these pages has an associated code, written in *C#*, where all the interface's functionalities are programmed.

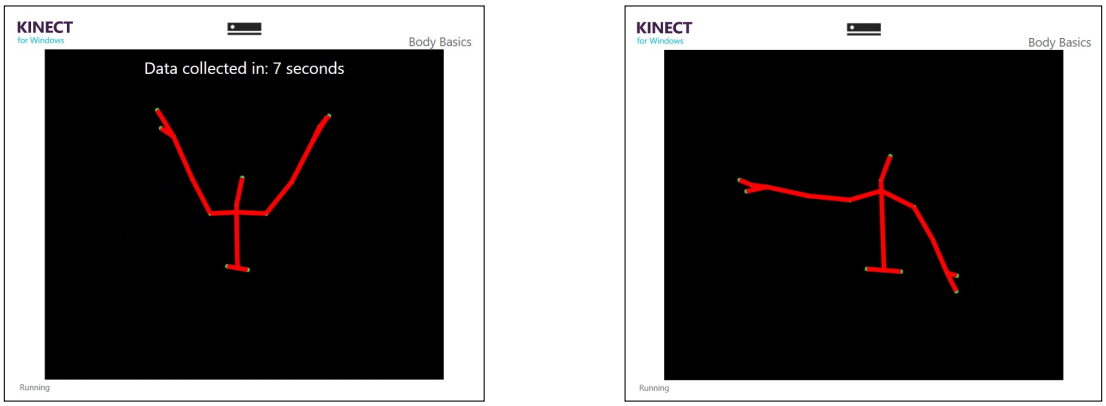
Firstly, the therapist chooses which side of the body will be assessed: the left or the right, as shown in Figure 5.23a. Next, a window will open with three options: perform the reference movement, start patient exercise and results - Figure 5.23b.



(a) (b)

Figure 5.23: User interface overview: (a) First interface window; (b) Second interface window, which has the same appearance whatever side is chosen.

In the first two options, the Kinect will be started and the movement will be detected. However, in the case of 'Begin Reference Exercise' (Figure 5.24a) it is expected that the pose will be demonstrated by the therapist and that this is the position he/she thinks the patient should reach. The data is saved 15 s after initialisation, and there is a countdown to the moment it is captured. In the case of 'Begin Patient Exercise' (Figure 5.24b), it should only be done by the patients, who should perform the movement to the best of their ability, with the data from the movement with the greatest amplitude being saved for the entire time the patient is doing the exercise; there may be several attempts, however, only the one with the best performance will be saved.



(a) (b)

Figure 5.24: User interface overview: (a) Reference exercise interface window; (b) Patient exercise interface window.

In this project, the data is saved at one point, after 15 s, and at another moment only when the maximum amplitude of the movement is reached. In both cases, the coordinates are only written to a json file when the programme is closed. In the latter case, it was considered that the maximum amplitude is reached when the y coordinate of the hand's keypoint is maximum. To determine this moment, it could be written all the coordinates to a database and then identify which case has the greatest value. However, this approach would consume a lot of memory and result in slower detection. Instead, the implemented solution compares the y coordinates of the hand's keypoint over time, updating the variable with the new coordinates whenever a value higher than the one saved is reached. Only at the end the coordinates of the moment when the patient reached the greatest amplitude will be saved. This approach ensures efficient use of memory and avoids file access conflicts, providing a more efficient solution for recording the coordinates of movements captured by Kinect.

The last option, 'Results', shows the latest results of the reference exercise and the patient's movement in two separate graphs, as well as the angle of the movement amplitude, as shown in Figure 5.25. This is so that the therapist can visualise the data obtained and, if he or she thinks the patient has performed the exercise badly or needs to adapt the reference movement, it is possible to do so without having to start the whole process all over again. Thus, all the therapist has to do is close the results window and repeat as necessary. When the results seem favourable to the rehabilitation objective, it has an option to send the data.

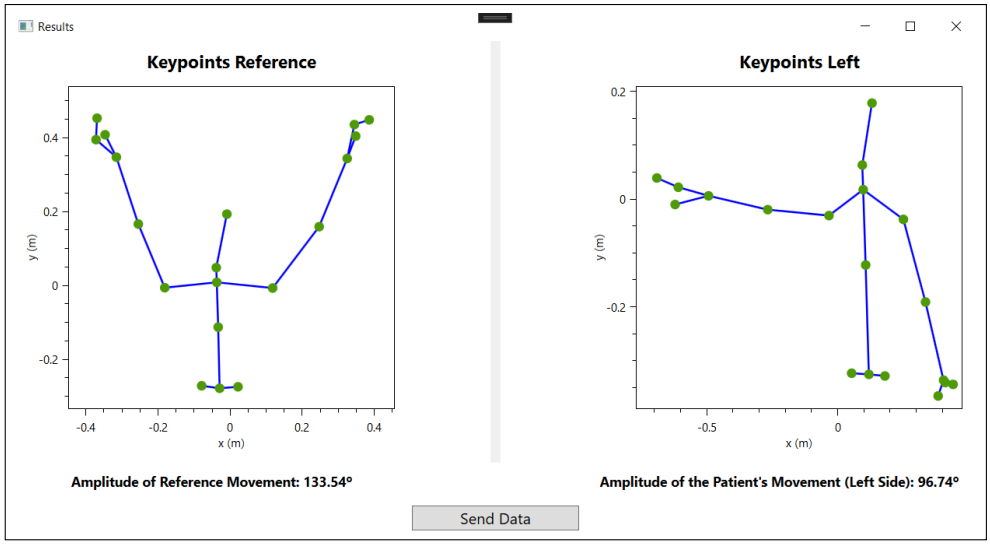


Figure 5.25: Results interface window.

The button 'Send Data' will open a new window - Figure 5.26. The values that are needed to define the trajectory points for UR5: the side that is being analysed, the position of the patient's shoulder relative to UR5, the maximum angle of movement and the radius, are calculated and displayed in this window.

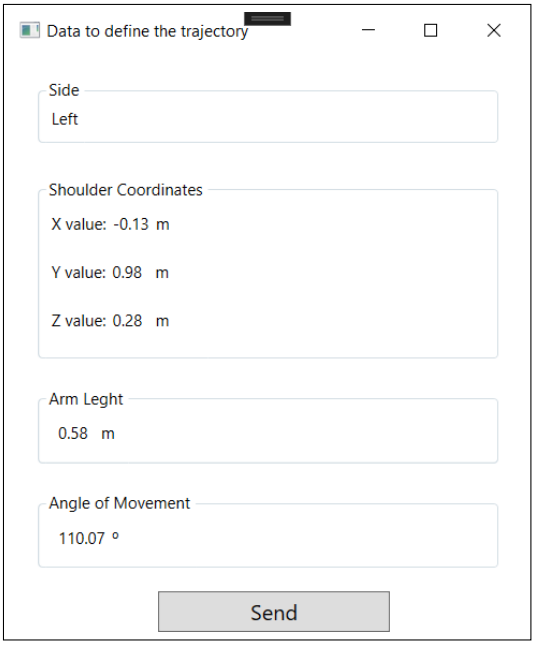


Figure 5.26: Interface window which displays the variables to define the trajectory.

Arrays containing the pose (3D coordinates and orientation) of the manipulator's tool will be sent to the UR5 when the 'Send' option is selected. If no errors occur, a message saying 'Trajectory sent successfully' will be displayed. Initially, the handle will move to a position aligned with the shoulder to ensure that it does not interfere with the patient or the bench before starting the rehabilitation movement, which will be repeated five times. Finally, the handle will be moved to a fixed position for secure storage.

5.4.4 Integrated system

After integrating the complete system, it is possible to summarise the types of connections and protocols used. As illustrated in Figure 5.27 - the Kinect connects via USB 3.0 to the computer, which itself has a connection to the UR5 with an Ethernet cable, and communicates with it via a TCP protocol.

Figure 5.27: Connections and protocols used in integrated system.

Regarding the information transmission, as shown in Figure 5.28, it can be concluded that RGB and depth information is obtained from Kinect. Using the Kinect SDK in Windows, the 3D coordinates of the keypoints of the upper limbs are extracted. From these coordinates, the centre,

radius, and maximum angle of movement are calculated to determine the trajectory, which is then used to send the 3D pose to the UR5.

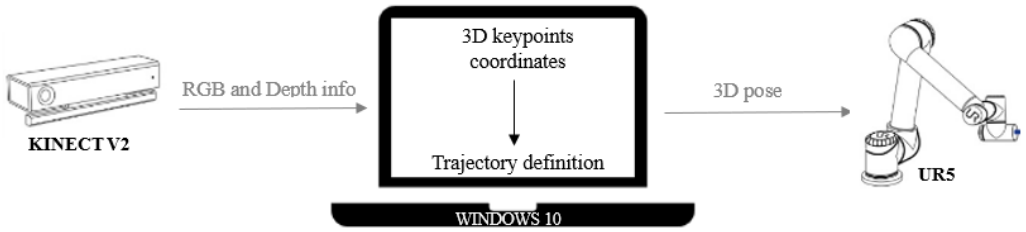


Figure 5.28: Information transmission in integrated system.

The Unified Modeling Language (UML) diagram for the entire integrated system is shown in Figure 5.29, which summarises all the processes that need to be carried out. To summarise, after starting the interface, the side to be analysed must be chosen, with the process being the same, only the calculation of the variables takes the chosen side into account. The therapist must then define, by performing the exercise, what the patient's objective is - 'Begin Reference Exercise' and, for their part, the patient must perform the abduction/adduction movement to the best of their ability - 'Begin Patient Exercise'. An analysis will then be made of the movement made and three variables are calculated: the centre, radius and maximum angle of movement. The trajectory will be calculated and then sent to the UR5 for it to carry out with the patient's arm passive.

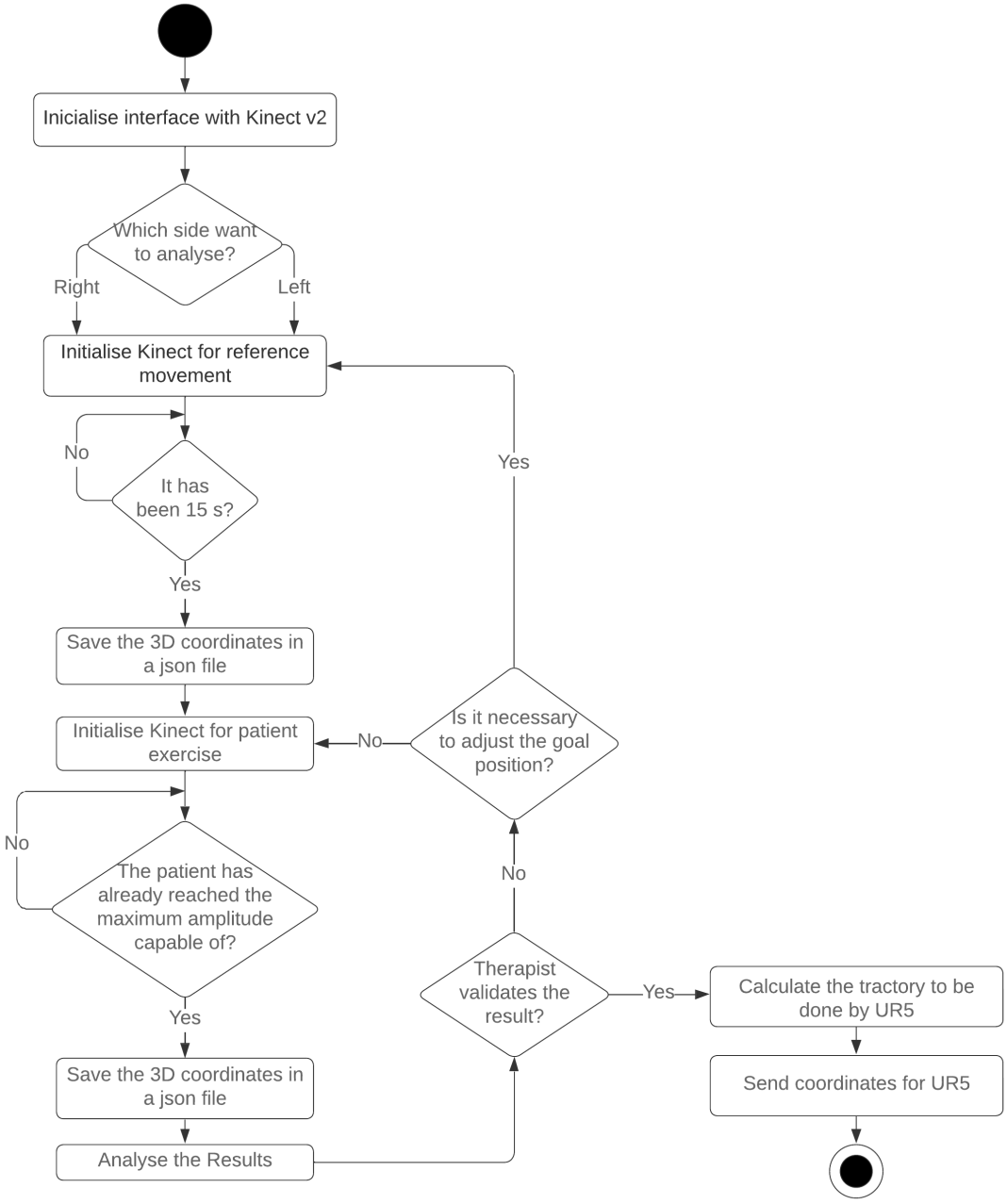


Figure 5.29: UML diagram of all the integrated system.

5.5 SWOT analysis

The SWOT analysis, summarised in Figure 5.30, provides a comprehensive overview of the project's strengths, weaknesses, opportunities and threats. Each aspect is detailed.

- The **strengths** include the fact that the system is user-friendly and the automation of the entire process. Once installed, the system is straightforward to operate and does not pose risks to the patients in a rehabilitation setting. In cases of extreme discomfort, the patients

can easily withdraw their arm from the movement. Additionally, allows the therapist to be present for support without interfering with the movement detection, which is a significant advantage. Moreover, the project can be easily adapted to other movements and different body parts.

- The main **weaknesses** involves the rigidity of the setup, which requires assembly and calibration dependent on a fixed layout, limiting flexibility. Maintaining stability in a clinical environment over extended periods can be challenging. Furthermore, the UR5 robotic arm does not have the medical certificate, which is a barrier for the implementation; however, it can be used a manipulator with this certificate, for example, the KUKA LBR Med robot (KUKA AG (2017)). In addition, the system does not consider compensatory movements that patients may make, such as moving the torso to facilitate the movement. This may occur because, with an end-effector manipulator, there are no restrictions on the body movement.

- The **opportunities** presented by this system include allowing patients to spend more time engaged in therapy and enabling therapists to oversee multiple patients simultaneously. These factors can lead to greater accessibility and improved outcomes for patients, making the system a valuable addition to the healthcare landscape.

- The primary **threats** include the possibility that older patients might find the technology unfamiliar, leading to resistance in adoption. Furthermore, physiotherapists may fear job loss due to the automation of certain tasks, which could create hesitation in embracing the system. The cost of the hardware is another concern, as it may limit accessibility and make widespread implementation challenging.

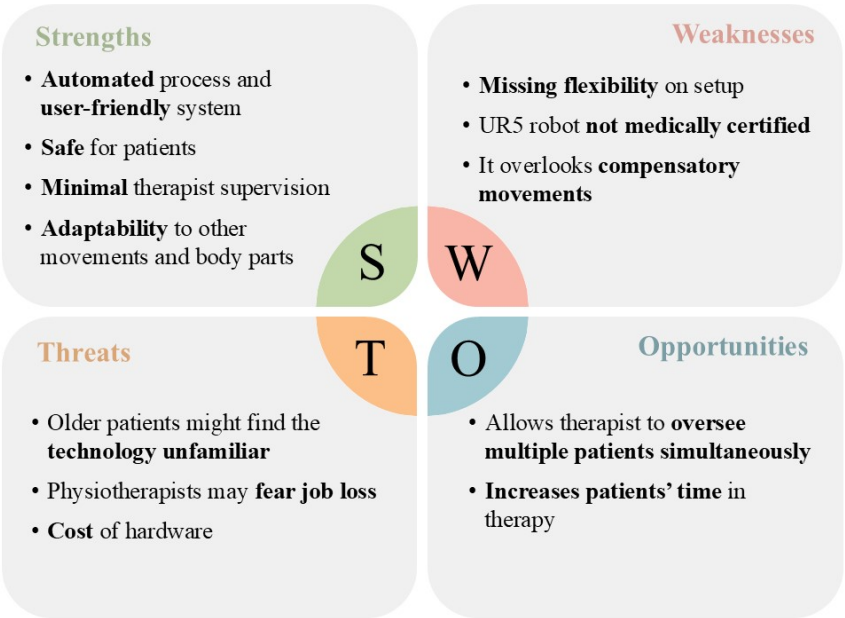


Figure 5.30: SWOT analysis.

Chapter 6

Conclusion and Future Work

In this chapter, the key findings and outcomes of the project will be summarised, highlighting the achievements and challenges found throughout its development. Additionally, the potential next steps will be discussed, focusing on the improvements and further research needed to improve the system.

6.1 Conclusions

Currently, to the best of the author' knowledge and based on the literature reviewed, there is no personalised system available for the rehabilitation of individuals with reduced upper limb mobility, which highlights a significant gap in the field. This project has addressed this shortcoming by developing a system that detects and analyses a patient's mobility and adjusts exercises automatically based on their individual capabilities.

To achieve this, an automated system was created that first detects the target movement the patient needs to achieve, performed by the therapist. It then assesses the patient's current mobility and calculates the amplitude required for the selected movement — adduction or abduction. The robotic arm subsequently moves, guiding the patient's arm passively and safely towards the target position.

Throughout the development of the project, the results and accuracy of each stage were analysed. The assessment of limb keypoint detection revealed that the Kinect v2 depth camera and its SDK can detect keypoints with precision. Comparing the real distances between two keypoints on the upper limbs with the euclidean distances calculated from Kinect data showed promising results, with standard deviations consistently under 4.6 cm and percentage errors below 5.0%. However, the largest deviations observed in some cases reached up to 8.5 cm, which could potentially impact the effectiveness of rehabilitation and necessitate repeated analyses of the patient's movements.

Additionally, it was noted that an individual with reduced mobility performed the exercises with the arm slightly tucked due to her limited range of motion, a scenario expected to occur

frequently during therapy. This highlighted the need for isolating each movement and ensuring the arm is positioned comfortably to minimise difficulties.

Factors such as brightness, background and occlusions were found not to affect detection significantly under non-extreme conditions. When multiple people were present within the camera's field of view, the method employed to detect only the first person worked effectively, without interference or alternated detection of individuals.

Since the UR5 robotic arm receives coordinates x, y, z, and orientations rx, ry, rz, it was necessary to define the trajectory in discrete steps. The variables required to define this trajectory were obtained with high accuracy. Although the rotation matrix needs to be recalculated with each setup change, the employed method proved to be reliable. The results were highly favourable, with the patient's final angle, obtained from the Kinect data, consistently deviating by less than 1.8° from the target angle.

An intuitive and user-friendly interface has been designed, so users do not need to interact directly with code or perform complex configurations to conduct therapy. This interface provides all necessary information throughout the process, enabling therapists to monitor and manage the therapy effectively.

Despite the significant hardware costs, particularly for the robotic arm, and potential market resistance — such as older patients who might be uncomfortable with new technology and therapists concerned about job security — it is essential to emphasise the substantial benefits this project offers to rehabilitation. The system has the potential to enhance therapy centre capacities and extend the time available for each patient, as therapists would be able to oversee multiple patients simultaneously.

In summary, this project has successfully developed a modular system for upper limb rehabilitation, facilitating more personalised and reliable therapy. By integrating advanced technologies like the Kinect SDK and the UR5 robotic arm, the system enables detailed analysis and real time adjustments, tailoring exercises to meet each patient's specific needs. This innovation promises to advance the field of rehabilitation, improving both patient outcomes and operational efficiency in therapy centres.

6.2 Future work

Although it has been demonstrated that automatic rehabilitation is feasible, it is crucial for therapists to evaluate the project. Their input would be invaluable for identifying potential complications and suggesting necessary adjustments, such as optimising movement speed and determining the ideal number of repetitions.

Moreover, to put the project's extrapolation into practice, it would be necessary to use a collaborative manipulator with a longer arm. For practical implementation, the UR5 robotic system will have to be modified due to the fact that it is not medical certified for these uses. The KUKA LBR Med robot is a potential candidate; however, its maximum arm length of 800 mm may still present reach challenges (KUKA AG (2017)).

It would also be interesting to replicate the system using an exoskeleton robot to understand and compare whether restricted movement would benefit therapy sessions and recovery. This way, compensatory movements can be limited, ensuring that the movements are performed more effectively.

Additionally, since the effectiveness of the system is highly dependent on the Kinect SDK's algorithm quality, training a dedicated pose detection algorithm with a rehabilitation specific dataset could enhance performance. Attempts to use the AlphaPose algorithm in 3D were hindered by insufficient computational resources, resulting in slow processing and inadequate real-time detection. An alternative approach involved using AlphaPose in 2D alongside the Kinect v2, but this also faced resource constraints. As future work, it would be interesting to compare the Kinect SDK, which is widely used, with a generic pose estimation algorithm, such as AlphaPose.

Furthermore, while this was not pursued due to the favourable results obtained, calibrating the Kinect v2 camera could be interesting. This could be accomplished using a chessboard pattern to fine-tune the alignment between the captured images and the depth data.

References

Maged S. AL-Quraishi, Irraivan Elamvazuthi, Siti Asmah Daud, S. Parasuraman, and Alberto Borboni. EEG-Based Control for Upper and Lower Limb Exoskeletons and Prostheses: A Systematic Review. *Sensors (Basel, Switzerland)*, 18(10):3342, October 2018. ISSN 1424-8220. doi: 10.3390/s18103342. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6211123/.

Justin Amadeus Albert, Victor Owolabi, Arnd Gebel, Clemens Markus Brahms, Urs Granacher, and Bert Arnrich. Evaluation of the pose tracking performance of the azure kinect and kinect v2 for gait analysis in comparison with a gold standard: A pilot study. *Sensors*, 20(18), 2020. ISSN 1424-8220. doi: 10.3390/s20185104. URL https://www.mdpi.com/1424-8220/20/18/5104.

Víctor Asanza, Enrique Peláez, Francis Loayza, Leandro L. Lorente-Leyva, and Diego H. Peluffo-Ordóñez. Identification of Lower-Limb Motor Tasks via Brain–Computer Interfaces: A Topical Overview. *Sensors (Basel, Switzerland)*, 22(5):2028, March 2022. ISSN 1424-8220. doi: 10.3390/s22052028. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8914806/.

Md Assad-Uz-Zaman, Md Rasedul Islam, Mohammad Habibur Rahman, Ying-Chih Wang, and Erin McGonigle. Kinect Controlled NAO Robot for Telerehabilitation. *Journal of Intelligent Systems*, 30(1):224–239, January 2021. ISSN 2191-026X. doi: 10.1515/jisys-2019-0126. URL https://www.degruyter.com/document/doi/10.1515/jisys-2019-0126/html.

Angelo Basteris, Sharon M. Nijenhuis, Arno H.A. Stienen, Jaap H. Buurke, Gerdienke B. Prange, and Farshid Amirabdollahian. Training modalities in robot-mediated upper limb rehabilitation in stroke: a framework for classification based on a systematic review. *Journal of NeuroEngineering and Rehabilitation 2014 11:1*, 11:1–15, 7 2014. ISSN 1743-0003. doi: 10.1186/1743-0003-11-111. URL https://link.springer.com/article/10.1186/1743-0003-11-111.

Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang, and Matthias Grundmann. BlazePose: On-device Real-time Body Pose tracking, June 2020. URL http://arxiv.org/abs/2006.10204.

Martyna Białecka, Kacper Gruszczyński, Paweł Cisowski, Jakub Kaszyński, Cezary Baka, and Przemysław Lubiatowski. Shoulder range of motion measurement using inertial measurement unit—validation with a robot arm. *Sensors 2023, Vol. 23, Page 5364*, 23:5364, 6 2023. ISSN 1424-8220. doi: 10.3390/S23125364. URL https://www.mdpi.com/1424-8220/23/12/5364.

Joseph E. Bielecki and Prasanna Tadi. Therapeutic Exercise. In *StatPearls [Internet]*. StatPearls Publishing, 7 2023. URL https://www.ncbi.nlm.nih.gov/books/NBK555914/. Last accessed 29 october 2023.

BIONIK. InMotion® ARM, 2023. URL https://bioniklabs.com/inmotion-arm/. Last accessed 26 october 2023.

Amy A. Blank, James A. French, Ali Utku Pehlivan, and Marcia K. O'Malley. Current Trends in Robot-Assisted Upper-Limb Stroke Rehabilitation: Promoting Patient Engagement in Therapy. *Current Physical Medicine and Rehabilitation Reports*, 2(3):184–195, September 2014. ISSN 2167-4833. doi: 10.1007/s40141-014-0056-z. URL https://doi.org/10.1007/s40141-014-0056-z.

Rocco Salvatore Calabrò, Giovanni Morone, Antonino Naro, Marialuisa Gandolfi, Vitalma Liotti, Carlo D'aurizio, Sofia Straudi, Antonella Focacci, Sanaz Pournajaf, Irene Aprile, Serena Filoni, Claudia Zanetti, Maria Rosaria Leo, Lucia Tedesco, Vincenzo Spina, Carmelo Chisari, Giovanni Taveggia, Stefano Mazzoleni, Nicola Smania, Stefano Paolucci, Marco Franceschini, and Donatella Bonaiuti. Robot-Assisted Training for Upper Limb in Stroke (ROBOTAS): An Observational, Multicenter Study to Identify Determinants of Efficacy. *Journal of Clinical Medicine*, 10(22):5245, November 2021. ISSN 2077-0383. doi: 10.3390/jcm10225245. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8622640/.

Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1302–1310, Honolulu, HI, July 2017. IEEE. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.143. URL http://ieeexplore.ieee.org/document/8099626/.

Alan Chan. A technical report on a novel robotic lower limb rehabilitation device - Is ROBERT® a cost-effective solution for rehabilitation in Hong Kong? *Hong Kong Physiotherapy Journal*, 42(2):75–80, December 2022. ISSN 1013-7025. doi: 10.1142/S1013702522710019. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10406640/.

Kenny Chen, Paolo Gabriel, Abdulwahab Alasfour, Chenghao Gong, Werner K. Doyle, Orrin Devinsky, Daniel Friedman, Patricia Dugan, Lucia Melloni, Thomas Thesen, David Gonda, Shifteh Sattar, Sonya Wang, and Vikash Gilja. Patient-Specific Pose Estimation in Clinical Environments. *IEEE Journal of Translational Engineering in Health and Medicine*, 6:2101111, October 2018a. ISSN 2168-2372. doi: 10.1109/JTEHM.2018.2875464. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6255526/.

Sung-Hua Chen, Wei-Ming Lien, Wei-Wen Wang, Guan-De Lee, Li-Chun Hsu, Kai-Wen Lee, Sheng-Yen Lin, Chia-Hsun Lin, Li-Chen Fu, Jin-Shin Lai, Jer-Junn Luh, and Wen-Shiang Chen. Assistive Control System for Upper Limb Rehabilitation Robot. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 24(11):1199–1209, November 2016. ISSN 1558-0210. doi: 10.1109/TNSRE.2016.2532478. URL https://ieeexplore.ieee.org/abstract/document/7419333.

Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded Pyramid Network for Multi-person Pose Estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7103–7112, Salt Lake City, UT, June 2018b. IEEE. ISBN 978-1-5386-6420-9. doi: 10.1109/CVPR.2018.00742. URL https://ieeexplore.ieee.org/document/8578840/.

Wai-tong Chien, Yuen-yu Chong, Man-kei Tse, Cheuk-woon Chien, and Ho-yu Cheng. Robot-assisted therapy for upper-limb rehabilitation in subacute stroke patients: A systematic review and meta-analysis. *Brain and Behavior*, 10(8):e01742, 2020. ISSN 2162-3279. doi: 10.1002/brb3.1742. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/brb3.1742.

Chuang-Yuan Chiu, Michael Thelwell, Terry Senior, Simon Choppin, John Hart, and Jon Wheat. Comparison of depth cameras for three-dimensional reconstruction in medicine. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 233(9): 938–947, September 2019. ISSN 0954-4119, 2041-3033. doi: 10.1177/0954411919859922. URL http://journals.sagepub.com/doi/10.1177/0954411919859922.

Jiaming Chua, Lee-Yeng Ong, and Meng-Chew Leow. Telehealth Using PoseNet-Based System for In-Home Rehabilitation. *Future Internet*, 13(7):173, July 2021. ISSN 1999-5903. doi: 10.3390/fi13070173. URL https://www.mdpi.com/1999-5903/13/7/173.

Jen-Li Chung, Lee-Yeng Ong, and Meng-Chew Leow. Comparative analysis of skeleton-based human pose estimation. *Future Internet*, 14(12), 2022. ISSN 1999-5903. doi: 10.3390/fi14120380. URL https://www.mdpi.com/1999-5903/14/12/380.

R. Colombo, F. Pisano, S. Micera, A. Mazzone, C. Delconte, M.C. Carrozza, P. Dario, and G. Minuco. Robotic techniques for upper limb evaluation and rehabilitation of stroke patients. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 13(3):311–324, September 2005. ISSN 1558-0210. doi: 10.1109/TNSRE.2005.848352. URL https://ieeexplore.ieee.org/abstract/document/1506818.

Louise A. Connell, Naoimh E. McMahon, Lisa A. Simpson, Caroline L. Watkins, and Janice J. Eng. Investigating measures of intensity during a structured upper limb exercise programme in stroke rehabilitation: An exploratory study. *Archives of physical medicine and rehabilitation*, 95(12):2410–2419, December 2014. ISSN 0003-9993. doi: 10.1016/j.apmr.2014.05.025. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4485917/.

Jean E. Cromie, Valma J. Robertson, and Margaret O. Best. Work-related musculoskeletal disorders in physical therapists: prevalence, severity, risks, and responses. *Physical therapy*, 80: 336–351, 2000. ISSN 0031-9023. doi: 10.1093/PTJ/80.4.336. URL https://pubmed.ncbi.nlm.nih.gov/10758519/.

Bappaditya Debnath, Mary O'Brien, Motonori Yamaguchi, and Ardhendu Behera. A review of computer vision-based approaches for physical rehabilitation and assessment. *Multimedia Systems*, 28(1):209–239, February 2022. ISSN 1432-1882. doi: 10.1007/s00530-021-00815-4. URL https://doi.org/10.1007/s00530-021-00815-4.

Microsoft Developers. Kinect for windows SDK 2.0, 2014. URL https://www.microsoft.com/en-us/download/details.aspx?id=44561. Last accessed 20 may 2024.

Bruce H. Dobkin and Andrew Dorsch. New evidence for therapies in stroke rehabilitation. *Current atherosclerosis reports*, 15(6):331, June 2013. ISSN 1523-3804. doi: 10.1007/s11883-013-0331-y. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3679365/.

Simone Dorsch, Kevin Weeks, Laura King, and Etesa Polman. In inpatient rehabilitation, large amounts of practice can occur safely without direct therapist supervision: an observational study. *Journal of Physiotherapy*, 65(1):23–27, January 2019. ISSN 1836-9553.

doi: 10.1016/j.jphys.2018.11.004. URL https://www.sciencedirect.com/science/article/pii/S1836955318301474.

Pamela W. Duncan, Dennis Wallace, Sue Min Lai, Dallas Johnson, Susan Embretson, and Louise Jacobs Laster. The Stroke Impact Scale Version 2.0. *Stroke*, 30(10):2131–2140, October 1999. doi: 10.1161/01.STR.30.10.2131. URL https://www.ahajournals.org/doi/10.1161/01.str.30.10.2131. Publisher: American Heart Association.

Christophe Duret, Ophélie Pila, anne gaelle Grosmaire, and Typhaine Koeppel. Can robot-based measurements improve prediction of motor performance after robot-assisted upper-limb rehabilitation in patients with moderate-to-severe sub-acute stroke? *Restorative Neurology and Neuroscience*, 37:119–129, April 2019. doi: 10.3233/RNN-180892.

Hao-Shu Fang, Jiefeng Li, Hongyang Tang, Chao Xu, Haoyi Zhu, Yuliang Xiu, Yong-Lu Li, and Cewu Lu. AlphaPose: Whole-Body Regional Multi-Person Pose Estimation and Tracking in Real-Time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7157–7173, June 2023. ISSN 1939-3539. doi: 10.1109/TPAMI.2022.3222784. URL https://ieeexplore.ieee.org/abstract/document/9954214.

Marco Germanotta, Laura Cortellini, Sabina Insalaco, and Irene Aprile. Effects of upper limb robot-assisted rehabilitation compared with conventional therapy in patients with stroke: Preliminary results on a daily task assessed using motion analysis. *Sensors*, 23:3089, 3 2023. ISSN 1424-8220. doi: 10.3390/S23063089. URL https://www.mdpi.com/1424-8220/23/6/3089.

David J. Gladstone, Cynthia J. Danells, and Sandra E. Black. The fugl-meyer assessment of motor recovery after stroke: a critical review of its measurement properties. *Neurorehabilitation and Neural Repair*, 16(3):232–240, September 2002. ISSN 1545-9683. doi: 10.1177/154596802401105171.

Ranathunga A. R. C. Gopura and Kazuo Kiguchi. Mechanical designs of active upper-limb exoskeleton robots: State-of-the-art and design difficulties. In *2009 IEEE International Conference on Rehabilitation Robotics*, pages 178–187, June 2009. doi: 10.1109/ICORR.2009.5209630. URL https://ieeexplore.ieee.org/document/5209630. ISSN: 1945-7901.

Ranathunga A. R. C. Gopura, D. S. V. Bandara, Kazuo Kiguchi, and G. K. I. Mann. Developments in hardware systems of active upper-limb exoskeleton robots: A review. *Robotics and Autonomous Systems*, 75:203–220, January 2016. ISSN 0921-8890. doi: 10.1016/j.robot.2015.10.001. URL https://www.sciencedirect.com/science/article/pii/S0921889015002274.

Yiwen Gu, Shreya Pandit, Elham Saraee, Timothy Nordahl, Terry Ellis, and Margrit Betke. Home-Based Physical Therapy with an Interactive Computer Vision System. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 2619–2628, Seoul, Korea (South), October 2019. IEEE. ISBN 978-1-72815-023-9. doi: 10.1109/ICCVW.2019.00320. URL https://ieeexplore.ieee.org/document/9022034/.

Riza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. DensePose: Dense Human Pose Estimation in the Wild. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7297–7306, June 2018. doi: 10.1109/CVPR.2018.00762. URL https://ieeexplore.ieee.org/document/8578860. ISSN: 2575-7075.

Muhammad Ahsan Gull, Shaoping Bai, and Thomas Bak. A review on design of upper limb exoskeletons. *Robotics 2020, Vol. 9, Page 16*, 9:16, 3 2020. ISSN 2218-6581. doi: 10.3390/ROBOTICS9010016. URL https://www.mdpi.com/2218-6581/9/1/16.

Andrea Guzzetta, Martin Staudt, Elisa Petacchi, Jan Ehlers, Michael Erb, Marko Wilke, Ingeborg Krägeloh-Mann, and Giovanni Cioni. Brain Representation of Active and Passive Hand Movements in Children. *Pediatric Research*, 61(4):485–490, April 2007. ISSN 1530-0447. doi: 10.1203/pdr.0b013e3180332c2e. URL https://www.nature.com/articles/pr200794.

Jungong Han, Ling Shao, Dong Xu, and Jamie Shotton. Enhanced Computer Vision With Microsoft Kinect Sensor: A Review. *IEEE Transactions on Cybernetics*, 43(5):1318–1334, October 2013. ISSN 2168-2275. doi: 10.1109/TCYB.2013.2265378. URL https://ieeexplore.ieee.org/abstract/document/6547194.

John T. Hansen. *Netter's Clinical Anatomy*. Elsevier, 4th edition, 2019.

Óscar G. Hernández, Vicente Morell, José L. Ramon, and Carlos A. Jara. Human Pose Detection for Robotic-Assisted and Rehabilitation Environments. *Applied Sciences*, 11(9):4183, January 2021. ISSN 2076-3417. doi: 10.3390/app11094183. URL https://www.mdpi.com/2076-3417/11/9/4183.

Hocoma. Armeo®Power, 2023a. URL https://www.hocoma.com/solutions/armeo-power/. Last accessed 26 october 2023.

Hocoma. Armeo®Spring, 2023b. URL https://www.hocoma.com/solutions/armeo-spring/. Last accessed 26 october 2023.

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, April 2017. URL http://arxiv.org/abs/1704.04861.

HUMANWARE. Motore, 2023. URL https://www.hmw.it/en/home/. Last accessed 26 october 2023.

Shafqatullah Jan, Haider Darain, Aatik Arsh, and Shehla Gul. A randomized control trial comparing the effects of motor relearning programme and mirror therapy for improving upper limb motor functions in stroke patients. *Journal of the Pakistan Medical Association*, 69, September 2019.

Prabhakara Rao Kapula, Anudeep Deekonda, Sanjay Dubey, B. Sushannth, B. Sai Srikar, and B. Varshith Reddy. Design and Implementation of Motion Imitation Robot. In *2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pages 1–5, Istanbul, Turkiye, June 2023. IEEE. ISBN 9798350337525. doi: 10.1109/HORA58378.2023.10156711. URL https://ieeexplore.ieee.org/document/10156711/.

Gurpreet Kaur, Coralie English, and Susan Hillier. How Physically Active Are People with Stroke in Physiotherapy Sessions Aimed at Improving Motor Function? A Systematic Review. *Stroke Research and Treatment*, 2012:820673, 2012. ISSN 2090-8105. doi: 10.1155/2012/820673. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3337516/.

M. Kelly-Hayes, J. T. Robertson, J. P. Broderick, P. W. Duncan, L. A. Hershey, E. J. Roth, W. H. Thies, and C. A. Trombly. The american heart association stroke outcome classification. *Stroke*, 29:1274–1280, 1998. ISSN 00392499. doi: 10.1161/01.STR.29.6.1274. URL https://www.ahajournals.org/doi/abs/10.1161/01.STR.29.6.1274.

Asifullah Khan, Anabia Sohail, Umme Zahoora, and Aqsa Saeed Qureshi. A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53(8):5455–5516, December 2020. ISSN 1573-7462. doi: 10.1007/s10462-020-09825-6. URL https://doi.org/10.1007/s10462-020-09825-6.

KUKA AG. KUKA LBR med lightweight robot, 2017. URL https://www.kuka.com/en-de/company/press/news/2017/08/kuka-lbr-med-for-integration-into-a-medical-product. Last accessed 22 august 2024.

Erik Kyrkjebø, Mads Johan Laastad, and Øyvind Stavdahl. Feasibility of the UR5 Industrial Robot for Robotic Rehabilitation of the Upper Limbs After Stroke. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–6, October 2018. doi: 10.1109/IROS.2018.8594413. URL https://ieeexplore.ieee.org/abstract/document/8594413. ISSN: 2153-0866.

Yujun Lai, Sheila Sutjipto, Matthew D. Clout, Marc G. Carmichael, and Gavin Paul. GAVRe2: Towards Data-Driven Upper-Limb Rehabilitation with Adaptive-Feedback Gamification. In *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 164–169, December 2018. doi: 10.1109/ROBIO.2018.8665105. URL https://ieeexplore.ieee.org/abstract/document/8665105.

Peter Langhorne, Fiona Coupar, and Alex Pollock. Motor recovery after stroke: a systematic review. *The Lancet. Neurology*, 8:741–754, 8 2009. ISSN 1474-4422. doi: 10.1016/S1474-4422(09)70150-4. URL https://pubmed.ncbi.nlm.nih.gov/19608100/.

Jeffrey Laut, Maurizio Porfiri, and Preeti Raghavan. The Present and Future of Robotic Technology in Rehabilitation. *Current Physical Medicine and Rehabilitation Reports*, 4(4):312–319, December 2016. ISSN 2167-4833. doi: 10.1007/s40141-016-0139-0. URL https://doi.org/10.1007/s40141-016-0139-0.

Stephanie Hyeyoung Lee, Gyulee Park, Duk Youn Cho, Ha Yeon Kim, Ji-Yeong Lee, Suyoung Kim, Si-Bog Park, and Joon-Ho Shin. Comparisons between end-effector and exoskeleton rehabilitation robots regarding upper extremity function among chronic stroke patients with moderate-to-severe upper limb impairment. *Scientific Reports*, 10(1):1806, February 2020. ISSN 2045-2322. doi: 10.1038/s41598-020-58630-2. URL https://www.nature.com/articles/s41598-020-58630-2.

Miaopeng Li, Zimeng Zhou, Jie Li, and Xinguo Liu. Bottom-up Pose Estimation of Multiple Person with Bounding Box Constraint. In *2018 24th International Conference on Pattern Recognition*, pages 115–120, August 2018. doi: 10.1109/ICPR.2018.8546194.

Life Science Robotics. Rehab Robot ROBERT supports physiotherapists in their work, 2020. URL https://www.kuka.com/en-se/company/press/news/2020/04/robert-rehab-robot. Last accessed 30 december 2023.

Yilin Liu, Shijia Zhang, and Mahanth Gowda. NeuroPose: 3D Hand Pose Tracking using EMG Wearables. In *Proceedings of the Web Conference 2021*, pages 1471–1482, Ljubljana Slovenia, 4 2021. ACM. doi: 10.1145/3442381.3449890. URL https://dl.acm.org/doi/10.1145/3442381.3449890.

Peter S. Lum, Charles G. Burgar, Peggy C. Shor, Matra Majmundar, and Machiel Van der Loos. Robot-assisted movement training compared with conventional therapy techniques for the rehabilitation of upper-limb motor function after stroke. *Archives of Physical Medicine and Rehabilitation*, 83(7):952–959, July 2002. ISSN 0003-9993. doi: 10.1053/apmr.2001.33101. URL https://www.sciencedirect.com/science/article/pii/S0003999302000114.

Cameron S. Mang, Kristin L. Campbell, Colin J.D. Ross, and Lara A. Boyd. Promoting Neuroplasticity for Motor Rehabilitation After Stroke: Considering the Effects of Aerobic Exercise and Genetic Variation on Brain-Derived Neurotrophic Factor. *Physical Therapy*, 93 (12):1707–1716, December 2013. ISSN 0031-9023. doi: 10.2522/ptj.20130053. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3870490/.

Stefano Masiero and Elena Carraro. Upper limb movements and cerebral plasticity in post-stroke rehabilitation. *Aging Clinical and Experimental Research*, 20(2):103–108, April 2008. ISSN 1720-8319. doi: 10.1007/BF03324755. URL https://doi.org/10.1007/BF03324755.

Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: real-time 3d human pose estimation with a single rgb camera. *ACM Trans. Graph.*, 36(4), jul 2017. ISSN 0730-0301. doi: 10.1145/3072959.3073596. URL https://doi.org/10.1145/3072959.3073596.

Jeison Mejia-Trujillo, Yor Castano-Pino, Andres Navarro, Juan Arango-Paredes, Domiciano Rincon, Jaime Valderrama, Beatriz Munoz-Ospina, and Jorge Orozco. Kinect™ and Intel RealSense™ D435 comparison: a preliminary study for motion analysis. In *2019 IEEE International Conference on E-health Networking, Application Services (HealthCom)*, pages 1–4, October 2019. doi: 10.1109/HealthCom46333.2019.9009433.

Pierre Merriaux, Yohan Dupuis, Rémi Boutteau, Pascal Vasseur, and Xavier Savatier. A Study of Vicon System Positioning Performance. *Sensors*, 17(7):1591, July 2017. ISSN 1424-8220. doi: 10.3390/s17071591. URL https://www.mdpi.com/1424-8220/17/7/1591. Number: 7 Publisher: Multidisciplinary Digital Publishing Institute.

Franco Molteni, Giulio Gasperini, Giovanni Cannaviello, and Eleonora Guanziroli. Exoskeleton and end-effector robots for upper and lower limbs rehabilitation: Narrative review. *PMR*, 10: S174–S188, 9 2018. ISSN 1934-1563. doi: 10.1016/J.PMRJ.2018.06.005. URL https://onlinelibrary.wiley.com/doi/full/10.1016/j.pmrj.2018.06.005.

Giovanni Morone, Marco Tramontano, Marco Iosa, Jacob Shofany, Antonella Iemma, Massimo Musicco, Stefano Paolucci, and Carlo Caltagirone. The Efficacy of Balance Training with Video Game-Based Therapy in Subacute Stroke Patients: A Randomized Controlled Trial. *BioMed Research International*, 2014:e580861, May 2014. ISSN 2314-6133. doi: 10.1155/2014/580861. URL https://www.hindawi.com/journals/bmri/2014/580861/.

Jyotindra Narayan, Bhaben Kalita, and Santosha Kumar Dwivedy. Development of Robot-Based Upper Limb Devices for Rehabilitation Purposes: a Systematic Review. *Augmented Human*

Research, 6(1):4, January 2021. ISSN 2365-4325. doi: 10.1007/s41133-020-00043-x. URL https://doi.org/10.1007/s41133-020-00043-x.

Ziad S. Nasreddine, Natalie A. Phillips, Valérie Bédirian, Simon Charbonneau, Victor Whitehead, Isabelle Collin, Jeffrey L. Cummings, and Howard Chertkow. The Montreal Cognitive Assessment, MoCA: A Brief Screening Tool For Mild Cognitive Impairment. *Journal of the American Geriatrics Society*, 53(4):695–699, 2005. ISSN 1532-5415. doi: 10.1111/j.1532-5415.2005.53221.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1532-5415.2005.53221.x. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1532-5415.2005.53221.x.

Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked Hourglass Networks for Human Pose Estimation. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, volume 9912, pages 483–499. Springer International Publishing, Cham, 2016. ISBN 978-3-319-46483-1 978-3-319-46484-8. doi: 10.1007/978-3-319-46484-8_29. URL http://link.springer.com/10.1007/978-3-319-46484-8_29.

Xuecheng Nie, Jiashi Feng, Junliang Xing, Shengtao Xiao, and Shuicheng Yan. Hierarchical Contextual Refinement Networks for Human Pose Estimation. *IEEE Transactions on Image Processing*, 28(2):924–936, February 2019. ISSN 1941-0042. doi: 10.1109/TIP.2018.2872628. URL https://ieeexplore.ieee.org/abstract/document/8482293.

Vítor M. Oliveira, Diogo Pereira, Bruno Oliveira, Pedro Morais, Duarte Duque, João L. Vilaça, and António H. J. Moreira. A Robotic and Game-Based Framework for Assisted Upper Limb Rehabilitation. In *2022 IEEE 10th International Conference on Serious Games and Applications for Health(SeGAH)*, pages 1–7, August 2022. doi: 10.1109/SEGAH54908.2022.9978593. URL https://ieeexplore.ieee.org/abstract/document/9978593. ISSN: 2573-3060.

Leonid Pishchulin, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Poselet Conditioned Pictorial Structures. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–595, Portland, OR, USA, June 2013. IEEE. ISBN 978-0-7695-4989-7. doi: 10.1109/CVPR.2013.82. URL http://ieeexplore.ieee.org/document/6618926/.

Leon Poltawski, Rhoda Allison, Simon Briscoe, Jennifer Freeman, Cherry Kilbride, Debbie Neal, Ailie J. Turton, and Sarah Dean. Assessing the impact of upper limb disability following stroke: a qualitative enquiry using internet-based personal accounts of stroke survivors. *Disability and Rehabilitation*, 38:945, 5 2016. ISSN 14645165. doi: 10.3109/09638288.2015.1068383. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4819828/.

Hassan M. Qassim and W. Z. Wan Hasan. A review on upper limb rehabilitation robots. *Applied Sciences 2020, Vol. 10, Page 6976*, 10:6976, 10 2020. ISSN 2076-3417. doi: 10.3390/APP10196976. URL https://www.mdpi.com/2076-3417/10/19/6976.

Preeti Raghavan. Upper Limb Motor Impairment Post Stroke. *Physical medicine and rehabilitation clinics of North America*, 26(4):599–610, November 2015. ISSN 1047-9651. doi: 10.1016/j.pmr.2015.06.008. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4844548/.

Varun Ramakrishna, Daniel Munoz, Martial Hebert, James Andrew Bagnell, and Yaser Sheikh. Pose Machines: Articulated Pose Estimation via Inference Machines. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, Lecture

Notes in Computer Science, pages 33–47, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10605-2. doi: 10.1007/978-3-319-10605-2_3.

Estiko Rijanto, Erik Adiwiguna, Latif Rozaqi, Aryo Putro Sadono, and M. Hafil Nugraha. Experimental Performance Evaluation of Computer Vision for an Upper Limbs Rehabilitation Robot. In *2019 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, pages 59–63, October 2019. doi: 10.1109/IC3INA48034.2019.8949599. URL https://ieeexplore.ieee.org/abstract/document/8949599?casa_token=75OI1PGtOB8AAAAA:C8MPPOlaLsb-h6g3wAF4nlTNM3i6Xs_wrlWAB-_jHsJp98gWW86s2YCWH7vTyHzbXQxIglCh5iE.

Marta Rodríguez-Hernández, Begoña Polonio-López, Ana-Isabel Corregidor-Sánchez, José L. Martín-Conty, Alicia Mohedano-Moriano, and Juan-José Criado-Álvarez. Can specific virtual reality combined with conventional rehabilitation improve poststroke hand motor function? A randomized clinical trial. *Journal of NeuroEngineering and Rehabilitation*, 20 (1):38, April 2023. ISSN 1743-0003. doi: 10.1186/s12984-023-01170-3. URL https://doi.org/10.1186/s12984-023-01170-3.

Silvia Salvalaggio, Luisa Cacciante, Lorenza Maistrello, and Andrea Turolla. Clinical predictors for upper limb recovery after stroke rehabilitation: Retrospective cohort study. *Healthcare*, 2023. doi: 10.3390/healthcare11030335. URL https://doi.org/10.3390/healthcare11030335.

SAMCON. ReoGo, 2018. URL https://samcon.be/Rehabilitation/Robotics/ReoGo. Last accessed 26 october 2023.

Valerie Scanlon and Tina Sanders. *Essentials of Anatomy and Physiology*. F. A. Davis Company, 2007.

Manali Shah and Ruchi Desai. Prevalence, Risk Factors and Prevention of Work-Related Musculoskeletal Disorders in Physiotherapist According to Their Specialization - A Review. *International Journal of Research and Review*, 9:485–495, March 2022. doi: 10.52403/ijrr.20220354.

Bo Sheng, Yanxin Zhang, Wei Meng, Chao Deng, and Shengquan Xie. Bilateral robots for upper-limb stroke rehabilitation: State of the art and future prospects. *Medical Engineering & Physics*, 38(7):587–606, July 2016. ISSN 1350-4533. doi: 10.1016/j.medengphy.2016.04.004. URL https://www.sciencedirect.com/science/article/pii/S1350453316300480.

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, January 2013. ISSN 0001-0782, 1557-7317. doi: 10.1145/2398356.2398381. URL https://dl.acm.org/doi/10.1145/2398356.2398381.

Valeria del C. Silva-Acosta, Israel Román-Godínez, Sulema Torres-Ramos, and Ricardo A. Salido-Ruiz. Automatic estimation of continuous elbow flexion–extension movement based on electromyographic and electroencephalographic signals. *Biomedical Signal Processing and Control*, 70:102950, September 2021. ISSN 1746-8094. doi: 10.1016/j.bspc.2021.102950. URL https://www.sciencedirect.com/science/article/pii/S1746809421005474.

Daniel Smilkov, Nikhil Thorat, Yannick Assogba, Ann Yuan, Nick Kreeger, Ping Yu, Kangyi Zhang, Shanqing Cai, Eric Nielsen, David Soergel, Stan Bileschi, Michael Terry, Charles Nicholson, Sandeep N. Gupta, Sarah Sirajuddin, D. Sculley, Rajat Monga, Greg Corrado, Fernanda B. Viégas, and Martin Wattenberg. TensorFlow.js: Machine Learning for the Web and Beyond, February 2019. URL http://arxiv.org/abs/1901.05350.

Cathy M Stinear, Catherine E Lang, Steven Zeiler, and Winston D Byblow. Advances and challenges in stroke rehabilitation. *The Lancet Neurology*, 19(4):348–360, April 2020. ISSN 1474-4422. doi: 10.1016/S1474-4422(19)30415-6. URL https://www.sciencedirect.com/science/article/pii/S1474442219304156.

Min Sun and Silvio Savarese. Articulated part-based model for joint object detection and pose estimation. In *2011 International Conference on Computer Vision*, pages 723–730, November 2011. doi: 10.1109/ICCV.2011.6126309. URL https://ieeexplore.ieee.org/document/6126309. ISSN: 2380-7504.

Tangfei Tao, Xingyu Yang, Jiayu Xu, Wei Wang, Sicong Zhang, Ming Li, and Guanghua Xu. Trajectory Planning of Upper Limb Rehabilitation Robot Based on Human Pose Estimation. In *2020 17th International Conference on Ubiquitous Robots (UR)*, pages 333–338, Kyoto, Japan, June 2020. IEEE. ISBN 978-1-72815-715-3. doi: 10.1109/UR49135.2020.9144771. URL https://ieeexplore.ieee.org/document/9144771/.

Sahel Taravati, Kazim Capaci, Hale Uzumcugil, and Goksel Tanigor. Evaluation of an upper limb robotic rehabilitation program on motor functions, quality of life, cognition, and emotional status in patients with stroke: a randomized controlled study. *Neurological Sciences*, 43 (2):1177–1188, February 2022. ISSN 1590-3478. doi: 10.1007/s10072-021-05431-8. URL https://doi.org/10.1007/s10072-021-05431-8.

Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 648–656, 2015. doi: 10.1109/CVPR.2015.7298664.

Alexander Toshev and Christian Szegedy. DeepPose: Human Pose Estimation via Deep Neural Networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, June 2014. doi: 10.1109/CVPR.2014.214. URL https://ieeexplore.ieee.org/document/6909610. ISSN: 1063-6919.

Emilio Trigili, Lorenzo Grazi, Simona Crea, Alessandro Accogli, Jacopo Carpaneto, Silvestro Micera, Nicola Vitiello, and Alessandro Panarese. Detection of movement onset using EMG signals for upper-limb exoskeletons in reaching tasks. *Journal of NeuroEngineering and Rehabilitation*, 16(1):45, March 2019. ISSN 1743-0003. doi: 10.1186/s12984-019-0512-1. URL https://doi.org/10.1186/s12984-019-0512-1.

Ourania Tsilomitrou, Konstantinos Gkountas, Nikolaos Evangeliou, and Evangelos Dermatas. Wireless Motion Capture System for Upper Limb Rehabilitation. *Applied System Innovation*, 4(1):14, March 2021. ISSN 2571-5577. doi: 10.3390/asi4010014. URL https://www.mdpi.com/2571-5577/4/1/14.

Universal Robots. UR5e Lightweight, versatile cobot, 2023a. URL https://www.universal-robots.com/products/ur5-robot/. Last accessed 30 december 2023.

Universal Robots. Ur5e technical specification, 2023b. URL https://www.universal-robots.com/media/1807465/ur5e-rgb-fact-sheet-landscape-a4.pdf. Last accessed 20 june 2024.

Universal Robots A/S. UniversalRobots/universal_robots_ros_driver, 2020. URL https://github.com/UniversalRobots/Universal_Robots_ROS_Driver. Last accessed 24 august 2024.

Universal Robots A/S. *The URScript Programming Language*, 2021a. URL https://s3-eu-west-1.amazonaws.com/ur-support-site/77592/scriptManual_CB3_3.14.3.pdf.

Universal Robots A/S. *User manual*, 2021b. URL https://s3-eu-west-1.amazonaws.com/ur-support-site/17958/manual_en_UR5_CB2_1.6_US.pdf.

Eline van der Kruk and Marco M. Reijne. Accuracy of human motion capture systems for sport applications; state-of-the-art review. *European Journal of Sport Science*, 18(6):806–819, July 2018. ISSN 1746-1391. doi: 10.1080/17461391.2018.1463397. URL https://doi.org/10.1080/17461391.2018.1463397.

Valeria Villani, Fabio Pini, Francesco Leali, and Cristian Secchi. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics*, 55:248–266, November 2018. ISSN 0957-4158. doi: 10.1016/j.mechatronics.2018.02.009. URL https://www.sciencedirect.com/science/article/pii/S0957415818300321.

Chen Wang, Feng Zhang, and Shuzhi Sam Ge. A comprehensive survey on 2D multi-person pose estimation methods. *Engineering Applications of Artificial Intelligence*, 102:104260, June 2021. ISSN 0952-1976. doi: 10.1016/j.engappai.2021.104260. URL https://www.sciencedirect.com/science/article/pii/S095219762100107X.

Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional Pose Machines. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4732, Las Vegas, NV, USA, June 2016. IEEE. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.511. URL http://ieeexplore.ieee.org/document/7780880/.

World Health Organization. World stroke day 2022, 2022. URL https://www.who.int/srilanka/news/detail/29-10-2022-world-stroke-day-2022. Last accessed 17 october 2023.

World Stroke Organization. Wso global stroke fact sheet 2022, 2022. URL https://www.world-stroke.org/news-and-blog/news/wso-global-stroke-fact-sheet-2022. Last accessed 17 october 2023.

Ying Xing, Si-Dong Yang, Fang Dong, Man-Man Wang, Ya-Shuo Feng, and Feng Zhang. The beneficial role of early exercise training following stroke and possible mechanisms. *Life Sciences*, 198:32–37, April 2018. ISSN 0024-3205. doi: 10.1016/j.lfs.2018.02.018. URL https://www.sciencedirect.com/science/article/pii/S0024320518300687.

Young Youl You, Jin Gang Her, Ji Hea Woo, Taesung Ko, and Sin Ho Chung. The effects of stretching and stabilization exercise on the improvement of spastic shoulder function in hemiplegic patients. *Journal of Physical Therapy Science*, 26:491, 2014. ISSN 09155287.

doi: 10.1589/JPTS.26.491. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3996406/.

Matteo Zago, Matteo Luzzago, Tommaso Marangoni, Mariolino De Cecco, Marco Tarabini, and Manuela Galli. 3d tracking of human motion using visual skeletonization and stereoscopic vision. *Frontiers in Bioengineering and Biotechnology*, 8:508997, 3 2020. ISSN 22964185. doi: 10.3389/fbioe.2020.00181.

Zhengyou Zhang. Microsoft Kinect Sensor and Its Effect. *IEEE Multimedia*, 19(2):4–10, February 2012. ISSN 1070-986X. doi: 10.1109/MMUL.2012.24. URL http://ieeexplore.ieee.org/document/6190806/.

Huiyu Zhou and Huosheng Hu. Human motion tracking for rehabilitation—a survey. *Biomedical Signal Processing and Control*, 3:1–18, 1 2008. ISSN 1746-8094. doi: 10.1016/J.BSPC.2007.09.001. URL https://www.sciencedirect.com/science/article/pii/S1746809407000778?via%3Dihub.

Yves Zimmermann, Alessandro Forino, Robert Riener, and Marco Hutter. ANYexo: A Versatile and Dynamic Upper-Limb Rehabilitation Robot. *IEEE Robotics and Automation Letters*, 4(4):3649–3656, October 2019. ISSN 2377-3766, 2377-3774. doi: 10.1109/LRA.2019.2926958. URL https://ieeexplore.ieee.org/document/8755457/.