Early hand rehabilitation with robotic manipulators

by

Pablo Isaí Núñez Isáis

Thesis submitted as partial requirement for a

MSc degree in Computer Science

at

Instituto Nacional de Astrofísica, Óptica y Electrónica
Santa María Tonantzintla, 72840, Puebla, México

Supervised by:

Luis Enrique Sucar Succar, Ph.D., INAOE

Iván Alejandro Gutiérrez Giles, Ph.D., INAOE

Acknowledgements

I would like to express my sincere thanks to my thesis director and co-director, Dr. Luis Enrique Sucar Succar and Dr. Iván Alejandro Gutiérrez Giles, for their continuous guidance, support and encouragement during the development of this project. Their combined experience, patience and dedication were instrumental in helping me grow both academically and personally. I especially appreciate the way they guided my knowledge based on their experience, skills and wisdom, and for the time and effort they invested in helping me succeed.

I am also grateful to the members of the review committee: Dra. Kelsey Alejandra Ramírez Gutiérrez, Dr. Saúl Eduardo Pomares Hernández and Dr. Carlos Alberto Reyes García, for their valuable time, insightful comments and constructive criticism that enriched the quality of this work.

To Instituto Nacional de Astrofisica, Optica y Electronica (INAOE), for allowing me to use their facilities and financial support.

To Secretaría de Ciencias, Humanidades, Tecnología e Innovación (Secihti), for financial support (1197660) used for the implementation of this research project.

Sincere thanks to my family and with affection to Victoria, for their unconditional support, love and encouragement. Their confidence in me has been a constant source of strength and motivation.

Finally, I extend my thanks to all my friends, mentors and colleagues who have been part of this journey, offering me their help, advice and company throughout these years.

Abstract

This study addresses the challenge posed by the limited availability of traditional hand rehabilitation therapy for post-stroke patients. We propose a sustainable and intensive therapeutic approach through the design, development, and selection of a general-purpose robotic manipulator, a safe control system, and the application of innovative robotic therapeutic strategies. Simultaneously, the system allows adaptation to hands of any size and shape. The system was evaluated by validating the safety skills. This validation was performed through multiple experimental tests to ensure that none of the skills exceeded the safety parameters, thus avoiding damage directly to the patient and the system during the hand manipulation task. The goal was to evaluate the system's ability to perform hand movement rehabilitation safely. Skills assessed included: limiting range of motion, limiting the safe manipulation area and aligning the virtual environment with the real environment. Aligning hand orientation and position, using information from the depth camera and the MediaPipe estimation tool. Limiting physical interaction, detecting the presence of collisions and the minimum pressure required for fine hand manipulation. Limiting motion energy, using a control method to obtain smooth and safe motion during fine hand manipulation. And adding an emergency stop system capable of stopping the manipulator trajectory at any time.

Resumen

Este estudio aborda el reto que plantea la limitada disponibilidad de la terapia tradicional de rehabilitación de la mano para pacientes que han sufrido una apoplejía. Proponemos un enfoque terapéutico sostenible e intensivo mediante el diseño, el desarrollo y la selección de un manipulador robótico de uso general, un sistema de control seguro y la aplicación de estrategias terapéuticas robóticas innovadoras. Simultáneamente, el sistema permite la adaptación a manos de cualquier tamaño y forma. El sistema se evaluó validando las aptitudes de seguridad. Esta validación se realizó a través de múltiples pruebas experimentales para garantizar que ninguna de las habilidades superaba los parámetros de seguridad, evitando así daños directos al paciente y al sistema durante la tarea de manipulación de la mano. El objetivo era evaluar la capacidad del sistema para realizar la rehabilitación del movimiento de la mano de forma segura. Las habilidades evaluadas incluyeron: limitar el rango de movimiento, limitar el área de manipulación segura y alinear el entorno virtual con el entorno real. Alineación de la orientación y posición de la mano, utilizando información de la cámara de profundidad y la herramienta de estimación MediaPipe. Limitación de la interacción física, detectando la presencia de colisiones y la presión mínima necesaria para la manipulación fina de la mano. Limitación de la energía de movimiento, utilizando un método de control para obtener un movimiento suave y seguro durante la manipulación fina de la mano. Y añadiendo un sistema de parada de emergencia capaz de detener la trayectoria del manipulador en cualquier momento.

Table of Contents

Acknowledgements **II**

Abstract **III**

Resumen **IV**

Table of Contents **VII**

List of Figures **XIII**

List of Tables **XIV**

1 Introduction **1**

1.1 Motivation 2

1.2 Justification 2

1.3 Objectives 3

1.3.1 General objective 3

1.3.2 Specific objectives 3

1.4 Methodology 4

1.5 Contributions 4

1.6 Thesis structure 5

2 Theoretical Framework **6**

2.1 General purpose manipulators 6

2.1.1 Kinova Jaco 7

2.2 Kinect Azure 8

2.3 Software Framework 9

2.3.1 ROS 9

2.3.2 RViz 10

2.3.3 OpenCV 10

2.3.4 Mediapipe 11

2.3.5 PCL 11

2.3.6 MoveIt 12

2.3.7 Grasp Pose Detection 13

2.4 Rehabilitation treatments 13

2.4.1 Passive Therapy 14

2.4.2 Active Therapy 14

2.4.3 Bilateral Therapy 15

2.4.4 Common motions in upper limbs 15

2.5 Summary 16

3 Related Work 18

3.1 Hand pose estimation 18

3.2 Collaborative Robots 20

3.3 Robotic rehabilitation systems 21

3.3.1 Exoskeletons 22

3.3.2 End-effector 24

3.4 Safety in robotic rehabilitation systems 26

3.5 Summary 27

4 Proposed hand rehabilitation system 28

4.1 Hand Pose Recognition 29

4.2 Kinova Arm Planning and Control 30

4.3 Safety Skills 32

4.3.1 Limit Range Manipulator Motion 34

4.3.2 Restraining Physical Interaction and Energy 38

4.3.2.1 Limit Restraining Energy 38

4.3.2.2 Limit Physical Interaction Energy 40

4.3.3 Orientation Alignment 41

4.3.4 Stop Emergency System 43

4.4 Experimental Validation Model 44

4.5 Summary 45

5 Experiments and Results **46**

5.1 Limit manipulator range of motion validation 47

5.2 Limit Physical Interaction Energy validation 56

5.3 Orientation Alignment validation 64

5.4 Limit Motion Energy validation 72

5.5 Stop Emergency System validation 75

5.6 Summary 76

6 Conclusions and Future Work **78**

6.1 Limitations 79

6.2 Future Work 79

Bibliography **80**

List of Figures

1.1 Performance of traditional hand rehabilitation therapy by a therapeutic specialist. 2

2.1 Kinova Jaco's degrees of freedom: one degree of freedom for each manipulator joint, where each has its own coordinate axis. Image taken from Zohour, Belzile, and St-Onge (2021). 8

2.2 Representative diagram of the Kinect Azure developed by Microsoft. . . . 9

2.3 Depth image capture by the Kinect Azure. 9

2.4 The hand landmarker model bundle contains a palm detection model and a hand landmarks detection model. 11

2.6 Schematic diagram of a) shoulder flexion-extension, abduction-adduction, and internal-external rotation, b) elbow flexion-extension, c) forearm supination-pronation, d) wrist flexion-extension and ulnar-radial deviation. Image taken from Narayan, Kalita, and Dwivedy (2021) 15

2.5 High-level system architecture for the primary node provided by MoveIt called move_group. This node serves as an integrator: pulling all the individual components together to provide a set of ROS actions and services for users to use. 17

3.1 Comparison of distances in joints a) Euclidean distance between joints b) Geodesic distance. Image taken from Esquivel-Alvarado (2018) 19

3.2 A 2D evidence estimation example. (a) input image x, (b) ground-truth 2D segmentation mask m of x, (c) 2D skeletal position heat map of the finger tip of middle finger overlaid on x, and (d) masked image x $\odot$ m. Image taken from Baek, Kim, and Kim (2019) 20

3.3 Standards hierarchy. Image taken from Kóczi and Sárosi (2022) 21

3.4 Image of a subject using HAHRR. Image taken from Xie et al. (2021) . . 23

3.5 The exoskeleton is composed by three joint configurations: 90-degrees joints, 180-degrees joints and axial joints. Together, the joints produce an exoskeleton structure that achieves full G-H, elbow, and wrist functionality. Image taken from Perry, Rosen, and Burns (2007) 24

3.6 Subjects performing (left) unilateral and (right) bilateral movements with Mirror Image Movement Enabler system. Image taken from Lum et al. (2006) 25

4.1 Operation diagram of the robotic system for hand rehabilitation, consisting of the robotic manipulator that interacts directly with the patient and through constant monitoring of the manipulator signals and patient feedback, sends the information to the adaptive control system based on safety skills, responsible for limiting, controlling and adjusting the parameters of the robotic manipulator to perform the safe manipulation. 29

4.2 Color image (left) and depth image (right), after MediaPipe processing to segment the 21 relevant hand joints. 30

4.3 move_group node diagram representation. 31

4.4 Planning Scene Monitor node diagram representation. 32

4.5 Overview of identified hazards, influencing factors, potential injuries and related safety skills. Note that this list is not extensive and additional relevant hazards and safety skills might be identified in the future. Image taken from Bessler et al. (2021) 33

4.6 Safe working area to avoid collisions with objects outside the interaction environment for rehabilitation delimited by green blocks within the virtual vision. 35
4.7 3D model of the hand segmented with respect to the point cloud. 35
4.8 Spatial representation of the position and orientation of the robot base and camera frame in a virtual environment. 36
4.9 Representation of the rotation performed by the position and orientation of the camera frame to approach the robot base. 36
4.10 Representation of the traslation performed by the position and orientation of the camera frame to approach the robot base. 37
4.11 Experimental relationship between the robot base frame and the camera after performing the rotation and translation process. 37
4.12 index finger point cloud segmented 38
4.13 Basic impedance control for robotic manipulators with one or more joints. The impedance control block receives information directly from the kinematics of the manipulator and its environment to compute the control law via inverse dynamics, finally adjusting the response of the robotic manipulator to perform fine and smooth hand manipulation. 39
4.14 Experimental example of pressure measurement through the pressure sensor attached to the 3D hand model. 40
4.15 Representation of the reference vectors used to determine the planar rotation angle of the hand with respect to the camera image. 41
4.16 Representative diagram of the arc-sine function to obtain the planar rotation angle of the hand. 42
4.17 Reference pivot to determine the orientation of the manipulator concerning the hand orientation. 43
4.18 Reference box, obtained by joining the fingertip joints and the interproximal phalanges of the index and middle fingers. 43

4.19 Control diagram for the rehabilitation system with the emergency stop module implemented. 44

4.20 Experimental Model in charge of performing the security tests to validate the security protocol. 45

5.1 Aruco code generator, capable of customizing the number of markers on the X, Y axis, their size, spacing, among other parameters to have a specific code for the required calibration task. 48

5.2 Result of approximation to the middle phalanx of the 3D model on the X-axis using the standard "Hand-Eye" calibration method. 49

5.3 Result of approximation to the middle phalanx of the 3D model on the Y-axis using the standard "Hand-Eye" calibration method.. 49

5.4 Result of approximation to the middle phalanx of the 3D model on the Z-axis using the standard "Hand-Eye" calibration method.. 50

5.5 RGB image of the Kinect camera, where it can be appreciated the manual placement of the Aruco code directly on the manipulator's end effector, approximating the transformation matrices. 51

5.6 Visualization of the match between the transformed end-effector matrices and the aruco code, after manual modification of the parameters within the "Hand-Eye" calibration tool. 51

5.7 Result of approximation to the middle phalanx of the 3D model on the X-axis using the matrix transform as reference. 52

5.8 Result of approximation to the middle phalanx of the 3D model on the Y-axis using the matrix transform as reference. 52

5.9 Result of approximation to the middle phalanx of the 3D model on the Z-axis using the matrix transform as reference. 53

5.10 Point cloud segmentation of the middle phalanx of the 3D model visualized within Rviz with respect to the robotic manipulator. 54

5.11 Manual manipulator match with respect to the real position of the 3D model visualized through Rviz. 54

5.12 Result of approximation to the middle phalanx of the 3D model on the X-axis using the Point cloud segmented as reference. 55

5.13 Result of approximation to the middle phalanx of the 3D model on the Y-axis using the Point cloud segmented as reference. 55

5.14 Result of approximation to the middle phalanx of the 3D model on the Z-axis using the Point cloud segmented as reference. 56

5.15 Representative end-effector force magnitude graph of the 3 best results during the development of the "home-setup" trajectory without collisions. 58

5.16 Representative end-effector force magnitude graph of the 3 best results during the development of the "setup-calibration" trajectory without collisions. 58

5.17 Representative end-effector force magnitude graph of the 3 best results during the development of the "home-setup" trajectory with collisions. . . 59

5.18 Representative end-effector force magnitude graph of the 3 best results during the development of the "setup-manipulation" trajectory with collisions. 59

5.19 Diagram of the circuit designed for the characterization of the pressure sensor, composed of two resistors, one with an experimental value of 100Ω, 1KΩ, 10KΩ, 100KΩ and 1MΩ and the other resistance, variable, is a strain gauge that converts the surface pressure into resistance. 60

5.20 Characterization graph of the pressure sensor, comparing the different results of the constant resistances to a maximum pressure exerted manually on the strain gauge. 61

5.21 Graph of the maximum voltage obtained by the pressure sensor as a result of the maximum possible pressure applied by the manipulator. 62

5.22 Graph of the maximum current obtained by the end-effector fingers as a result of the maximum possible pressure applied by the manipulator. . . . 63

5.23 Graph of voltages obtained by the pressure sensor during the development of fine hand manipulation. 63
5.24 Graph of the currents obtained by the terminal effector fingers 64
5.25 Hand rotation at 0 degrees 66
5.26 Hand rotation at 90 degrees 66
5.27 Hand rotation at -90 degrees 67
5.28 Hand rotation at 180 degrees 67
5.29 Downward fingertips orientation 68
5.30 Upward fingertips orientation 68
5.31 Pinky upward orientation 69
5.32 Thumb upward orientation 69
5.33 Reference position of the index finger with an inclination of 0 degrees. . . 70
5.34 Reference position of the index finger with an inclination of 10 degrees. . 70
5.35 Reference position of the index finger with an inclination of 20 degrees. . 70
5.36 Reference position of the index finger with an inclination of 30 degrees. . 71
5.37 Force plot during fine hand control trajectory using low-level control. . . . 73
5.38 Force plot during fine hand control trajectory using PID control. 74
5.39 Force plot during fine hand control trajectory using impedance control through Moveit. 75

List of Tables

5.1 Table of results for the different inclinations of the test finger 72

5.2 Table of results during general trajectory interruption 76

Chapter 1

Introduction

A cerebrovascular accident, commonly known as a stroke, ranks as the third leading cause of death globally and the primary cause of disability among older adults (Thayabaranathan et al., 2022). These incidents occur suddenly, either as ischemic or hemorrhagic episodes, directly affecting blood circulation in brain tissues and leading to partial destruction. The damage impacts the cerebral cortex, including the sensorimotor areas of the brain. Consequently, traditional rehabilitation systems have played a significant role in closely monitoring and addressing such cases.

Robotic rehabilitation systems have emerged as pivotal tools in recovering and enhancing motor skills for patients with neurological or muscular impairments. They complement traditional rehabilitation methods, proving valuable aids in the rehabilitation process (Bertani et al., 2017). Recent technological advancements in robotics have expanded possibilities for improving the quality of life, particularly in rehabilitation. The increasing prominence of industrial collaborative robotic systems designed for specific tasks demanding precision marks a technological milestone. These robots have robust safety features to ensure their safe operation in spaces shared with human operators. This technological convergence has enabled rehabilitation robotic systems to leverage the interaction protocols developed for collaborative systems, thereby expanding their capabilities and applications in rehabilitation.

1.1 Motivation

The increasing number of stroke patients each year has underscored the limitations of traditional rehabilitation therapies. The shortage of specialized rehabilitation therapists has resulted in an overwhelming workload within rehabilitation centres, making it challenging to provide optimal and personalized care to all patients in need. This situation highlights the urgent need to seek innovative solutions that can effectively address the growing challenge of delivering high-quality rehabilitation therapies to an expanding number of stroke patients. Traditional rehabilitation therapy involves a personal interaction between the therapeutic specialist and the patient as shown in the figure 1.1. In this context, robotic rehabilitation systems emerge as a promising tool to complement and improve the effectiveness of traditional therapies, offering a viable alternative to meet the growing demand for rehabilitation services by implementing robotic manipulators that mimic traditional rehabilitation therapies.

Figure 1.1: Performance of traditional hand rehabilitation therapy by a therapeutic specialist. Image taken from https://www.exceltherapy.com/2023/06/05/5-facts-about-hand-therapy/.

1.2 Justification

Humans rely on our knowledge and immediate response to stimuli to perform specific tasks. This ability is crucial for rehabilitation specialists manipulating patients' limbs using their hands and mechanical devices. The primary aim is to improve joint movement and reactivate motor function in areas affected by stroke.

Technological advancements in robotics have led to the development of systems capable of matching or surpassing human dexterity in performing specific tasks. For example, in the automotive industry, robots are used to assemble vehicles, in medicine, surgical robots assist in high-risk operations. These technological advances can be leveraged to develop robotic systems based on traditional rehabilitation therapies.

Robotic systems aim to replicate rehabilitation tasks and goals automatically without human intervention. They can enhance the effectiveness of traditional therapies by providing additional improvement without requiring constant therapist supervision.

1.3 Objectives

1.3.1 General objective

To design and develop a system capable of performing early rehabilitation of the hand, implemented with robotic manipulators considering safety and control aspects for fine manipulation.

1.3.2 Specific objectives

- To estimate and detect hand poses accurately within the system.
- To integrate a validation protocol to ensure the safety of the system.
- To plan the trajectory for fine control of robotic manipulators.
- To conduct tests on 3D-printed models equipped with sensors to validate the system's safety.

1.4 Methodology

The methodology developed is listed below:

- To explore the use of image processing tools for accurate hand pose estimation, aiming to implement algorithms for hand joint detection, tracking, and motion analysis.

- To implement precise trajectory planning in general-purpose manipulators, aiming for fine control of the hand to simulate the rehabilitation movement developed by rehabilitation specialists.

- To delve into the development of a safety protocol based on Bessler et al. (2021) and acquire safety skills that ensure a secure rehabilitation system for both the patient and the system environment.

- To develop an experimental scenario enabling the evaluation and validation of safety skills applied to a robotic manipulator.

- Experimentally evaluate and analyze each of the safety skills developed for fine manipulation of the hand individually.

- Identify security issues by experimentally validating the proposed protocol. Each validated skill is integrated with others in a secure and feedback-enhanced system

1.5 Contributions

The principal contributions of this thesis are:

- Based on the study of skills and safety aspects in robotic rehabilitation systems Bessler et al. (2021), this research introduces a novel security protocol for enabling safe hand manipulation explicitly designed for rehabilitation robotics applications. The proposed protocol represents an effort to ensure the integrity of the patient.

- We introduce a methodology based on computer vision, hand pose estimation algorithms, and deep learning-based estimation tools to estimate the manipulator's pose and ensure specific manipulation within the expected area of the hand. We aim to prevent alignment errors between the manipulator and the patient's desired hand pose.

1.6 Thesis structure

The rest of the thesis is structured as follows. Chapter 2 describes the theoretical concepts necessary to implement a hand rehabilitation system using general purpose manipulators. Chapter 3 consists of a review of the state of the art, mentioning the main related works in the area of rehabilitation with robotic systems. Chapter 4 describes the methodology followed for this work. Chapter 5 presents the experiments performed during this research and the results of the proposed protocol. Finally, Chapter 6 presents the conclusions and experimental analysis, together with future work in the rehabilitation area.

Chapter 2

Theoretical Framework

This chapter provides general information about the concepts necessary for the development of this research. These concepts describe the robotic manipulator, the depth camera, the robotic operational system in charge of controlling the different devices and modules implemented, and the libraries and tools necessary for the development of this thesis.

2.1 General purpose manipulators

The development of manipulators in robotics has been a significant milestone in the field, enabling precise and controlled manipulation of objects in various applications. One of the pioneering figures in this domain was Raymond Goertz, who, in the mid-20th century, designed and patented master-slave manipulators with advanced features like force feedback technology and teleoperation capabilities (Gasparetto & Scalera, 2019). These manipulators, such as the CRL Model 8, marked a crucial advancement in robotic technology, emphasizing the importance of haptic senses and bilateral coupling for delicate object manipulation. Furthermore, the industrial robotics landscape saw significant evolution with the introduction of the Unimate in 1961, the first industrial robot developed by George Devol and Joseph Engelberger (Moran, 2007). This hydraulic manipulator arm revolutionized manufacturing processes, particularly in the automotive industry, by automating tasks like metalworking and welding with high precision and efficiency. The Unimate's success paved the way for the widespread adoption of industrial robots, showcasing the potential of robotic manipulators in enhancing productivity and safety in industrial settings. It is a fact that the technology used in the development of robotic manipulators has increased the number of applications that can be given to these tools, allowing to expand the field of research and the areas where this type of robotic systems can be used, where the common goal is to develop manipulators capable of performing more tasks that are complex for us humans.

2.1.1 Kinova Jaco

The Kinova Jaco manipulator is a cutting-edge robotic arm renowned for its advanced capabilities in various applications. Developed by Kinova Robotics, the Jaco manipulator stands out for its versatility and precision, making it a valuable tool in research and industry. The design of the Kinova Jaco has 6 degrees of freedom and two or 3-fingered end effectors, which enable complex and delicate manipulation tasks. Figure 2.1 shows a representation of the six degrees of freedom of the Jaco arm. Furthermore, the research underscores the Kinova Jaco's integration with control systems like PID for enhanced motion control and trajectory planning, showcasing its efficiency and adaptability in diverse environments.

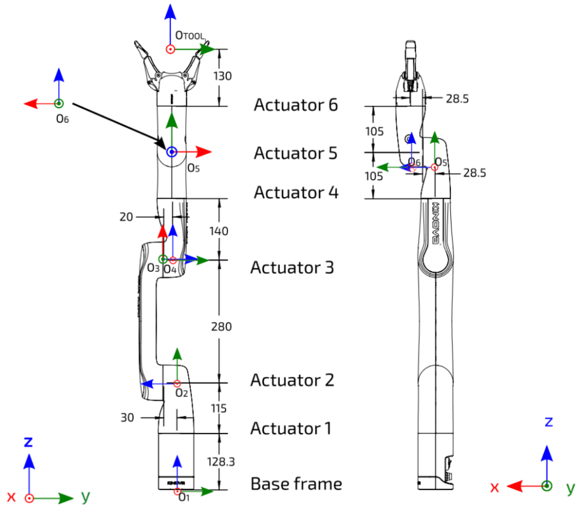


Figure 2.1: Kinova Jaco's degrees of freedom: one degree of freedom for each manipulator joint, where each has its own coordinate axis. Image taken from Zohour et al. (2021).

2.2 Kinect Azure

Developed by Microsoft, the Kinect Azure sensor (figure 2.2) is equipped with cutting-edge depth-sensing technology, enabling precise and detailed depth perception in real-time scenarios. Scientific articles (Han et al., 2024; Tölgyessy, Dekan, Chovanec, & Hubinskỳ, 2021) highlight the Kinect Azure's ability to accurately capture high-resolution depth images (figure 2.3), making it a valuable tool for 3D mapping, object recognition, and navigation tasks in robotics. Moreover, the integration of Kinect Azure with the Robot Operating System (ROS) has opened up new possibilities for robotic applications. Research emphasizes the seamless compatibility of Kinect Azure with ROS, enabling efficient data processing, sensor fusion, and integration with robotic platforms for enhanced perception and decision-making capabilities.

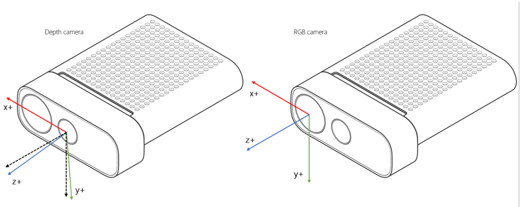


Figure 2.2: Representative diagram of the Kinect Azure developed by Microsoft. Image taken from https://learn.microsoft.com/es-es/azure/kinect-dk/coordinate-systems

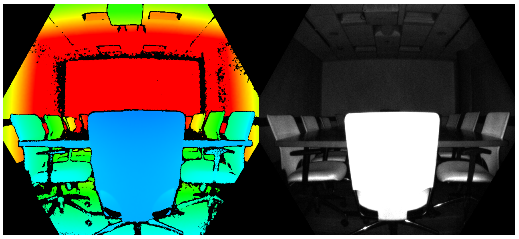


Figure 2.3: Depth image capture by the Kinect Azure. Image taken from https://learn.microsoft.com/es-es/azure/kinect-dk/depth-camera

2.3 Software Framework

The main framework of this work is based on ROS to develop the appropriate communication between the Kinect, the Jaco manipulator and the software tools or libraries needed to develop the thesis project.

2.3.1 ROS

The Robot Operating System (ROS) (Quigley et al., 2009) is a versatile framework designed to assist developers in creating robotic applications. It offers many features, including hardware abstraction, device drivers, libraries, visualization tools, message-passing capabilities, package management, graphical interfaces, and compilers. ROS is organized into packages containing a collection of nodes and other files that serve specific purposes. Nodes, which are executable files, communicate with each other using ROS client libraries. They can publish or subscribe to topics, allowing for the simultaneous execution of various processes. ROS is licensed under an open-source BSD license, fostering a vibrant community that continuously contributes open-source modules for others to utilize. For the general purpose of this research it was developed in the Ubuntu 20.04 LTS environment with the ROS Noetic version.

2.3.2 RViz

RViz (ROS Visualization) is a powerful 3D visualization tool widely used in robotics for visualizing sensor data, robot models, and various algorithms in a simulated environment. Developed as part of the Robot Operating System (ROS), RViz provides a user-friendly interface for displaying and interacting with complex robotic systems. This versatile tool allows users to visualize robot configurations, sensor data, and navigation paths in real time, aiding in developing, testing, and debugging robotic applications. RViz offers a range of features such as interactive markers, point clouds, and robot models, enabling users to visualize and analyze data from sensors like LiDAR, cameras, and depth sensors. Its integration with ROS facilitates seamless communication with robotic components, making it an essential tool for robot designers, researchers, and developers.

2.3.3 OpenCV

OpenCV (Open Source Computer Vision Library) is a widely-used open-source computer vision and machine learning software library that has become a fundamental tool in image and video processing. Developed in C++ with bindings for Python, Java, and other languages, OpenCV provides a comprehensive set of algorithms and functions for a wide range of computer vision applications. One of the critical strengths of OpenCV is its ability to process images and videos in real time, making it highly suitable for applications that require fast and efficient image analysis. The library offers many features, including image and video I/O, image processing, feature detection, object recognition, and machine learning algorithms (Bradski & Kaehler, 2015).

2.3.4 Mediapipe

MediaPipe is a versatile framework developed by Google (Lugaresi et al., 2019) that offers a wide range of solutions for multimedia processing tasks, particularly in the realm of computer vision and machine learning. This open-source framework provides developers with tools to build applications that analyze and process media data efficiently. MediaPipe Hands is a remarkable solution that facilitates hand detection, figure 2.4 shows the 21 3D reference points providing for each hand, including finger positions and palm location. These solutions leverage machine learning models to deliver precise and efficient results for hand tracking and gesture recognition tasks.

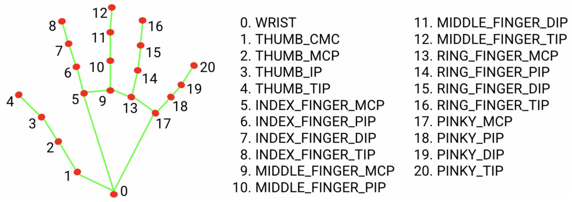


Figure 2.4: The hand landmarker model bundle contains a palm detection model and a hand landmarks detection model. Image taken from https://developers.google.com/mediapipe/solutions/vision/hand_landmarker.

2.3.5 PCL

Point Cloud Library (PCL) is a robust open-source framework for 2D/3D image and point cloud processing in robotics, computer vision, and augmented reality (Rusu & Cousins, 2011). Developed in C++ with bindings for Python, PCL offers a comprehensive set of algorithms and tools for processing, filtering, segmentation, registration, and visualization of point cloud data. This powerful library enables researchers, developers, and enthusiasts to efficiently work with point cloud data, making it a valuable resource for various applications. PCL's extensive feature set includes algorithms for point cloud registration, surface reconstruction, feature extraction, and object recognition, allowing users to perform complex operations on point cloud data efficiently. The library also integrates with other popular libraries and frameworks, such as OpenCV and ROS, enhancing its capabilities and interoperability in various projects.

2.3.6 MoveIt

MoveIt was developed as an open-source project to create a comprehensive platform for robotic manipulation, grounded in the Robot Operating System (ROS) (Grushko et al., 2020). The primary aim of MoveIt is to provide a versatile and robust suite of tools designed to facilitate various aspects of robotic manipulation, including motion planning, manipulation, kinematics, control, collision detection, and navigation (figure 2.5). As a platform, MoveIt integrates advanced algorithms and techniques to enable efficient and precise motion planning, allowing robots to navigate complex environments and perform intricate tasks with a high degree of autonomy. The motion planning capabilities of MoveIt include collision detection, path optimization, and real-time adjustments, ensuring that the robot can operate safely and effectively in dynamic settings. Moreover, MoveIt supports a wide range of robotic hardware, making it adaptable to different types of robots and applications. The kinematics tools within MoveIt provide solutions for both forward and inverse kinematics, enabling precise control over the robot's movements and ensuring that the end-effector can reach desired positions accurately. Control and navigation are also critical components of MoveIt. The platform offers interfaces for integrating various control strategies, from simple position control to more complex impedance and force control. Additionally, navigation tools allow robots to move through their environment, avoiding obstacles and reaching target locations efficiently. Overall, MoveIt stands out as a powerful and flexible platform for robotic manipulation, widely adopted in academic research and industrial applications. Its open-source nature encourages continuous development and collaboration, driving advancements in robotics and expanding the possibilities for automated systems.

2.3.7 Grasp Pose Detection

The GPD repository on GitHub (https://github.com/atenpas/gpd) is a tool for detecting 6-DOF (6 degrees of freedom) grasp poses in 3D point clouds. It is designed to work with a 2-finger robot hand, such as a parallel jaw gripper. The main strengths of GPD are: It can work with novel objects without requiring CAD models for detection. It can operate in dense clutter environments. It outputs 6-DOF grasp poses, enabling more than just top-down grasps. The GPD process involves two main steps: 1) Sampling a large number of grasp candidates, and 2) Classifying these candidates as viable grasps or not. The repository provides detailed instructions on requirements, installation, generating grasps for point cloud files, parameters, views, input channels for neural networks, CNN (Convolutional Neural Network) frameworks, network training, and more.

In this section, we found the different tools that will be used for the system's design, testing, and experimental validation. Each tool will be used together to obtain control and interaction between the manipulator, the hand point cloud, and the estimation of the hand's orientation and position.

2.4 Rehabilitation treatments

The therapist's task is to find the optimal procedure to treat the affected limb and this depends directly on the motor control that the patient has over the extra limb and the degree of complexity that the injury may present. The choice of this procedure is crucial to ensure the effectiveness of the treatment (Proietti, Crocher, Roby-Brami, & Jarrasse, 2016). Robotic therapy, thanks to its ability to perform repetitive and precise movements, has enabled the development of the 3 main types of rehabilitation: passive, active and bilateral.

2.4.1 Passive Therapy

Passive therapy requires no effort on the part of the patient and is generally used in the early stages after a stroke, especially when the affected limb is unresponsive. This therapy is usually prescribed to hemiplegic patients with paralysis on one side of the body. It consists of repeatedly moving the affected limb following a specific trajectory during the session, usually performed by a limited-range motion rehabilitation robot. The trajectory of the movement is carefully planned to avoid any harm to the patient or misalignment of the joints.

This treatment focuses on stretching and contracting the affected upper extremity and is also used to assess its range of motion. In the treatment, exoskeleton robots perform repetitive movements according to the range of motion. A clinical study on passive therapy involving three subjects showed that a 40-minute training session effectively reduced spasms and stiffness of the affected limbs (Ren, Kang, Park, Wu, & Zhang, 2012).

2.4.2 Active Therapy

This type of treatment is prescribed to patients who can move the affected limb to some extent, albeit with limitations. Active therapy refers to the ability to move the affected limb, although not efficiently. It can be classified into active-assistive therapy and active-resistive therapy. Active-assistive therapy involves the application of an external force by a therapist or robot to help the patient perform the assigned task and improve the range of motion. On the other hand, active-resistive therapy involves the application of an opposing force on the affected limbs provided by a therapist or robot. Studies show that patients' performance gradually improves when the opposing force is gradually increased, determined by an algorithm according to the patient's ability (Fasoli et al., 2004).

2.4.3 Bilateral Therapy

Bilateral therapy is based on the reflex principle during rehabilitation. In this therapy, the affected limb mimics the movement of the functional limb, giving the user full control over the affected limb. Devices such as the Mirror Image Movement Enabler (MIME) and other exoskeletons use this therapy model (Sheng, Zhang, Meng, Deng, & Xie, 2016).

2.4.4 Common motions in upper limbs

Regardless of the type of rehabilitation that can be applied, there are basic locomotion movements, both for the shoulder and wrist, which must be developed properly to help the patient improve these ranges of motion, can be seen in figure 2.6, the different types of basic movements for upper extremities.

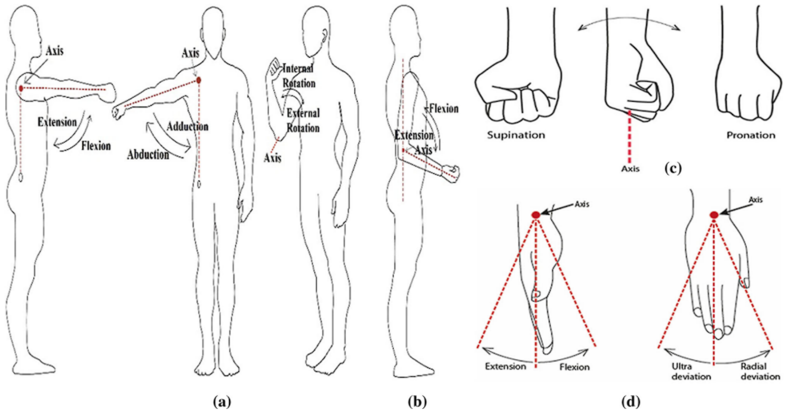


Figure 2.6: Schematic diagram of a) shoulder flexion-extension, abduction-adduction, and internal-external rotation, b) elbow flexion-extension, c) forearm supination-pronation, d) wrist flexion-extension and ulnar-radial deviation. Image taken from Narayan et al. (2021)

In this section we were able to learn about the different types of rehabilitation used for the recovery of mobility of the affected limbs, although there are differences between the degree of rehabilitation required depending on factors such as the limb, the degree of voluntary mobility of the patient, among others, these types of rehabilitation have the same purpose of promoting the recovery of voluntary mobility and improve the quality of life of people affected by an accident or disease that has significantly impaired the movement of the limb or limbs in question.

2.5 Summary

This chapter has provided the theoretical framework for understanding the proposed research and has offered a solid argumentation for the feasibility of the work. The hardware components used are described, as the main software tools for hand pose detection, estimation, processing, and analysis, and the libraries used to develop this thesis. The differences between traditional rehabilitation methods are recognized, taking into account that there are mainly 3 types: active, passive, and bilateral, and it is primarily expected that this project will focus on the passive type of rehabilitation, to evaluate an early stage of hand rehabilitation in patients with low mobility.

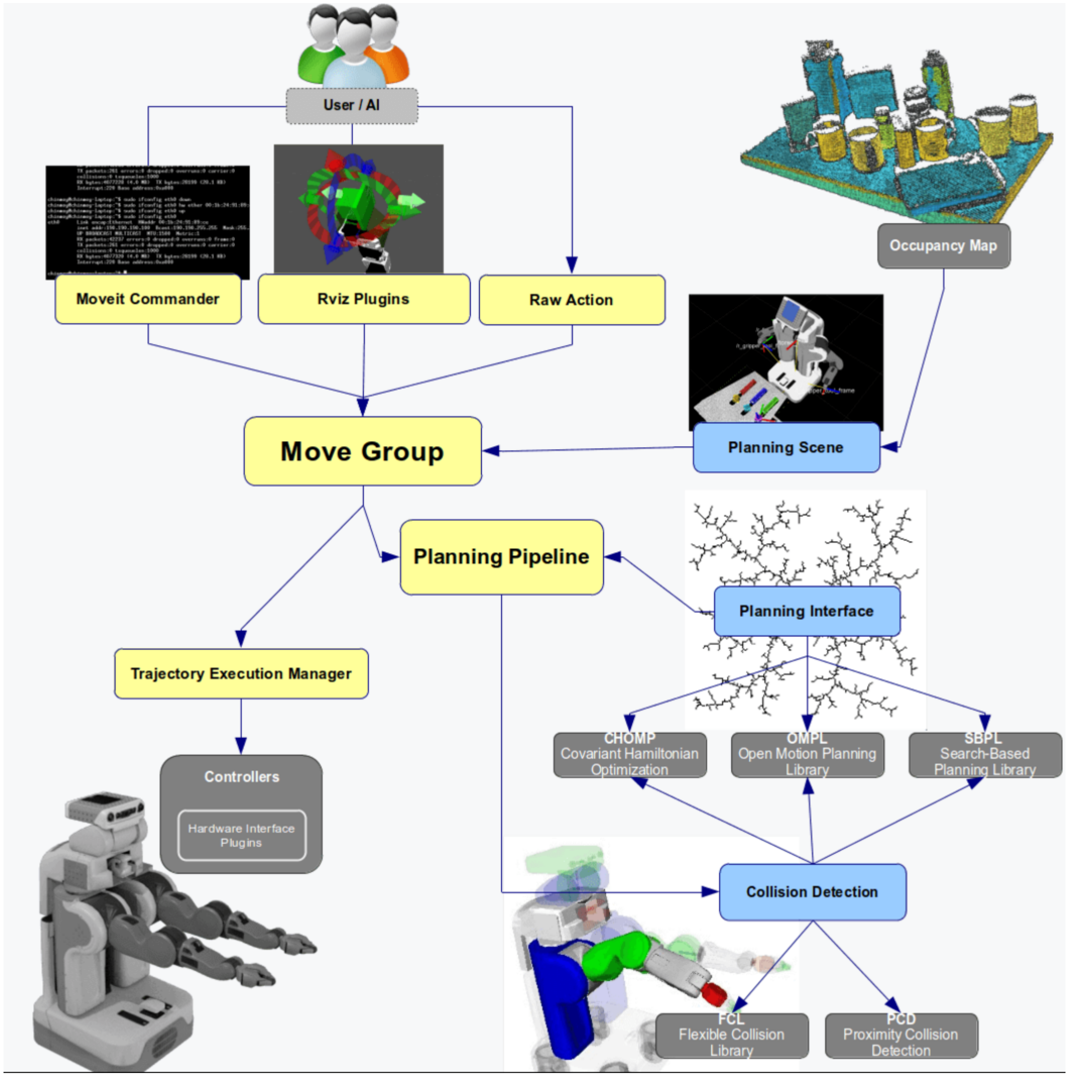

User / AI
Moveit Commander
Rviz Plugins
Raw Action
Occupancy Map
Planning Scene
Move Group
Planning Pipeline
Planning Interface
Trajectory Execution Manager
Controllers
Hardware Interface Plugins
CHOMP
Covariant Hamiltonian Optimization
OMPL
Open Motion Planning Library
SBPL
Search-Based Planning Library
Collision Detection
FCL
Flexible Collision Library
PCD
Proximity Collision Detection

Figure 2.5: High-level system architecture for the primary node provided by MoveIt called move_group. This node serves as an integrator: pulling all the individual components together to provide a set of ROS actions and services for users to use. Image taken from https://moveit.ros.org/documentation/concepts/.

Chapter 3

Related Work

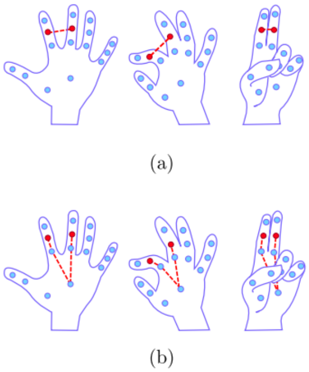
This chapter provides general information on related work on hand pose estimation, a brief introduction to collaborative robots, and the different robotic rehabilitation systems currently under investigation, including advances in protocols and safety systems within these novel rehabilitation systems using robotic elements.

3.1 Hand pose estimation

The estimation of hand pose is a crucial area of research in computer vision and human-machine interaction, often examined within the broader scope of related work. Over the years, researchers have explored various techniques and approaches to tackle the complex challenge of accurately determining the pose of the hand in different contexts. These efforts have yielded significant advancements, contributing to a rich landscape of methodologies and algorithms. Within this realm of related work, numerous methodologies have been developed to address the diverse needs of applications related to hand pose estimation.

One of the key references in hand pose estimation is the survey paper by Chen et al. (2020). This work thoroughly reviews the different techniques used for hand pose estimation, including wearable sensor-based and computer-vision-based methods. The survey discusses the evolution of hand pose recognition and sign language recognition, highlighting the advancements enabled by depth-capturing devices and cost-effective computational systems.

Some references delve into specific methodologies for hand pose estimation, such as Esquivel-Alvarado (2018), which uses multiple cues for hand tracking and model refinement based on hand topology to estimate the hand pose (figure 3.1). The approach recognizes that the joints in the human hand are not independent but have a hierarchical structure and topological relationships. The method involves a training process that learns to predict the 3D positions of the hand joints, taking into account the topological constraints and relationships between the joints.



(a)

(b)

Figure 3.1: Comparison of distances in joints a) Euclidean distance between joints b) Geodesic distance. Image taken from Esquivel-Alvarado (2018)

The key innovation of this work is the explicit modelling of the topological relationships between hand joints, which is leveraged to improve the accuracy and efficiency of 3D hand pose estimation compared to approaches that treat the hand joints as independent entities. This hierarchical and anatomically-informed approach is intended to make the pose estimation more robust and reliable.

As explained in Baek et al. (2019) incorporates a CNN-based framework for reconstructing hand poses and shapes in a 3D model, using individual RGB images as input. The dense hand poses estimation (DHPE) network they propose is decomposed into the 2D evidence estimator, the 3D mesh estimator, and the projector. An example of the 2D evidence estimator is presented in Figure 3.2 where the texture extracted from the camera viewpoint is generated through the segmented 2D mask. This paper discusses the challenges posed by the loss of information, in addition to using only RGB images as input information to reconstruct the hand pose.

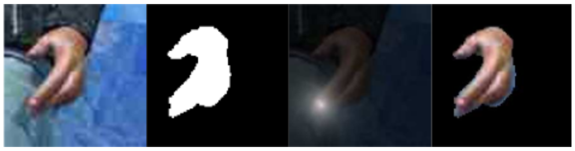


Figure 3.2: A 2D evidence estimation example. (a) input image x, (b) ground-truth 2D segmentation mask m of x, (c) 2D skeletal position heat map of the finger tip of middle finger overlaid on x, and (d) masked image x $\odot$ m. Image taken from Baek et al. (2019)

It also highlights the experimental results of applying the proposed approach to sequences with articulation and occlusion, demonstrating the method's effectiveness in estimating hand poses from RGB video data.

3.2 Collaborative Robots

Collaborative robots, or cobots, are designed to work alongside humans, enabling direct physical interaction. These robots have safety features that allow them to operate near humans without physical barriers. The primary focus of cobots is to ensure the safety of human-robot interaction by minimizing risks associated with collisions, unwanted contacts, and human manipulation. Safety standards are crucial in protecting personnel working with cobots, ensuring minimal interference with operational efficiency. Cobots are characterized by their light structure, compact design, and slower movements compared to traditional robots, enhancing safety in shared workspaces. Integrating sensors like cameras, force-torque, and tactile sensors enables cobots to swiftly detect and respond to potential hazards, making interaction with humans intuitive and safe.

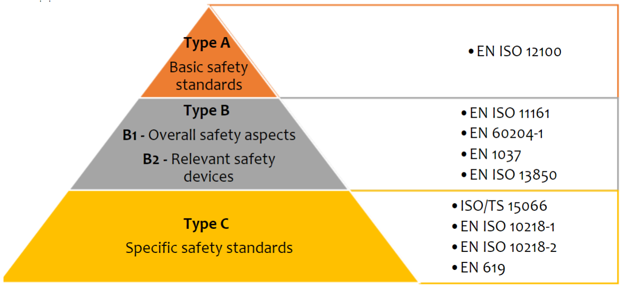


Figure 3.3: Standards hierarchy. Image taken from Kóczi and Sárosi (2022)

Safety issues in cobots are addressed through rigorous risk assessments and adherence to safety standards. The International Organization for Standardization (ISO) has developed technical specifications like ISO/TS 15066 (Specific safety standard: Type C, according to Figure 3.3) to guide the safe collaboration between humans and robots. These standards emphasize that contact between cobots and humans should not result in pain or injury, highlighting the importance of maintaining a safe working environment. The goal is to enhance safety while promoting efficiency and ease of use in collaborative robot applications.

3.3 Robotic rehabilitation systems

Robotic rehabilitation systems are advanced devices designed to aid in diagnosing and treating health conditions affecting the musculoskeletal and nervous systems. These systems play a crucial role in helping individuals regain lost functions by activating dormant nerve cells. By utilizing cutting-edge technology, robotic rehabilitation systems reduce disorders, improve functions, and rehabilitate various neurological and orthopaedic diseases. These systems provide intensive and comprehensive neurorehabilitation, positively impacting the reorganization of the nervous system and assisting patients in regaining functionality at the highest possible level. Robotic rehabilitation involves using adjustable and programmable robots tailored to each patient's specific needs. These systems incorporate reinforced orthoses with computer-controlled motors to support joint movement, virtual environments for visual feedback, and body weight support systems for balance training. Robotic rehabilitation can be initiated at any stage of a disease, but starting treatment early has been shown to enhance recovery rates, increase independence in daily activities, and reduce post-disease dysfunction. Implementing new graphical interfaces, ecological scenarios, and cognitively demanding tasks in robotic therapy engages patients actively, maintaining their motivation and desire to participate in the treatment process.

3.3.1 Exoskeletons

Exoskeletons are wearable robotic devices designed to assist and augment human physical capabilities. They are instrumental in rehabilitation applications, where they can help individuals regain lost functions and improve their quality of life. Exoskeletons can be categorized into two main types: upper-limb exoskeletons and lower-limb exoskeletons. Upper-limb exoskeletons, like the one used in Xie et al. (2021), are designed to support and assist the arm, hand, and shoulder movements. They are crucial in rehabilitation, as they can help patients recover from neurological or orthopaedic conditions that impair their ability to perform everyday tasks involving the upper extremities. These exoskeletons are often integrated with advanced control systems, sensors, and actuators to provide personalized assistance and promote active participation from the user.

The HAHRR (Hybrid Arm-Hand Rehabilitation Robot) integrates a cable-driven module for assisting arm motion and a hand exoskeleton for assisting hand motion, represented in Figure 3.4. Using an exoskeleton for the hand is crucial, as, without hand motion, people cannot perform critical daily activities like eating, drinking, writing, or typing. Similarly, people can only lift or manipulate objects they grasp with arm motion.

Previous attempts have been made to integrate exoskeleton systems for the shoulder, elbow and hand wrist to assist in reach-and-grasp tasks. However, these

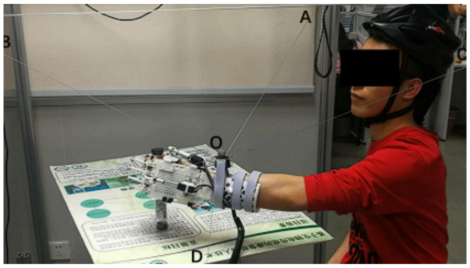


Figure 3.4: Image of a subject using HAHRR. Image taken from Xie et al. (2021)

approaches relied on predefined trajectories and semi-automatic control, needing more continuous voluntary user participation. In contrast, the HAHRR in this study allows for continuous voluntary participation by using an EMG-based admittance control framework, which continuously perceives the user's motion intention from EMG signals and translates it into control commands for the robot.

Another approach considering the total upper limb impairment control is proposed in Perry et al. (2007), in which the mechanical and anthropomorphic joint configurations necessary for the exoskeleton to achieve full functionality are identified. This approach involves modelling the human arm and its joints, including the shoulder, elbow, forearm, and wrist, to accurately replicate the movements and rotations required for daily activities. The exoskeleton presented in Figure 3.5 is designed to mimic these movements, ensuring that it can effectively assist individuals in performing tasks that involve the upper limb. The methodology used in this study involves designing a powered upper-limb exoskeleton that accurately replicates the movements and rotations of the human arm. The exoskeleton comprises seven single-axis revolute joints, each designed to mimic the specific movements and rotations of the corresponding human joints. The joints are classified into three configurations: 90-degree, 180-degree, and axial.

The exoskeleton's mechanical design involves the use of cable routing and pulleys to achieve the desired range of motion, and the study aims to place the singularity in an unreachable or near-unreachable location to ensure that the exoskeleton can achieve the desired range of motion without compromising its functionality.

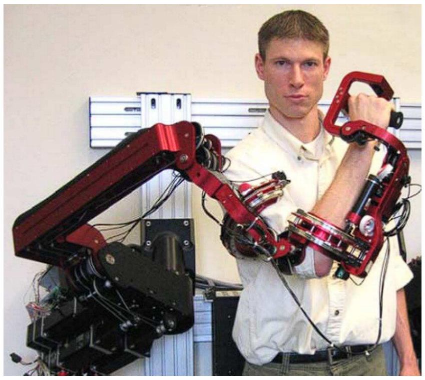


Figure 3.5: The exoskeleton is composed by three joint configurations: 90-degrees joints, 180-degrees joints and axial joints. Together, the joints produce an exoskeleton structure that achieves full G-H, elbow, and wrist functionality. Image taken from Perry et al. (2007)

3.3.2 End-effector

End-effector robotic rehabilitation systems are advanced devices used in physical therapy to assist individuals in regaining motor functions after injuries or neurological conditions. These systems focus on the end-effector, the part of the robot that interacts directly with the patient, allowing for precise and controlled movements during therapy sessions. Unlike exoskeleton-based systems that the patient wears, end-effector systems are designed to interact with the patient's limb through a specialized interface, such as a handle or attachment point. This approach allows the system to precisely control the motion and forces applied to the patient's limb without needing a complete exoskeleton structure. The key advantages of end-effector rehabilitation systems include:

- Targeted therapy: These systems can be tailored to target specific joints or muscle groups by focusing on the end-effector, enabling more focused and practical rehabilitation exercises.

- Adaptability: The end-effector can be designed to accommodate different patient sizes and limb configurations, making this kind of system more versatile and accessible to a broader range of patients.

- Safety: Without a complete exoskeleton structure, end-effector systems can be designed with more straightforward and intuitive safety mechanisms, reducing the risk of injury during therapy sessions.
- Cost-effectiveness: Compared to complete exoskeleton systems, end-effector rehabilitation devices can be more cost-effective to develop and maintain, making them more accessible to healthcare providers.

The end-effector Mirror Image Movement Enabler (MIME) presented in Lum et al. (2006) is a robotic device for upper-limb neurorehabilitation in subacute stroke subjects (Figure 3.6). The study aimed to assess the benefits of the MIME device in improving motor function and reducing motor impairment in patients who had suffered a stroke. The trial involved a randomized controlled design, where patients were randomly assigned to either a robot-assisted treatment group or a conventional therapy group. The study results showed that the robot-assisted treatment group, which included both unilateral and bilateral training modes, significantly improved motor function and reduced motor impairment compared to the conventional therapy group.

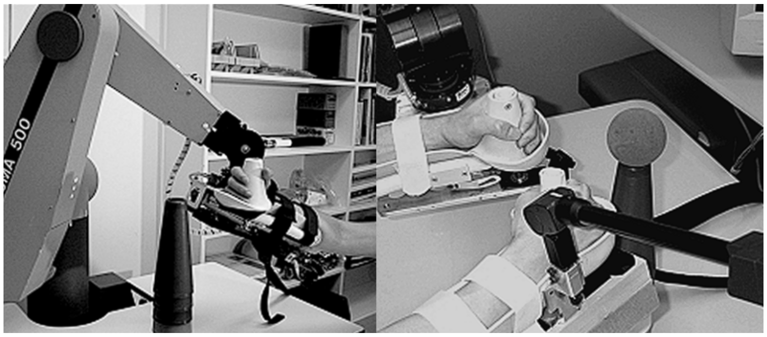


Figure 3.6: Subjects performing (left) unilateral and (right) bilateral movements with Mirror Image Movement Enabler system. Image taken from Lum et al. (2006)

The study also found that the bilateral training mode, which involved mirroring movements of the unaffected arm to assist the affected arm, was particularly effective in improving motor function. The study's findings support the use of MIME robotic devices in the rehabilitation of stroke patients, particularly in the subacute phase, and highlight the potential benefits of incorporating bilateral training modes into rehabilitation protocols.

3.4 Safety in robotic rehabilitation systems

Rehabilitation robots are unique collaborative robots that work in close physical interaction with patients and therapists. This introduces distinct safety challenges compared to industrial collaborative robots. The safety of patients and clinicians is paramount when using these rehabilitation devices, as any malfunction or unexpected behavior could result in serious injury. To address these safety concerns, the concept of “safety skills” has been proposed as a framework in Bessler et al. (2021) for assessing and validating the safety of rehabilitation robots. Safety skills are abstract representations of the robot’s ability to mitigate specific hazards and risks associated with physical human-robot interaction. These include skills such as limiting interaction forces and pressures, limiting interaction energy, maintaining safe interaction distances, and detecting and reacting to unexpected human behavior. By validating these critical safety skills through structured testing protocols, rehabilitation robot developers can help ensure their systems’ mechanical safety, even without comprehensive safety standards and regulations for this specialized domain. This is crucial, as rehabilitation robots work directly with vulnerable patient populations, so comprehensive safety assessment is essential for their safe deployment and widespread clinical adoption. The safety skills approach provides a flexible and adaptable framework, as the same core skills can be implemented differently depending on the specific application requirements. This allows the framework to be applied across various rehabilitation robot technologies as the field evolves. Overall, the safety skills concept represents an essential step towards a comprehensive safety assessment of rehabilitation robots, a vital aspect of their development and clinical use.

3.5 Summary

Robotic rehabilitation systems have been developed based on the techniques used for collaborative robots, which is the basis of vital importance in determining an effective safety environment and maintaining the integrity of the patient and the environment without injury. In general, robotic rehabilitation systems are categorized into two types: 1) Exoskeletons and 2) End-effector. Although each has its advantages and disadvantages, in both proposed models, simplicity safety skills are developed from the construction of the robotic system itself, such as the use of mechanical blockers for the joints in the case of exoskeletons to the use of external sensors, such as biological measurement systems, such as EEG or ECG, to retrieve feedback from the patient and accurately estimate the interaction of the patient with the rehabilitation system. Based on this overview of the different proposed systems, we seek to implement a safety protocol that allows the development of rehabilitation systems in conjunction with general-purpose manipulators to cover the early hand rehabilitation task, being an innovative idea that will allow the use of rehabilitation systems that can ensure a safe environment before clinical trials, without compromising the integrity of the patient or the rehabilitation environment.

Chapter 4

Proposed hand rehabilitation system

In this chapter, we describe the methodology to obtain a safe robotic manipulation system capable of performing rehabilitation exercises on a hand, the hand pose estimation method, the validation protocol and the safety skills proposed for the project, and the experimental model to perform the validation of the proposed protocol. Figure 4.1, depicts the method to operate and control the robotic rehabilitation system. This project is divided into five control modules, for which three important phases are encompassed: 1) Hand pose recognition, represented by the feedback control block, 2) Validation protocol, represented by the adaptive control system block, this phase represents the whole safety system from experimentation to evaluation, and finally, 3) Experimental validation model, represented by the remaining 3 blocks in conjunction with the previous two blocks, together forming the human-robot interaction system capable of developing a safe therapy for hand rehabilitation.

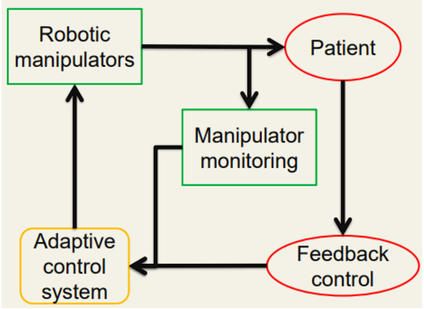


Figure 4.1: Operation diagram of the robotic system for hand rehabilitation, consisting of the robotic manipulator that interacts directly with the patient and through constant monitoring of the manipulator signals and patient feedback, sends the information to the adaptive control system based on safety skills, responsible for limiting, controlling and adjusting the parameters of the robotic manipulator to perform the safe manipulation.

4.1 Hand Pose Recognition

As a fundamental component for the development of this project, real-time hand pose estimation is crucial. The tool for achieving this estimation is MediaPipe, as discussed in Chapter 2. MediaPipe enables the detection of the hand's silhouette and, through a machine learning model, it can recognize and segment the hand into 21 relevant articulation points, as illustrated in Figure 2.4. Given the capabilities of the hand pose estimation tool, color and depth images obtained directly from the Azure Kinect (as shown in Figure 4.2) are used to segment the 21 joints in both images. Subsequently, the joints relevant for the particular rehabilitation scenario are selected, and the position information of each joint is extracted referred to the image from which it is obtained. This, combined with the point cloud data from the depth image, allows for an accurate estimate of the position and depth of the rehabilitation target. This approach ensures precise monitoring and assessment of hand movements, which is essential for effective rehabilitation. The integration of color and depth data enhances the accuracy of joint position estimation, providing a comprehensive solution for real-time hand pose tracking in rehabilitation applications.

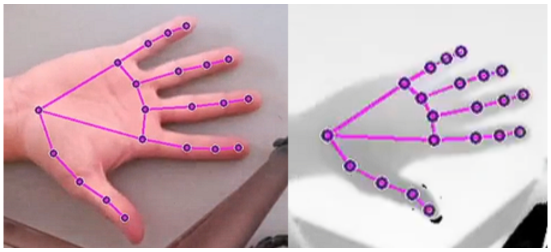


Figure 4.2: Color image (left) and depth image (right), after MediaPipe processing to segment the 21 relevant hand joints.

4.2 Kinova Arm Planning and Control

As previously mentioned, MoveIt includes a control node (move_group) to develop the primary tasks of joint movement for a robotic manipulator. This node is divided into four parts, as shown in Figure 4.3. It contains a user interface to access the main control and manipulation functions in various ways, such as C++ programming, Python, or other interfaces. It can even be used as a high-level control interface, utilizing GUI tools like plugins within Rviz. The move_group node functions similarly to any ROS node, allowing access to the robot's configuration information, such as the robot's semantic description parameters. Communication between the node and the robotic manipulator is achieved through ROS topics and actions, utilizing recurrent robot sensor's values and sending the necessary control information to the control topics. Ultimately, everything converges in the move_group node to develop a trajectory plan considering all the aforementioned parameters.

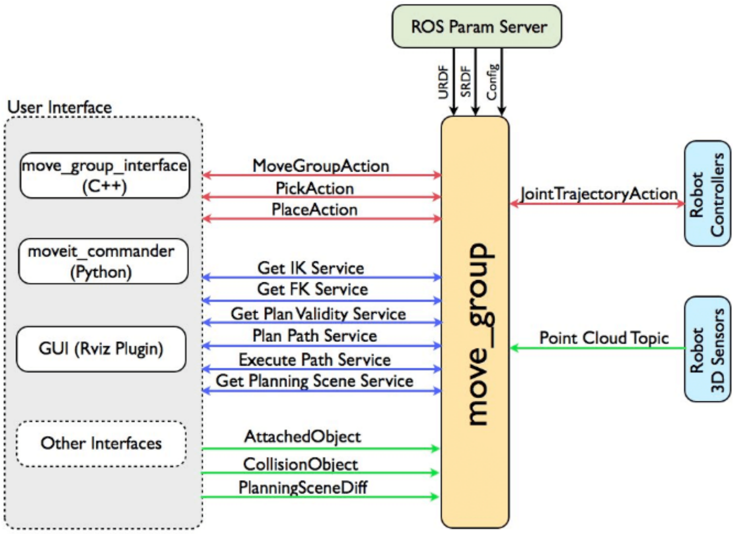


Figure 4.3: Representation of the "move_group" node diagram that interacts directly with the different user interfaces, ROS parameters and controllers or 3D sensors of the robot to control the different functions of the robot. Image taken from https://moveit.ros.org/documentation/concepts/.

Similarly, MoveIt works with motion planners through an interface plugin. This way, MoveIt can communicate with various motion planners. By default, move_-group is configured to use OMPL (Open Motion Planning Library), an open-source motion planning library that primarily implements random motion planners. Consequently, a subnode called Planning Scene Monitor is generated (Figure 4.4), which is responsible for generating pre-processing to solve problems before calculating a trajectory, such as initializing joints outside the configured limits. It also performs post-processing, which is useful in situations like converting the paths generated for a robot into time-parameterized trajectories.

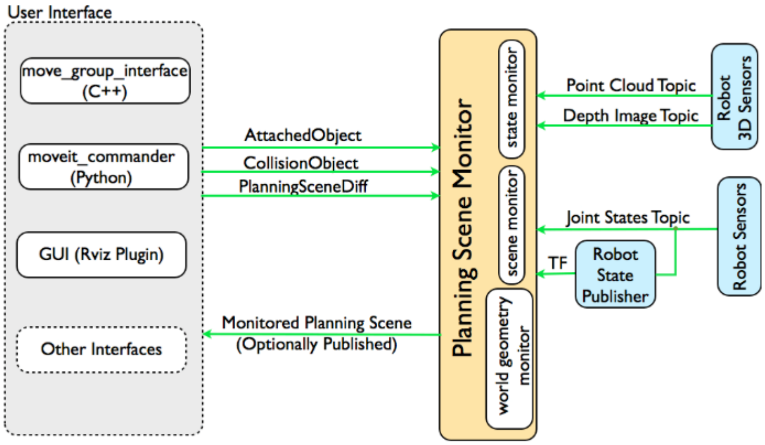


Figure 4.4: Representation of the diagram of the "Planning Scene Monitor" node that interacts directly with the different user interfaces, and controllers, 3D sensors and publishers of the state of the robot's joints to control the different functions of the robot. Image taken from https://moveit.ros.org/documentation/concepts/.

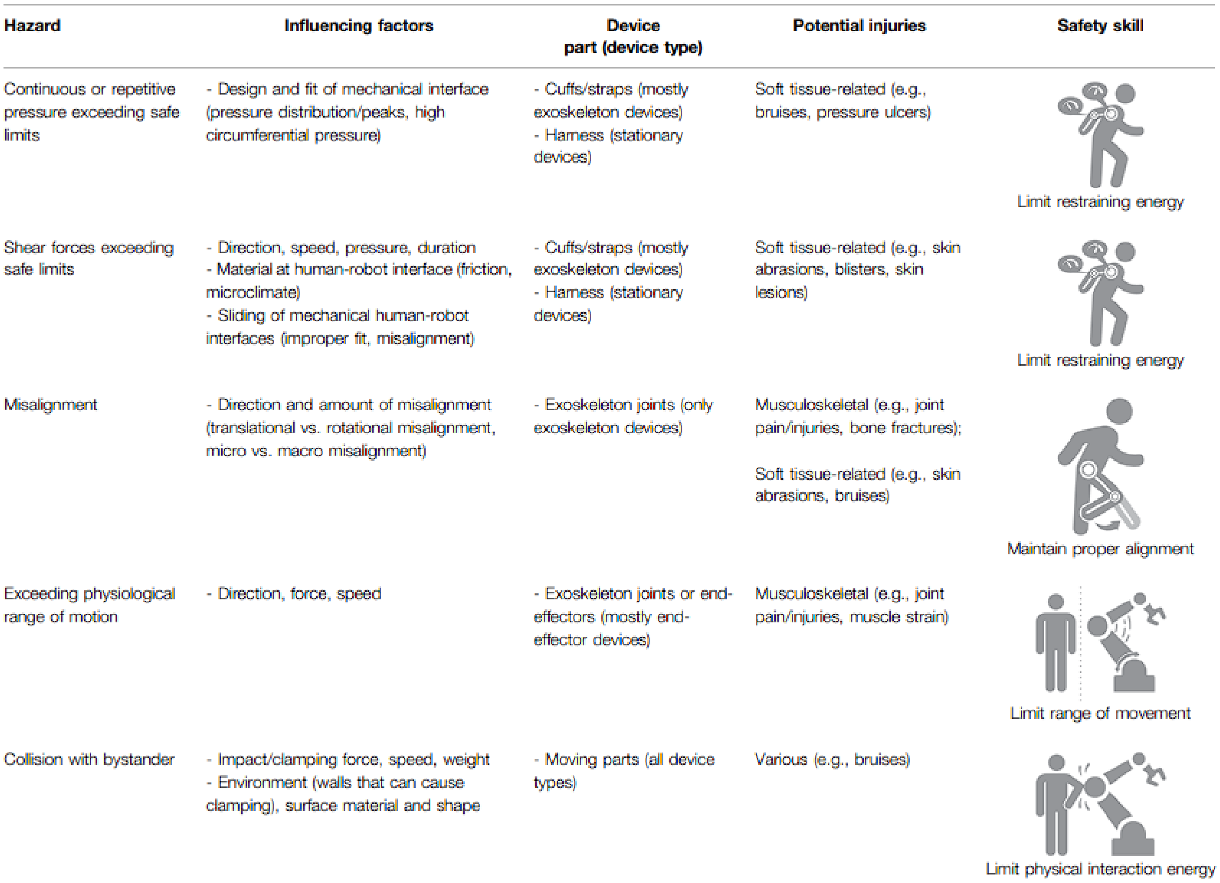
4.3 Safety Skills

The concept of rehabilitation robots is rooted in the regulations designed for collaborative robots commonly found in the industrial sector. However, rehabilitation robotic systems are governed by each country's medical device regulatory frameworks. This regulatory variation often leads to uncertainties regarding the necessary considerations for each specific rehabilitation robotic system.

The study carried out by Bessler et al. (2021) addresses these concerns by proposing a safety protocol based on safety skills. This protocol identifies and mitigates the primary hazards associated with human-robot interaction in rehabilitation environments. The study aims to experimentally validate these safety skills, ultimately ensuring the development of a safe system capable of performing the intended rehabilitation tasks effectively.

As a result, the protocol evaluates skills by identifying hazards, assessing the influence of various factors, examining the interacting parts of the device with the human body part, and considering potential injuries that may occur. This evaluation is summarized in Table 4.5. Our proposal references the safety skills identified in this study. Following a thorough hazard analysis of our specific proposal, we have determined the following safety skills that require validation:

- Limit Range Manipulator Motion
- Restraining Motion Energy
- Limit Physical Interaction Energy
- Orientation Alignment
- Stop Emergency System



Hazard	Influencing factors	Device part (device type)	Potential injuries	Safety skill
Continuous or repetitive pressure exceeding safe limits	- Design and fit of mechanical interface (pressure distribution/peaks, high circumferential pressure)	- Cuffs/straps (mostly exoskeleton devices) - Harness (stationary devices)	Soft tissue-related (e.g., bruises, pressure ulcers)	Limit restraining energy
Shear forces exceeding safe limits	- Direction, speed, pressure, duration - Material at human-robot interface (friction, microclimate) - Sliding of mechanical human-robot interfaces (improper fit, misalignment)	- Cuffs/straps (mostly exoskeleton devices) - Harness (stationary devices)	Soft tissue-related (e.g., skin abrasions, blisters, skin lesions)	Limit restraining energy
Misalignment	- Direction and amount of misalignment (translational vs. rotational misalignment, micro vs. macro misalignment)	- Exoskeleton joints (only exoskeleton devices)	Musculoskeletal (e.g., joint pain/injuries, bone fractures); Soft tissue-related (e.g., skin abrasions, bruises)	Maintain proper alignment
Exceeding physiological range of motion	- Direction, force, speed	- Exoskeleton joints or end-effectors (mostly end-effector devices)	Musculoskeletal (e.g., joint pain/injuries, muscle strain)	Limit range of movement
Collision with bystander	- Impact/clamping force, speed, weight - Environment (walls that can cause clamping), surface material and shape	- Moving parts (all device types)	Various (e.g., bruises)	Limit physical interaction energy

Figure 4.5: Overview of identified hazards, influencing factors, potential injuries and related safety skills. Note that this list is not extensive and additional relevant hazards and safety skills might be identified in the future. Image taken from Bessler et al. (2021)

4.3.1 Limit Range Manipulator Motion

In the work developed by Bessler et al. (2021), it is stated that limiting the range of motion in robotic rehabilitation systems aims to restrict the degree of freedom of a joint to ensure it does not exceed the physiological range of motion of the limb being treated. For devices with more complex joints or end-effectors, the spatial range of motion can be measured using a marker-based motion analysis system. Following the logic of the marker-based system and considering the use of a robotic manipulator to make contact with its end-effector and the finger of a hand, a methodology must be implemented. This methodology involves first delimiting the safe area of action for the manipulator. Subsequently, it is essential to establish an accurate spatial relationship between the manipulator and the rehabilitation target, which in this case is a finger of the hand.

To limit the safe area of the manipulator, it is essential to consider the position of the table supporting the manipulator, as well as the placement of other elements in the rehabilitation system, such as the camera base and the 3D hand model base. We are using the MoveIt tool for direct manipulation and integration of the manipulator joints' information with other safety-relevant data, with this tool virtual blocks were added (Figure 4.6). These blocks represent the movement limits for the manipulator and serve as obstacles to prevent collisions.

By incorporating these blocks, the MoveIt tool ensures that the manipulator operates within a predefined safe area, thereby minimizing the risk of unintended interactions and enhancing the overall safety of the rehabilitation system.

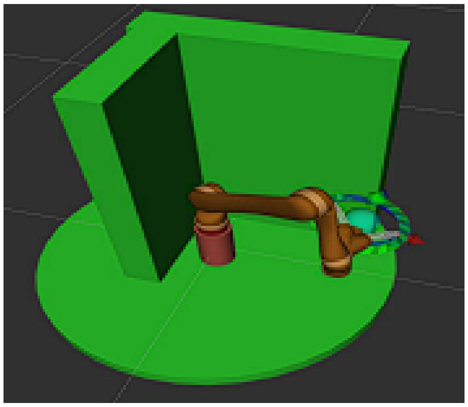


Figure 4.6: Safe working area to avoid collisions with objects outside the interaction environment for rehabilitation delimited by green blocks within the virtual vision.

To complement the limited range of movement to control the manipulator, the point cloud from the Kinect depth sensor is obtained and segmented based on the prior estimation of the hand pose. This results in the silhouette of the 3D hand model being represented by the point cloud, as illustrated in Figure 4.7.

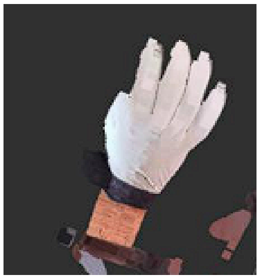


Figure 4.7: 3D model of the hand segmented with respect to the point cloud.

Finally, it is necessary to implement the marker-based analysis system to establish the spatial relationship between the manipulator and the 3D hand model and segment the point cloud region of interest to implement the Deep Grasp approximation tool. A practical and robust solution involves considering the initial position and orientation of each element. This approach entails obtaining the image transformations of the manipulator, the depth camera, and the segmented 3D model of the point cloud.

Figure 4.8 provides a general composition of the system, using the manipulator base as a reference. The process includes performing rotation (Figure 4.9) and translation (Figure 4.10) transformations from the initial position of the depth camera. With the aid of a marker code, the spatial relationship between the camera and the marker is calculated. This allows us to determine the spatial relationship between the depth camera and the manipulator base (Figure 4.11).

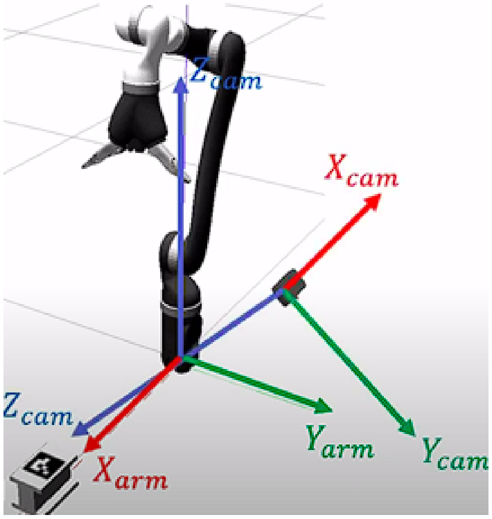


Figure 4.8: Spatial representation of the position and orientation of the robot base and camera frame in a virtual environment.

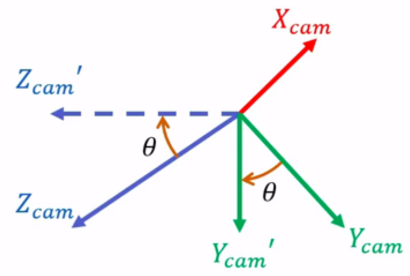


Figure 4.9: Representation of the rotation performed by the position and orientation of the camera frame to approach the robot base.

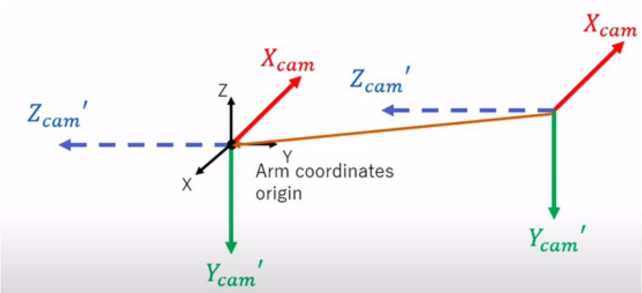


Figure 4.10: Representation of the traslation performed by the position and orientation of the camera frame to approach the robot base.

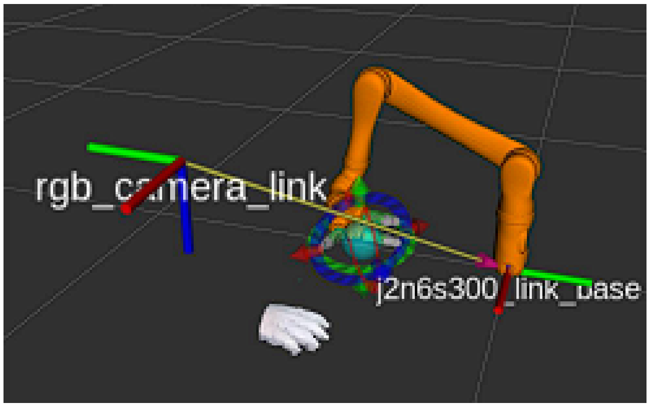


Figure 4.11: Experimental relationship between the robot base frame and the camera after performing the rotation and translation process.

At this point, we obtained the spatial relationship between the manipulator and the 3D hand model, observed through the point cloud obtained by the depth camera. We segmented the bottom and the rest of the hand, leaving only the region of interest of the hand, which in this case corresponds to the section from the interproximal phalanx to the tip of the index finger (Figure 4.12) since this will be the region of manipulation used to develop the validation experiments of some of the safety skills including the limit of the range of motion of the manipulator.

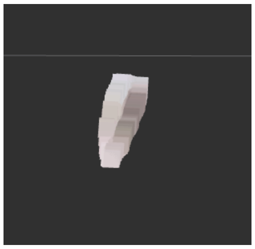


Figure 4.12: Segmented point cloud, where only the silhouette of the index finger up to the interproximal phalanx can be observed, taking the original point cloud obtained by the depth camera as a reference.

4.3.2 Restraining Physical Interaction and Energy

In general, whenever there is direct interaction between any parts of the rehabilitation robot and human tissue, physical interaction must be limited, which involves directly limiting the robot's energy and the forces it can exert in such cases to keep interactions within safe limits, thereby preventing any harm to human skin. This includes limiting the energy directly applied by the robot to the movements of the end effector joints to maintain safe pressure levels that can be exerted on human skin and the power of the other joints of the robot to avoid damage during collisions with the system's environment, the patient, or any other person who may be present during the therapy session, such as the rehabilitation specialist or a technician in charge of the system. The following subsections describe the method employed to limit the robot's energy in each of the previously described cases.

4.3.2.1 Limit Restraining Energy

The limit restraining energy is applied directly to the configuration files of the robotic manipulator, such as the Unified Robot Description Format (URDF) files, which allow defining some of the robot configuration parameters. This safety skill aims to directly limit these executable properties both on the robot and in simulation environments. However, in addition to initially limiting these parameters, it is necessary to develop a control system for the specific rehabilitation task. In our case, the system will have direct contact between the manipulator's end effector and one of the hand's phalanges. Various control methods exist for achieving more suitable trajectories when manipulating soft or rigid objects. Similarly, in Abdullahi, Haruna, and Chaichaowarat (2024) , an adaptive hybrid control system is presented, capable of alternating between an admittance or impedance control block to develop smoother trajectory control when implementing a robotic rehabilitation system based on an exoskeleton for upper limbs. The hybrid control model offers an advantage for exoskeletons due to the strict contact these systems have with the skin and arm joints. However, for the development of our project, it will only be necessary to implement an impedance-based control system, which achieves better results in contact with soft surfaces.

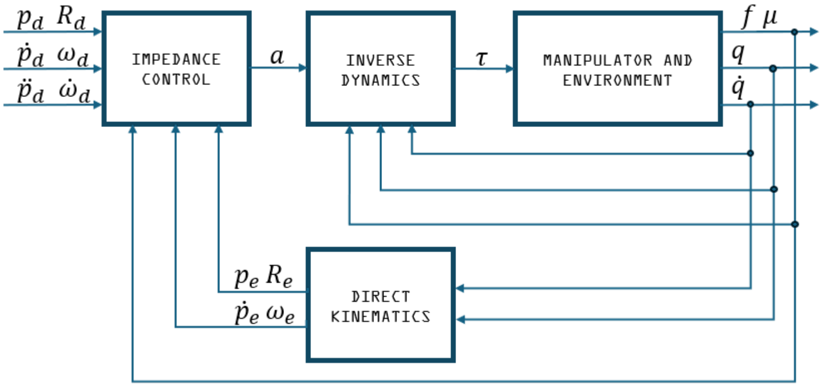


Figure 4.13: Basic impedance control for robotic manipulators with one or more joints. The impedance control block receives information directly from the kinematics of the manipulator and its environment to compute the control law via inverse dynamics, finally adjusting the response of the robotic manipulator to perform fine and smooth hand manipulation.

The impedance control block shown in Figure 4.13, contains the main control elements for a robotic manipulator system. On one side, the desired values for position, orientation, rotation matrix, and both linear and angular velocities and accelerations are inputs into the impedance control block, along with the current values for force, position, rotation matrix, and angular velocities of the manipulator. In the impedance block, the necessary calculations are performed to obtain a new control input, then passes to the inverse dynamics block. Subsequently, we obtain a torque value and, finally, the force and each joint velocities that manipulator will execute. This process is fed back continuously throughout the manipulation task to ensure that stable and smooth control is maintained at all times for the specific task.

4.3.2.2 Limit Physical Interaction Energy

In the case of limiting the energy of physical interaction, the aim is to restrict the power and speed of the manipulator's actuators to prevent damage that may arise from accidental contact between any part of the manipulator and any internal or external elements present during the rehabilitation session, such as a wall, the table, the therapist, or the patient.

The pressure sensor is the main component responsible for measuring the pressure exerted by the manipulator on the 3D hand model. The characterization and parameterization of the sensor for the specific rehabilitation task will obtain the pressure values that validate the safety capability. The pressure sensor will be secured to one of the phalanges of the 3D hand model, and pressure and impact tests will be conducted to validate the system's safety as shown in figure 4.14, ensuring that safe pressure values are not exceeded at any time.

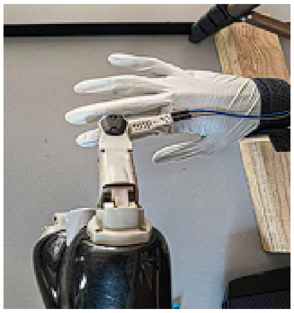


Figure 4.14: Experimental example of pressure measurement through the pressure sensor attached to the 3D hand model.

In this way, the system will utilize the same pressure sensor, and trajectories will be plotted to collide with the system's external elements. If the safe pressure limits for the various external elements are not exceeded in any repetition, it will be assumed that the validation of the safety capability is complete.

4.3.3 Orientation Alignment

The safety skill proposed in Bessler et al. (2021) aims to ensure a safe alignment between the joints of the limb being rehabilitated and the joints of an exoskeleton used as a robotic rehabilitation device. In our case, these types of joints have no direct alignment or coupling. However, the direct manipulation between the end effector of the robotic manipulator and the phalanges of the hand must maintain precise alignment to prevent any damage to the limb during manipulation and the execution of therapeutic exercises.

By obtaining a relationship between the orientation of the robotic manipulator and the hand, a safe trajectory for grasping the manipulation target, in this case, the 3D hand model, can be estimated. To do this, it is first necessary to obtain the orientation of the hand and the grip angle, and then the relationship between the orientation of the hand and the manipulator is calculated.

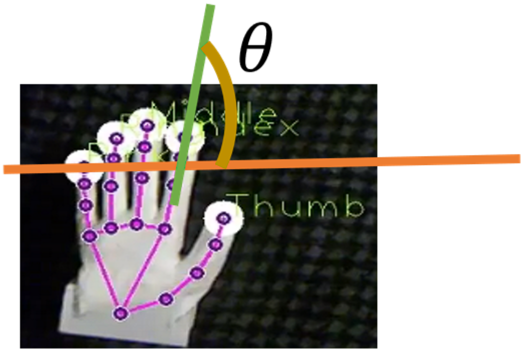


Figure 4.15: Representation of the reference vectors used to determine the planar rotation angle of the hand with respect to the camera image.

For the first part, the camera position is taken into account, the hand pose is estimated and two reference lines are drawn on the color image (Figure 4.15).

The first line represents a horizon directly related to the camera position and the rehabilitation environment, whereas the second line represents the direct connection between the tip of the index finger and its inter-proximal phalanx.

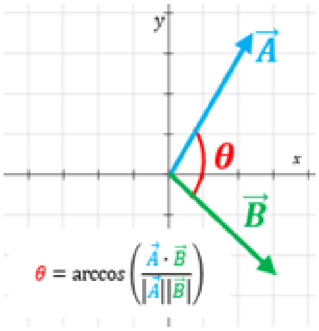

y
$\vec{A}$
θ
x
$\vec{B}$
$\theta = \arccos\left(\frac{\vec{A}\cdot\vec{B}}{|\vec{A}||\vec{B}|}\right)$

Figure 4.16: Representative diagram of the arc-sine function to obtain the planar rotation angle of the hand.

By observing these two reference lines as vectors originating from the same point at the interproximal phalangeal joint, we can apply the arc-sine relationship between these vectors to obtain the hand's rotation angle (Figure 4.16), ranging from 0 to 360°. This information allows us to validate that the hand's rotation falls within the acceptable range for the predefined manipulation area, ensuring the manipulator operates within safe limits and prevents any risk of injury to the patient. Additionally, using the calculated rotation angle, we can determine the expected orientation of the manipulator before planning the trajectory. This ensures the manipulator is correctly aligned with the hand, facilitating safe and precise manipulation. By integrating these calculations, the system can dynamically adjust to the hand's orientation, maintaining a safe and effective trajectory throughout the rehabilitation exercises, thereby enhancing the overall safety and efficacy of the robotic rehabilitation process.

At this point, we can only ensure the estimation of the hand rotation from the camera view, which means that a depth view is needed to detect the spatial orientation of the hand before calculating the manipulation trajectory. First, we take the same reference of the tip of the index finger and its inter-proximal phalanx and draw a line perpendicular to our reference, as a pivot to adjust the orientation of the manipulator (Figure 4.17). After that, a new reference is taken between the tip of the middle finger and its respective inter-proximal phalanx, the estimated coordinates of the hand pose of these four joints are joined together forming a box, together with the estimation of the depth camera in the points of the joints, the depth difference between each of the joints concerning their respective edges is obtained. According to the depth difference between the edges of the frame, a relation with the position and orientation of the hand can be obtained as shown in the example of Figure 4.18.

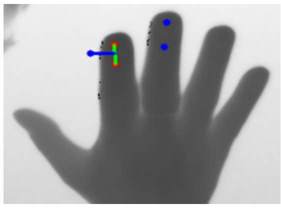


Figure 4.17: Reference pivot to determine the orientation of the manipulator concerning the hand orientation.

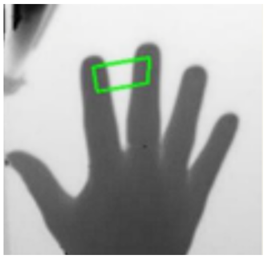


Figure 4.18: Reference box, obtained by joining the fingertip joints and the interproximal phalanges of the index and middle fingers.

4.3.4 Stop Emergency System

Traditional therapy aims for progressive improvement through exercises that challenge the joints without causing harm to the patient. In this case, the objective is to develop a safe system for hand rehabilitation. This skill aims to complement previously discussed skills with an immediate stop system controlled by the patient, mimicking the interaction they would have with a therapist if an exercise became uncomfortable. The system includes an emergency button as a switch that disconnects the power supply or stops the current task. This complements other components, such as hand pose and orientation estimation, validated safety skills, and an analogue signal controlled by the patient. In series with the monitoring system and the robotic manipulator's actions, set up a safe and effective rehabilitation system for hands with low mobility (Figure 4.19). The integration of these elements ensures that the patient can immediately halt the rehabilitation session if discomfort arises, providing a level of safety and responsiveness akin to a human therapist's intervention. This holistic approach aims to enhance patient safety while effectively performing the required rehabilitation exercises.

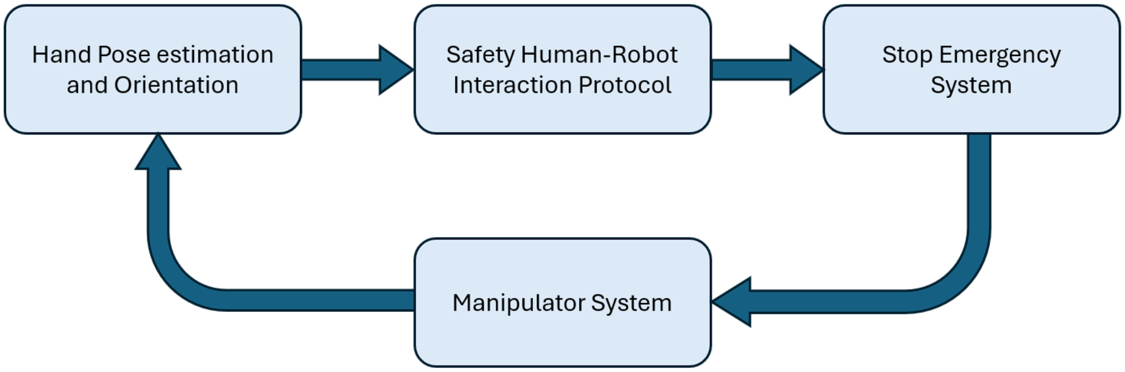

Hand Pose estimation and Orientation
Safety Human-Robot Interaction Protocol
Stop Emergency System
Manipulator System

Figure 4.19: Control diagram for the rehabilitation system with the emergency stop module implemented.

4.4 Experimental Validation Model

Finally, it is necessary to develop an experimental design and broadly outline the validation of the safety skills. The proposed robotic rehabilitation system consists of four physical elements: the depth camera, the robotic manipulator, a 3D model capable of measuring pressures and forces, and an emergency button. The depth camera and the manipulator share the skill to limit the range of motion and ensure orientation alignment. From the manipulator's side, and by using information from the 3D hand model, it is implemented the limited physical interaction and restraining energy skills. The emergency button provides immediate feedback and control, allowing the patient to halt the system if necessary. These components work together to ensure a safe and effective rehabilitation environment. The experimental setup is depicted in Figure 4.20, illustrating how each element contributes to the overall safety and functionality of the system.

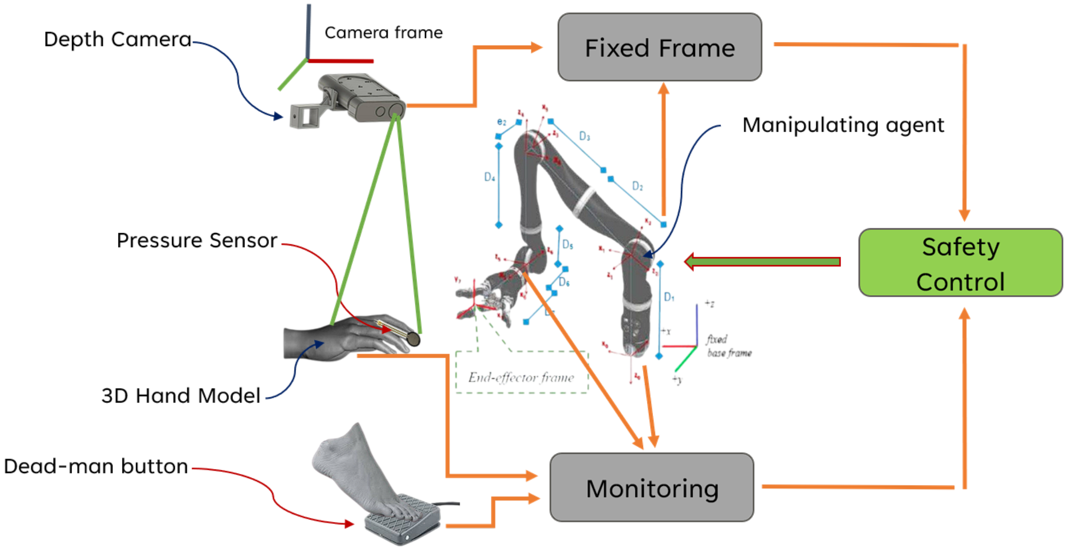


Figure 4.20: Experimental Model in charge of performing the security tests to validate the security protocol.

4.5 Summary

This chapter expresses in detail the methodology to carry out the experimental development of the thesis, addressing issues such as the method for estimating the hand pose, the approach and validation of the specific safety skills for this project and the experimental model designed to develop the necessary tests, seeking to validate the safety skills and determine that the system is capable of developing a rehabilitation routine without compromising the integrity of the patients, their environment and the rehabilitation system itself.

Chapter 5

Experiments and Results

In this chapter, we present the results of our experimental model. First, we provide a comprehensive summary of the experiments conducted within the framework of the experimental model to validate the safety skills. This involves a detailed account of the procedures followed, the conditions under which the experiments were performed, and the criteria used for validation. We ensure that the experimental setup aligns with the standards required for rigorous testing.

Next, we detail the specific experimental considerations for each safety skill, utilizing the tools and sensors outlined in the methodology section. This includes an in-depth description of the tools and sensors used, their calibration, and their role in capturing the necessary data. For each safety skill, we discuss the setup, execution, and data collection processes. We also highlight any challenges encountered during the experiments and the steps to mitigate them.

Finally, the results are summarized to validate the safety of the human-robot interaction protocol. This summary includes comparing the expected outcomes with the actual results, highlighting the discrepancies and their possible reasons. The chapter concludes by critically evaluating the experimental model and discussing its strengths and limitations.

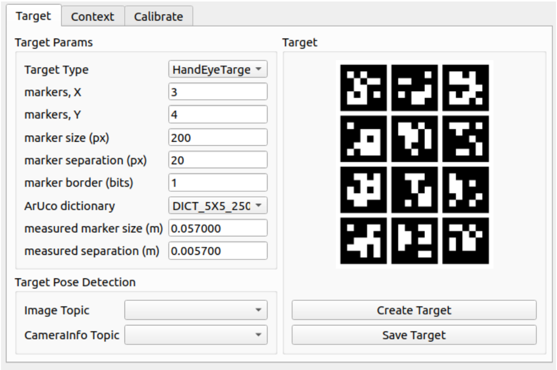
In a general way, an example of estimation of the pose and fine manipulation of the hand can be observed in the following link, directed to a video demonstration of the expected final result: Demostrative Video.

5.1 Limit manipulator range of motion validation

As mentioned earlier in the methodology chapter, the first step to limiting the manipulator's range of motion is to delimit the working area, this can be achieved with the help of MoveIt tools to create virtual obstacles, which work as walls to avoid collisions during the trajectory execution by the manipulator. In this way, we manage to delimit the safe working area (Figure 4.6).

Subsequently, the spatial relationship between the position of the robotic manipulator and the position of the depth camera was used, to obtain the precise position of the 3D hand model, visualized directly with the depth camera. For this propose, the calibration tool "Hand-Eye" was used, to obtain the position and orientation of the depth camera with respect to the base of the manipulator.

The calibration tool makes use of an Aruco reference code that can be generated automatically by the tool itself (Figure 5.1). Once the code is obtained we must configure the calibration parameters and select the type of calibration. In our case it corresponds to "Hand-Eye" calibration since the depth camera is positioned in a specific place, without modification, instead of being positioned directly above the manipulator. It is also necessary to select the transformed matrix of the base and end-effector of the manipulator. Once all the necessary parameters for the calibration have been selected, we must position the depth camera in the viewing location, place the aruco code previously generated on the manipulator end-effector clamp and finally we will take samples of the manipulator position by orienting and placing the aruco code in different positions, without leaving the viewing range of the depth camera.



Target Params

Target Type	HandEyeTarge
markers, X	3
markers, Y	4
marker size (px)	200
marker separation (px)	20
marker border (bits)	1
ArUco dictionary	DICT_5X5_25C
measured marker size (m)	0.057000
measured separation (m)	0.005700

Target Pose Detection

Image Topic	
CameraInfo Topic	

Figure 5.1: Aruco code generator, capable of customizing the number of markers on the X, Y axis, their size, spacing, among other parameters to have a specific code for the required calibration task.

Following this initial configuration, several samples were taken, until the projection error was reduced as much as possible. Within the first approximation, a projection error of 0.03 m was obtained after 100 samples from the manipulator holding the Aruco code in different positions. Then, by using the Grasp Pose Detection tool directly on the segmented point cloud corresponding to the phalanx of the index finger, the approximate grip position of the manipulator was obtained. With this information the manipulator was sent to the estimated position and in the case that the manipulator was not in the expected position, it was manually repositioned and the difference in its 3 coordinate axes was obtained, after 50 experimental repetitions the following results were obtained with this approximation (Figures 5.2 to 5.4).

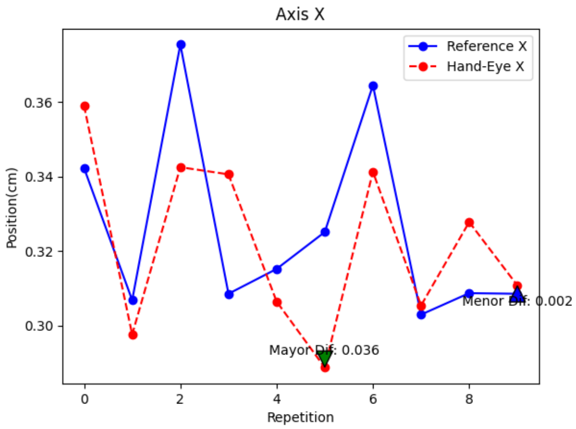


Figure 5.2: Result of approximation to the middle phalanx of the 3D model on the X-axis using the standard “Hand-Eye” calibration method.

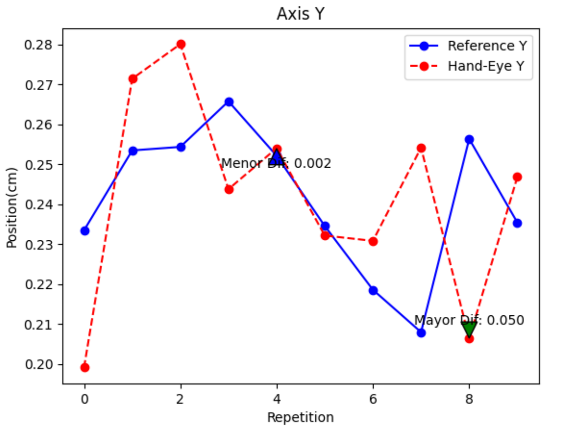


Figure 5.3: Result of approximation to the middle phalanx of the 3D model on the Y-axis using the standard “Hand-Eye” calibration method..

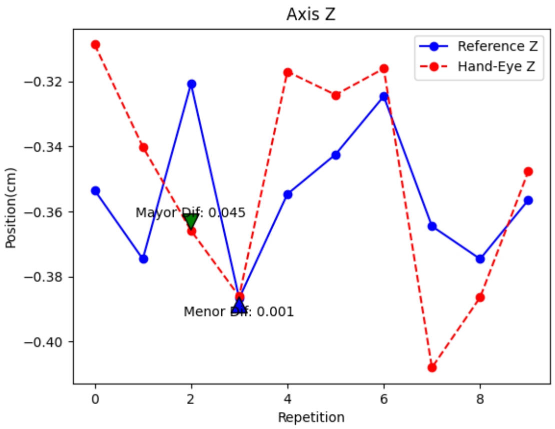


Figure 5.4: Result of approximation to the middle phalanx of the 3D model on the Z-axis using the standard “Hand-Eye” calibration method..

As can be seen in the previous results, there is a notorious variation between the expected position provided by GPD and the real position of the manipulator with this calibration method. In the best case, a difference of 1 cm was obtained between the real and approximate position, however, in the worst case, the difference was up to 5 cm. Taking into account that the objective is a phalanx that measures approximately 2 to 3 cm, this is a great difference and therefore it is not possible in this way to ensure the position of the manipulator to perform a safe fine manipulation for the patient.

For the next approach, the same configuration parameters for the “Hand-Eye” calibration tool were employed, but the camera position was manually modified to match the position of the end effector coordinate axis with the Aruco code coordinate axis. This process was first performed in the real environment (Figure 5.5) and then the parameters of the calibration tool were modified until the coupling of the manipulator and the Aruco code was obtained (Figure 5.6).

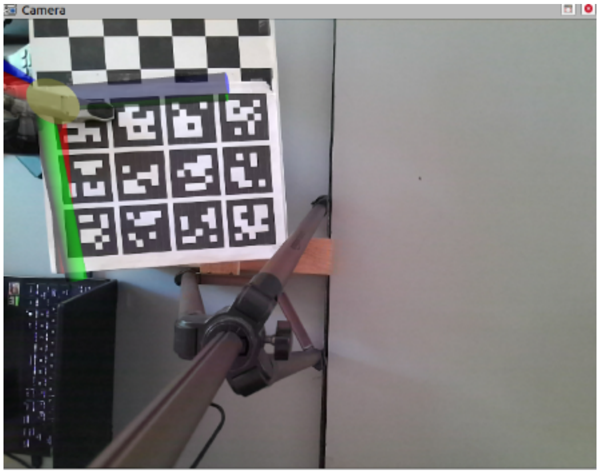


Figure 5.5: RGB image of the Kinect camera, where it can be appreciated the manual placement of the Aruco code directly on the manipulator's end effector, approximating the transformation matrices.

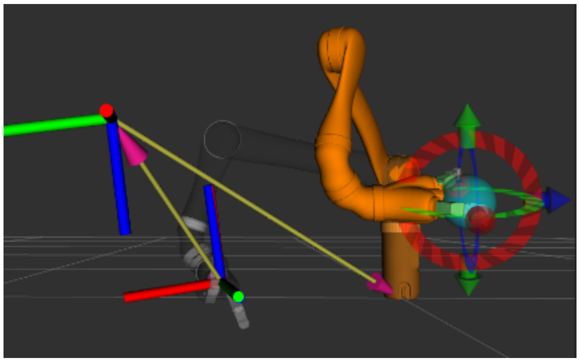


Figure 5.6: Visualization of the match between the transformed end-effector matrices and the aruco code, after manual modification of the parameters within the "Hand-Eye" calibration tool.

With this approach, 50 more experiments were performed and the best results are shown in figures 5.7 to 5.9.

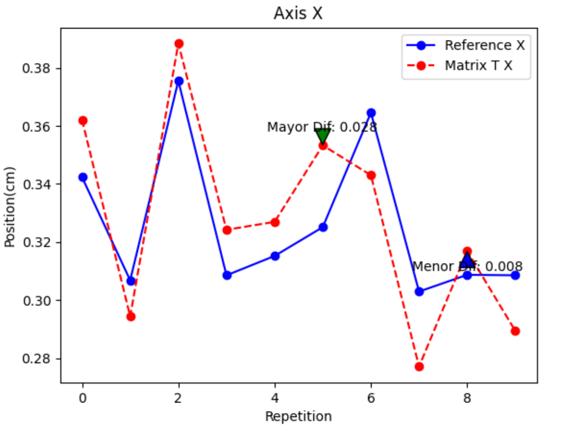


Figure 5.7: Result of approximation to the middle phalanx of the 3D model on the X-axis using the matrix transform as reference.

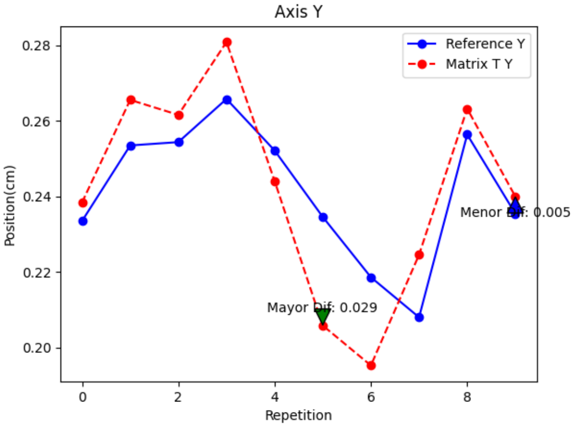


Figure 5.8: Result of approximation to the middle phalanx of the 3D model on the Y-axis using the matrix transform as reference.

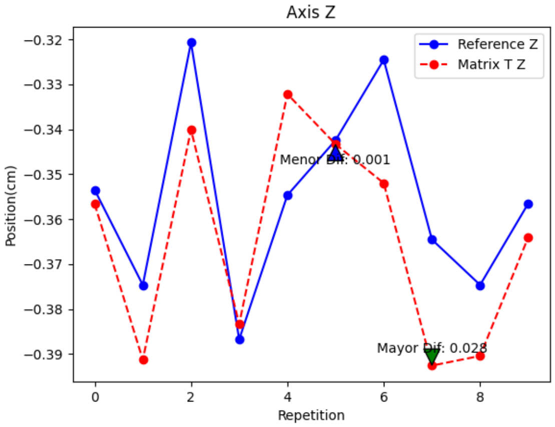


Figure 5.9: Result of approximation to the middle phalanx of the 3D model on the Z-axis using the matrix transform as reference.

In the above results, when considering the alignment approximation of the transformed matrices between the end effector and the Aruco code board, better results were obtained, with a maximum difference between the expected position and the real position of up to 3 cm. However, it can be noted that there is still a large difference in most of the results obtained and 3 cm is still a considerable enough difference to conclude that the fine manipulation of the hand cannot be performed.

The last approximation required first segmenting the point cloud to obtain only the point cloud corresponding to the middle phalanx of the index finger (Figure 5.10), then manually placing the manipulator over the expected grip area to perform the fine manipulation, and finally manually adjusting the "Hand-Eye" parameters to approximate in the display (RViz) the real position of the 3D model with respect to the manipulator (Figure 5.11).

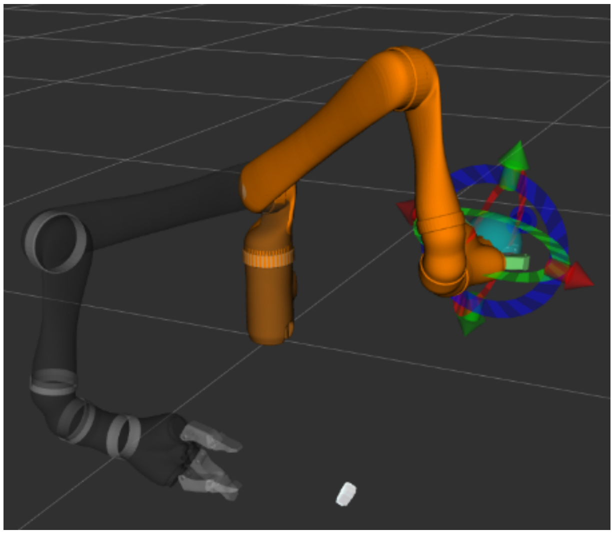


Figure 5.10: Point cloud segmentation of the middle phalanx of the 3D model visualized within Rviz with respect to the robotic manipulator.

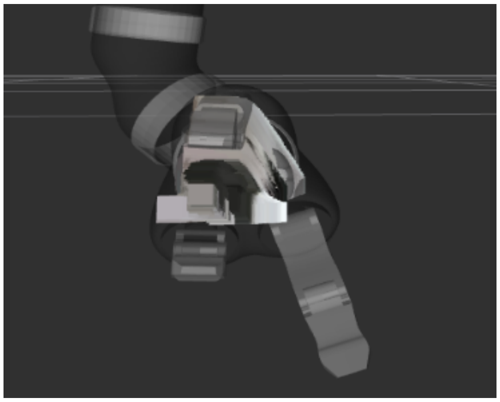


Figure 5.11: Manual manipulator match with respect to the real position of the 3D model visualized through Rviz.

With this approach, 50 more experiments were performed and the best results are shown in figures 5.12 to 5.14.

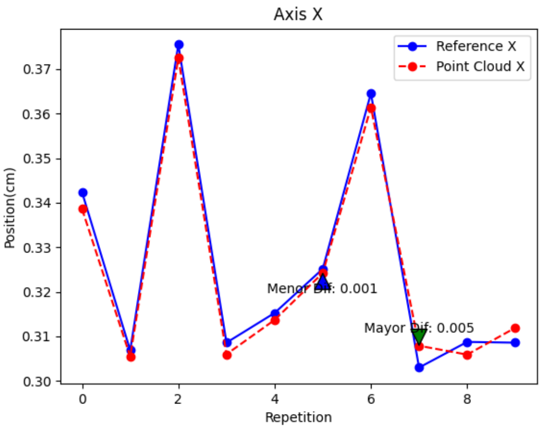


Figure 5.12: Result of approximation to the middle phalanx of the 3D model on the X-axis using the Point cloud segmented as reference.

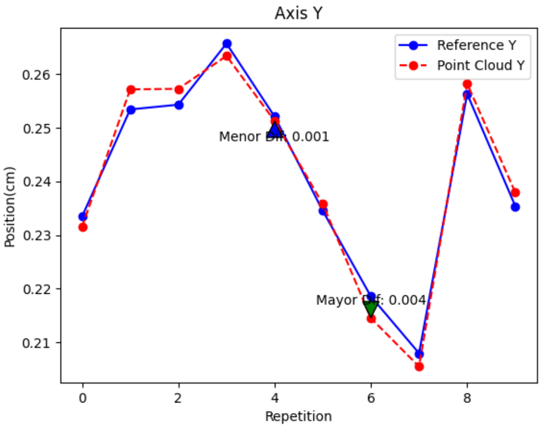


Figure 5.13: Result of approximation to the middle phalanx of the 3D model on the Y-axis using the Point cloud segmented as reference.

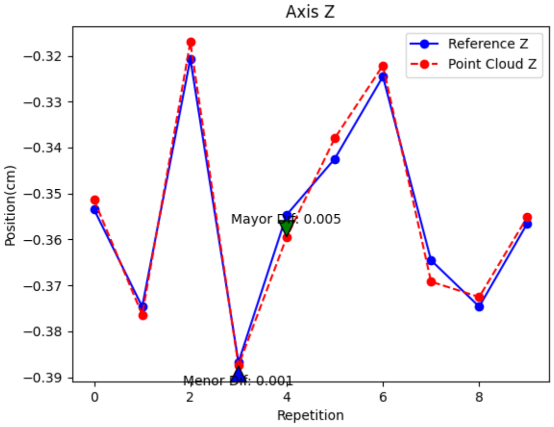


Figure 5.14: Result of approximation to the middle phalanx of the 3D model on the Z-axis using the Point cloud segmented as reference.

Comparing the 3 approximations using the “Hand-Eye” calibration method, it is observed that the best result was obtained using the point cloud approximation, which obtained a maximum difference of 0.5 cm and was, therefore, the best result among the 3 approximations because, taking into account that the manipulation target is the phalanx of a hand, precision is fundamental considering that on average a phalanx can measure 2 to 3 cm, being the other calibration methods greater than 3 cm of error, it could mean less precise approximations and generate significant damage to the patient’s integrity.

5.2 Limit Physical Interaction Energy validation

The physical interaction limit is mainly in charge of generating a safe interaction environment during the hand manipulation actions and execution of trajectories for the approach of the manipulator with the patient’s hand. Thus, this safety skill becomes one of the most important skills to avoid damage to the patient and the system environment. To perform the proper validation of this skill, it is necessary to divide it into two phases. The first phase corresponds to the detection of unexpected collisions during the execution of trajectories that correspond to the positioning of the manipulator in the work area close to the hand and the approach of the manipulator to grasp the hand. Both motions are executed with the same trajectory planning method and are expected to fulfil the main function of approaching in two steps to the current position of the hand to subsequently perform the fine manipulation. The second phase in which this safety skill is divided corresponds to the pressure or force exerted on the finger (or fingers) of the hand that will be manipulated to perform the rehabilitation exercises.

To validate the first phase of this safety skill, a trajectory was defined that allows moving the manipulator from its main "home" position to a position called "setup", which has the objective of placing the robotic manipulator in the safe manipulation zone and oriented with respect to the estimated position and orientation of the hand, Subsequently, a trajectory was defined that allows moving the manipulator from the "setup" zone to a general approach position, simulating the precise approach of the manipulator to the hand, in order to bring the manipulator gripper closer to the fine manipulation zone of the hand and to detect if there is any collision during the planning and execution of this trajectory.

Ten experiments were performed for the "home - setup" trajectory path and 10 more for the "setup - manipulation" trajectory, data were collected from the information provided by the communication topics of the robotic manipulator "/j2n6s300_driver /out/tool_wrench" during the action of each path, where the following representative results were obtained in figures 5.15 to 5.18.

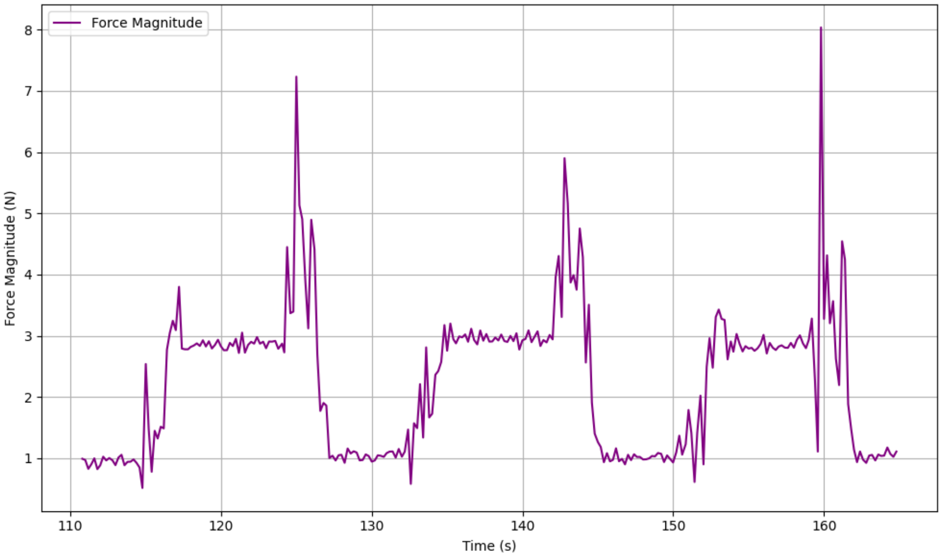


Figure 5.15: Representative end-effector force magnitude graph of the 3 best results during the development of the "home-setup" trajectory without collisions.

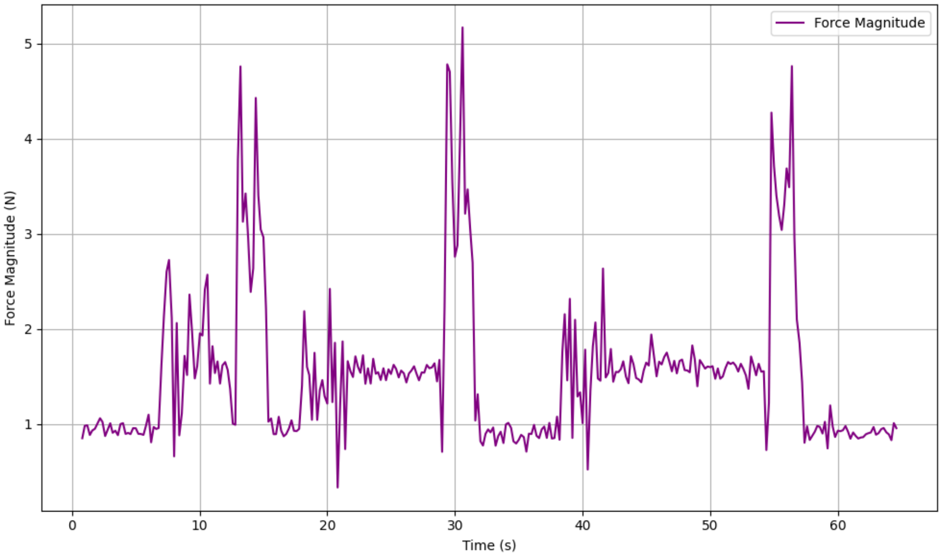


Figure 5.16: Representative end-effector force magnitude graph of the 3 best results during the development of the "setup-calibration" trajectory without collisions.

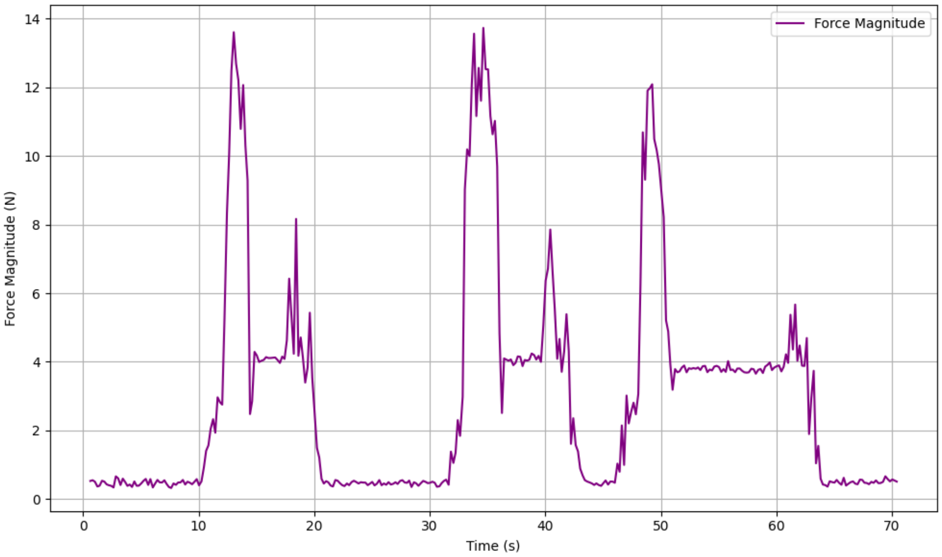


Figure 5.17: Representative end-effector force magnitude graph of the 3 best results during the development of the “home-setup” trajectory with collisions.

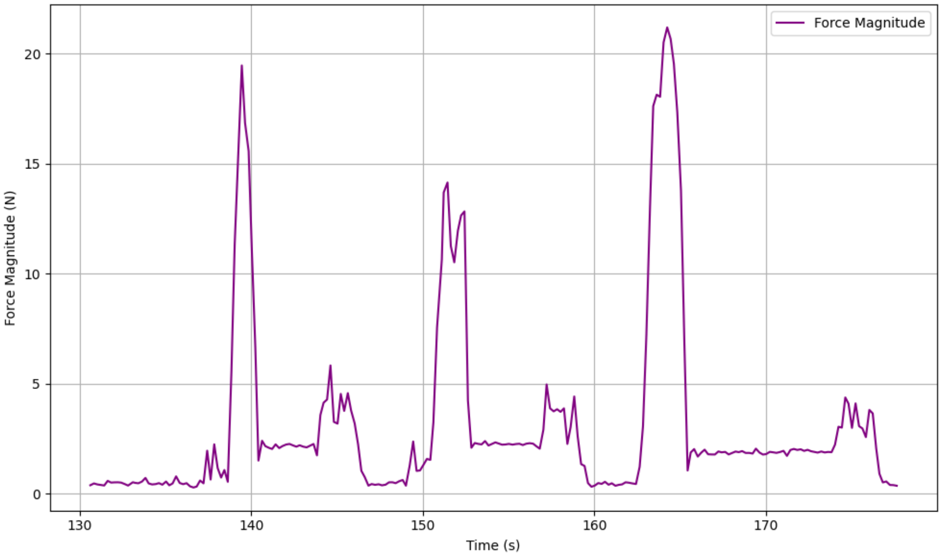


Figure 5.18: Representative end-effector force magnitude graph of the 3 best results during the development of the “setup-manipulation” trajectory with collisions.

In the first 2 graphs we can find the results after applying the modulus to the end-effector forces result in 3 different repetitions, both for the “home-setup” trajectory (Figure 5.15) and for the “setup-manipualation” trajectory (Figure 5.16), in both cases the trajectory is clean, without obstacles or resistance that could indicate a collision during the development of both trajectories.

On the other hand, the following 2 graphs represent the result after applying the module to the end-effector forces resulting in 3 different repetitions, both for the “home-setup” trajectory (Figure 5.17) and for the “setup-manipulation” trajectory (Figure 5.18). In these cases, a small resistance was recreated during the development of both trajectories to simulate a small collision between one of the moving parts of the manipulator and an object with low motion resistance (or mass).

It can be observed that during the experimental run of the trajectories with collision there is a considerable amplitude between the results of the same trajectory without collision, for the case of the first trajectory we have a value of 8 N as maximum value compared to the value of 20 N, giving as a result that during this first trajectory, values above 8 N (Newtons) could mean a collision during the development of this trajectory.

In the second phase, it was necessary to know how to characterize the pressure sensor, which, according to the diagram in figure 5.19, is an arrangement composed of the pressure sensor, a 1MΩ resistor and the power supply, all connected through an Arduino board to recover the analog value delivered by the pressure sensor and thus obtain the value of the force exerted on the 3D hand model.

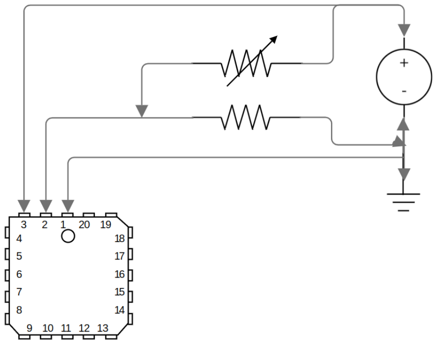


Figure 5.19: Diagram of the circuit designed for the characterization of the pressure sensor, composed of two resistors, one with an experimental value of 100Ω, 1KΩ, 10KΩ, 100KΩ and 1MΩ and the other resistance, variable, is a strain gauge that converts the surface pressure into resistance.

The variable resistance was measured manually and a range of resistance was obtained that goes from 8kΩ when the highest measurable pressure is exerted by the sensor to 2MΩ or maximum resistance of the material, when no force is exerted on the sensor. The constant resistances were exchanged experimentally from the 5.19 diagram, finally it was obtained that the resistance of 1MΩ, showed to cover the whole range of the voltage used for the tests (5V). This resistance value allowed to measure a wider range of voltage in response to the intensity of pressure exerted on the sensor.

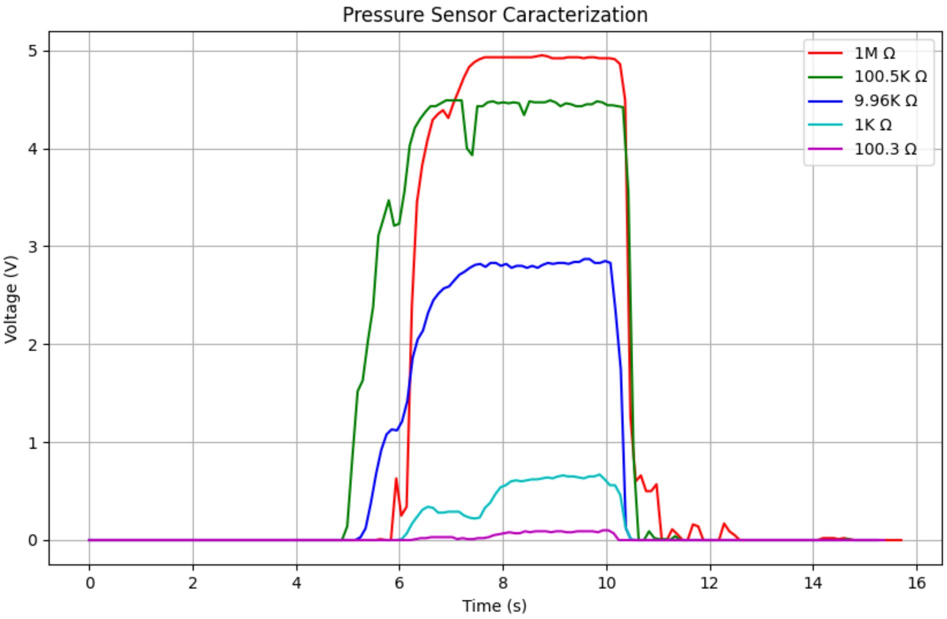


Figure 5.20: Characterization graph of the pressure sensor, comparing the different results of the constant resistances to a maximum pressure exerted manually on the strain gauge.

After the characterization, we place the sensor on the index finger of the 3D model and hold it with a little tape, allowing the center of pressure of the sensor to be positioned freely and on the middle phalanx of the index finger. The gripper of the robotic manipulator was manually placed on the center of the middle phalanx of the index finger and we began the validation process.

With the constant resistance of 1 MΩ and the Arduino power supply of 5 V, as initial configuration. Started to close the gripper of the robotic manipulator manually and exert the maximum possible pressure on the 3D model and the center of the pressure sensor (figure 5.20), from this measurement, it was performed finger manipulation tests, making short trajectories of rotation and elevation of the end effector.

First, several tests of the maximum pressure that the end-effector fingers could exert were performed and the pressure sensor (Figure 5.21) and manipulator finger current (Figure 5.22) plots were obtained.

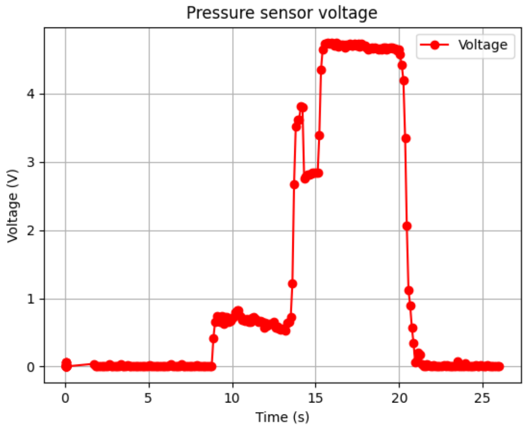


Figure 5.21: Graph of the maximum voltage obtained by the pressure sensor as a result of the maximum possible pressure applied by the manipulator.

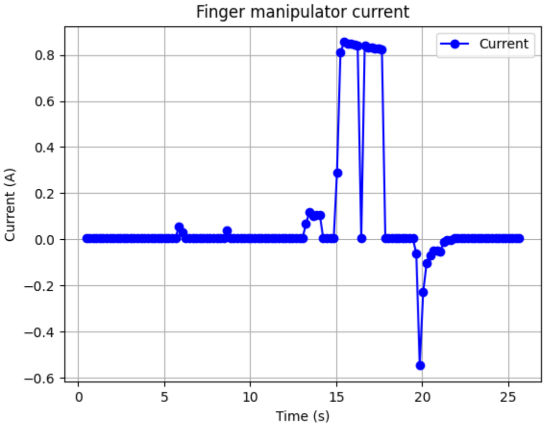


Figure 5.22: Graph of the maximum current obtained by the end-effector fingers as a result of the maximum possible pressure applied by the manipulator.

It can be seen that in the characterization graph of the pressure sensor, a range of 0 to almost 5 V is obtained for the maximum pressure. On the other hand, in the graph of the current of the fingers of the manipulator, a current slightly above 0.8 A is obtained. Taking these values as a reference, fine manipulation movements were performed directly on the 3D hand model, obtaining the following representative results.

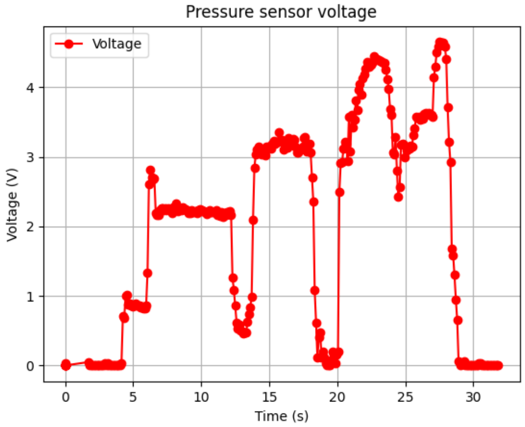


Figure 5.23: Graph of voltages obtained by the pressure sensor during the development of fine hand manipulation.

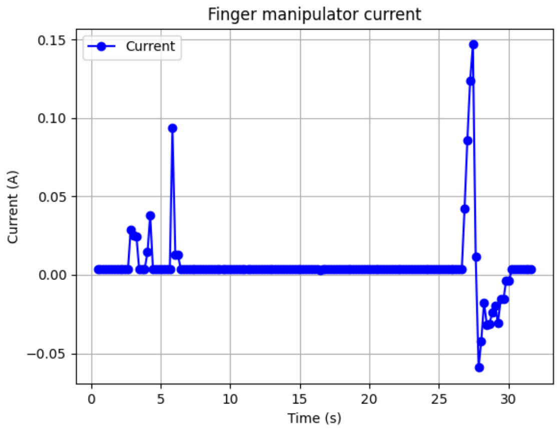


Figure 5.24: Graph of the currents obtained by the terminal effector fingers during the development of the fine manipulation of the hand, current peaks can be observed, which correspond to pressure points exerted by the robotic fingers during the manipulation action.

In the graph of Figure 5.23, it can be observed the value of the voltage that is registered by the pressure sensor during the fine manipulation action. In turn in the graph of Figure 5.24, the current value of the manipulator's fingers during the fine manipulation action is shown. In this example, the voltage value has variations throughout the fine manipulation trajectory because the pressure sensor is always in contact with one of the fingers of the end effector. On the other hand in the current graph, only the current value of the manipulator fingers is recorded during the contact that the manipulator finger makes with the sensor while the action of lifting the phalanx of the 3D model is performed.

5.3 Orientation Alignment validation

The orientation between the manipulator and the hand is crucial for the safety of the rehabilitation system, as incorrect alignment can cause direct harm to the limb treated or damage the rehabilitation equipment. Therefore, it is essential to validate this safety skill continuously whenever hand manipulation is performed, in addition to the experimental validation conducted for other safety skills.

Taking into account that our objective is to estimate the position and orientation of the hand, the spatial estimation of the hand will be divided into 3 parts. First, the rotation of the whole hand concerning the camera position was estimated. For this, the points of interest provided by MediaPipe that correspond to the phalanges of the index finger was taken as reference, from the interproximal intersection to the tip of the finger. Considering the interproximal intersection, a line is drawn that passes through that point and is drawn directly on the horizontal plane of the camera. Then, a straight line is drawn on the 3 phalanges of the index finger, and finally, as shown in the figure 4.15, the value of the angle between the two vectors is calculated and the rotation of the hand is obtained in 2D with respect to the camera position. The second part corresponds to calculating the inclination of the hand in any of the 4 directions that the hand can take or any combination of these. For this, the points provided by MediaPipe will be taken as reference, corresponding to the fingertips and interproximal connections of both the index finger and the middle finger, forming a reference plane as shown in figure 4.18. In this way, the difference between each of the vertices was calculated and the angle of inclination of the edges was obtained. Finally, for the last part, the angle of inclination of the target finger is calculated, in this case, the index finger. In such a way, the reference points provided by MediaPipe and corresponding to the interproximal connection and the tip of the finger are taken into account once again, the difference between each of these points is obtained and the angle of inclination is finally obtained.

The 3 parts of the process to obtain the position and orientation of the hand have been divided in that order, to first ensure the orientation and position of the complete hand and finally, with this information, to be able to calculate the necessary grip angle for the robotic manipulator.

For the results of the first part, the 3D hand model was placed in front of the depth camera, and a full rotation was performed to observe the angle between the reference vectors mentioned above.

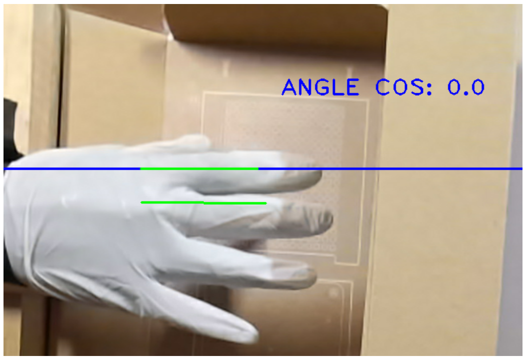


Figure 5.25: Experimental results during hand rotation. It can be observed that the hand model in this image at an angle of 0 degrees, when the hand is oriented with respect to the horizontal pointing the fingers to the right.

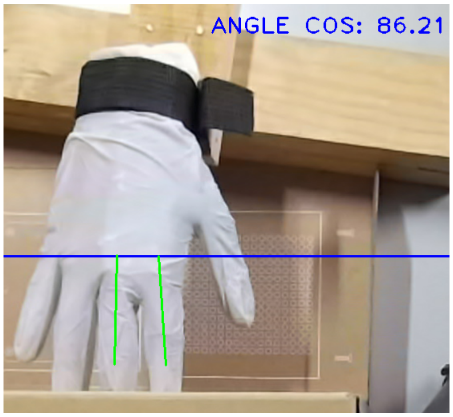


Figure 5.26: Experimental result during hand rotation, an angle of 86 degrees can be observed in this image, approximately 90 degrees, with the fingertips oriented downward with respect to the horizontal.

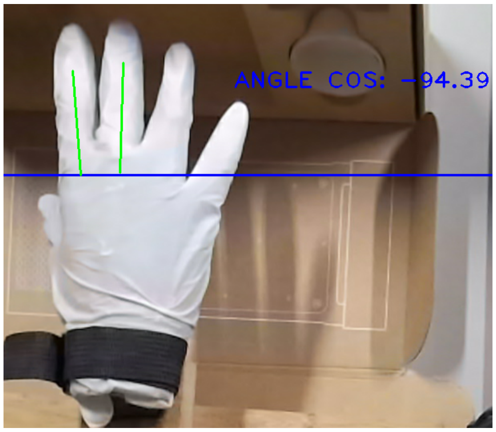


Figure 5.27: Experimental result during hand rotation, an angle of -94 degrees can be observed in this image, with the fingertips oriented upwards with respect to the horizontal.

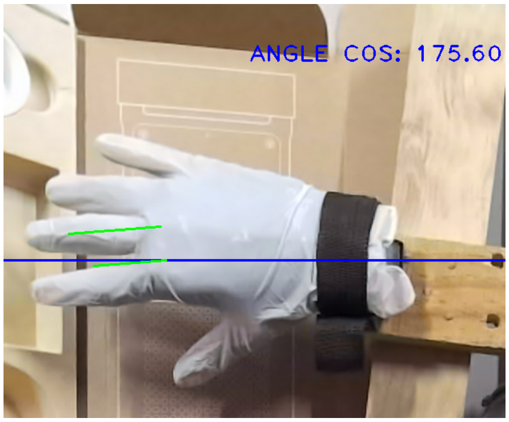


Figure 5.28: Experimental result during hand rotation, an angle of 175 degrees can be observed in this image, approximately 180 degrees, with the fingertips oriented to the left with respect to the horizontal.

In Figures 5.25 to 5.28, it can be seen the different positions that the hand can adopt, By taking as a reference the position, the orientation of the hand can be obtained. With this information, the system is able to determine wherer or not it is safe to perform the fine manipulation task.

For the second part, the 3D hand model was placed with the fingertips fully extended and pointing directly to the top of the image, then, slight tilting was performed in 4 different orientations, elevating the fingertips (Figure 5.30), elevating the wrist (Figure 5.29), tilting the hand to the (Figure right 5.32) and finally tilting the hand to the left (Figure 5.31).

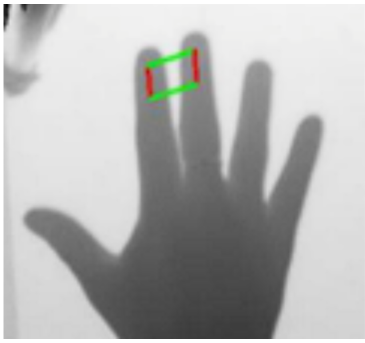


Figure 5.29: Experimental result of the planar orientation of the hand, two vertical red lines can be observed representing the downward orientation of the fingertips, referring to an angle less than 0 with respect to the difference between the fingertips and the interproximal phalanges.

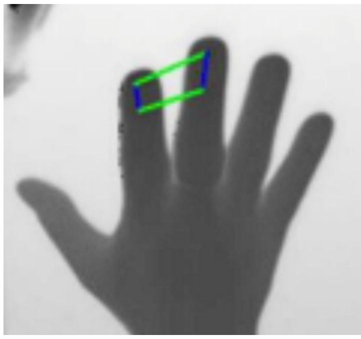


Figure 5.30: Experimental result of the planar orientation of the hand, two vertical blue lines can be observed representing the upward orientation of the fingertips, referring to an angle greater than 0 with respect to the difference between the fingertips and the interproximal phalanges.

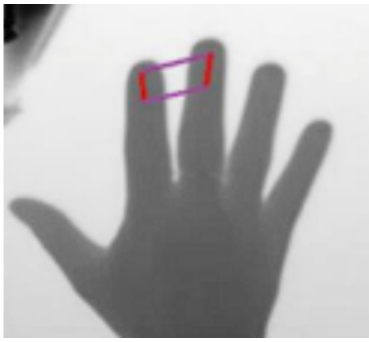


Figure 5.31: Experimental result of the planar orientation of the hand, two horizontal purple lines are observed representing the upward orientation of the little finger, referred to an angle less than 0 with respect to the difference between the tips of the index and middle fingers.

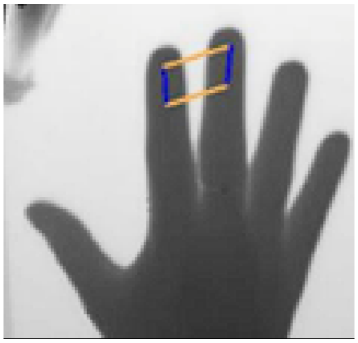


Figure 5.32: Experimental result of the planar orientation of the hand, two horizontal yellow lines are observed representing the upward orientation of the thumb finger, referred to an angle greater than 0 with respect to the difference between the tips of the index and middle fingers.

Finally, for the last part, the 3D hand model was placed by pointing the fingers toward the top of the image. All fingers were fully extended and specific inclinations were performed at 10° (Figure 5.34), 20° (Figure 5.35) and 30° (Figure 5.36), taking as reference the interproximal connection and the tip of the index finger.

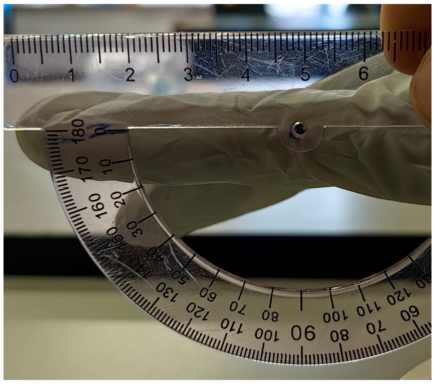


Figure 5.33: Reference position of the index finger with an inclination of 0 degrees.

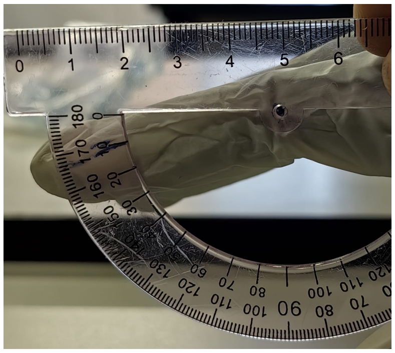


Figure 5.34: Reference position of the index finger with an inclination of 10 degrees.

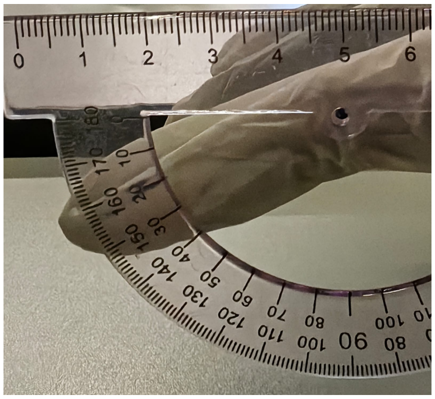


Figure 5.35: Reference position of the index finger with an inclination of 20 degrees.

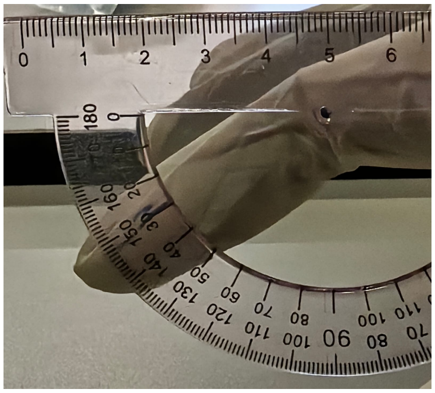


Figure 5.36: Reference position of the index finger with an inclination of 30 degrees.

The results of the last two parts of the process to estimate the position and orientation of the hand were performed by taking into consideration 3 methods, combining the use of MediaPipe and the grayscale information from the depth camera. In the first scenario, the RGB image is used, applying the MediaPipe processing and extracting the depth information provided by it, which is obtained in the form of a floating unit that calculates the depth based on the position of the other points and the hand pose estimation system. In the next scenario, the grayscale image is taken from the depth camera and only the position information in the 2D image is used, from there the grayscale value of the centroids of interest is obtained to determine the depth of that specific point within the image. Finally, the grayscale image is taken again, the MediaPipe processing is performed and in the same way as in the first scenario, the depth value provided by MediaPipe is obtained. The results of the following 3 scenarios were evaluated by placing the hand in different positions, taking into account that no part of the 3D model was out of the camera's view.

Angle (Degrees)	MediaPipe (float 64)	Depth (gray scale)	MediaDepth (float 64)
0	0.0092	4.3	0.016
10	0.0075	2.99	0.015
20	0.0066	2.92	0.014
30	0.0058	-0.2	0.013

Table 5.1: Comparison of the methods implemented to determine the relationship between the angle of inclination of the finger and the estimation of the depth difference.

After performing several tests of the different inclinations that the fingers of the hand model could take in different positions within the vision of the camera, an average was obtained and the results are presented in the Table 5.1. It can be seen that there is greater accuracy and difference between the reference values obtained through the first method, using the hand depth estimation data by MediaPipe with the RGB image as there is a greater difference and a consistent change by gradually increasing the inclination angle, the third scenario has a similar result like the first one. However, there is a small difference between the gradual changes of inclination, which would not allow having an accurate result in case of inclination changes of less than 10 degrees. Finally, the second scenario obtained the worst results by obtaining disproportionate changes with respect to the gradual change of inclination.

5.4 Limit Motion Energy validation

The direct manipulation between the robotic manipulator and the 3D hand model should be performed taking into consideration that the action of moving one or more robotic fingers should not be at any time a harmful action for the patient's hand. Thus, the use of an impedance-based control is proposed, for its advantages during the use in soft tissues. However, the manipulation tests were performed in 3 control methods: ROS topic control, PID control, and impedance control.

The ROS topic control corresponds to the speed control of the movement of the manipulator's joints during the fine manipulation action. For this, a program was developed that takes the topics with information on the position of the manipulator's joints and captures the signal to send the speed of the joints in charge of the manipulation action, the module of the forces resulting from the vector of forces of the end effector was calculated and the following result were obtained, as depicted in Figures 5.37 to 5.39.

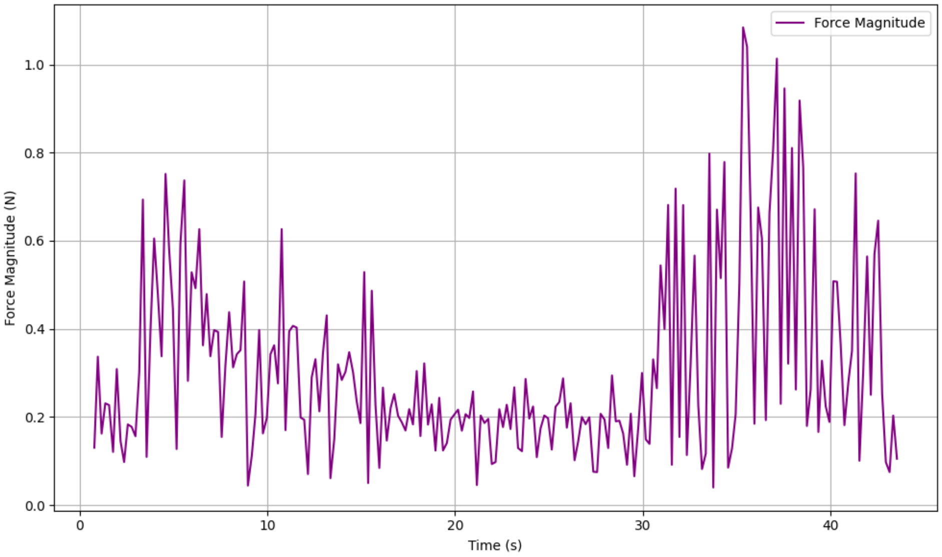


Figure 5.37: Force plot during fine hand control trajectory using low-level control.

The PID control was integrated as an algorithm that takes the position information of the manipulator's joints through ROS topics and calculates the movement points that the trajectory of the joints in charge of the fine manipulation must follow. The modulus of the resulting forces of the end-effector force vector was calculated and the result was plotted in Figure 5.38.

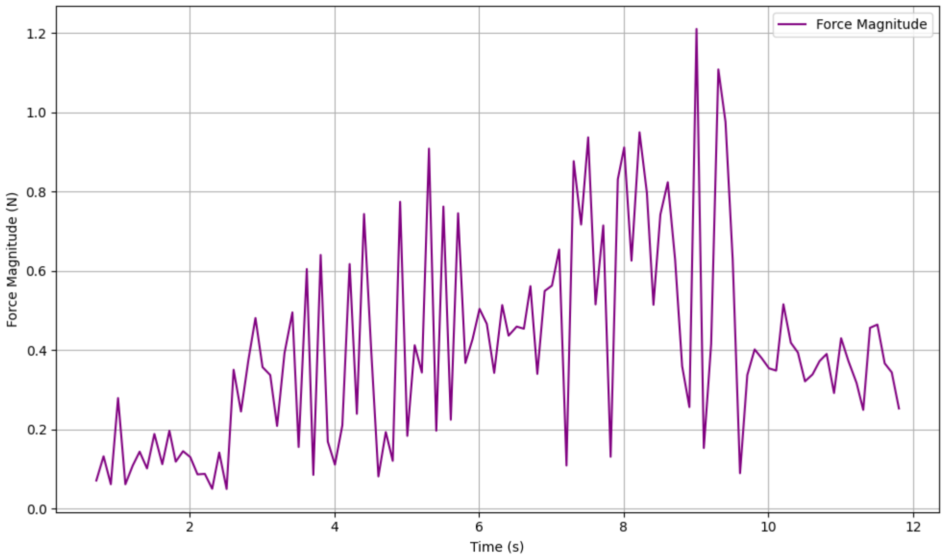


Figure 5.38: Force plot during fine hand control trajectory using PID control.

Finally, the impedance control was implemented through a program that takes the target or expected position of the manipulator concerning the current position, calculates the points of the trajectory according to the impedance control and sends the signal through the manipulator control via Moveit, calculates the modulus of the forces resulting from the force vector of the end effector and graphs the result in Figure 5.39.

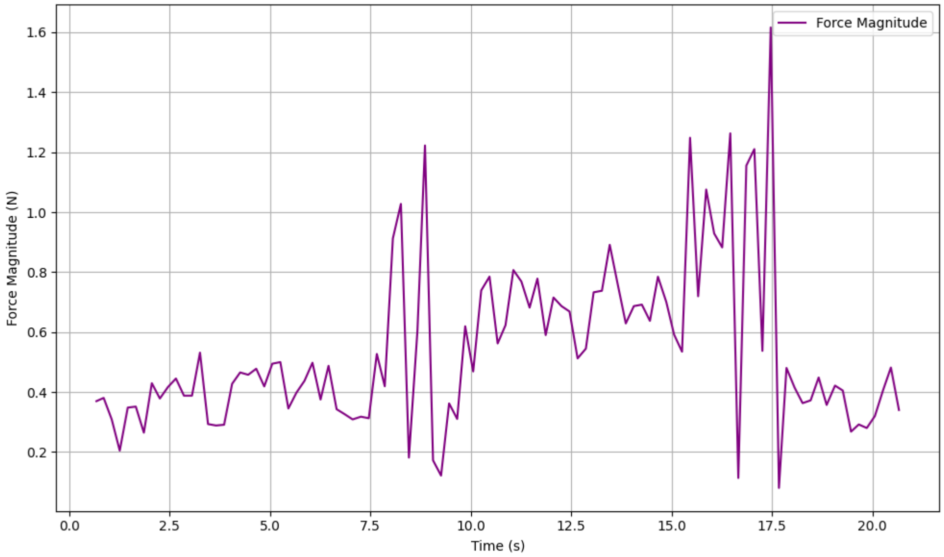


Figure 5.39: Force plot during fine hand control trajectory using impedance control through Moveit.

The above graphs represent the results of the end-effector force modulus during the fine manipulation action with the 3 different controls mentioned above. In the last case, for the impedance control, it can be observed that there is a less abrupt change between the force values over time and likewise, a smoother trend can be observed during the fine manipulation action (figure 5.39).

5.5 Stop Emergency System validation

The emergency button system is implemented through ROS services, which are already integrated into the control node of the robotic manipulator through the ROS connection node. To stop a manipulator trajectory, you must be subscribed to the service and send the “Stop” message, when it is required to completely stop the system, this service allows blocking the signals that are sent to the manipulator. To validate this security ability, being a message that interacts directly with the connection between the manipulator and the commands or signals that can be sent through the different control methods of the manipulator, a table is presented that evaluates if the emergency button actuated through the manipulator services stops its movement, during the control through different methods. I this case, the control methods evaluated are: Moveit (Python), ROS topics (low-level control), Moveit plugin (Rviz) and API, being these the only control methods with which we worked during this project with the proposed experimental model.

Manipulator Control	Emergency Stop Trajectory
Moveit (Python)	YES
Moveit (Rviz)	YES
ROS topic (Python)	YES
API	YES
Moveit and ROS topics (Python)	YES

Table 5.2: Table of results during general trajectory interruption in some of the control cases used during the process of experimentation and validation of safety skills.

In the table above it can be seen the different types of control that were developed during the experimental tests of the proposed model and the result of the emergency button to stop the trajectory of the manipulator. In all cases the use of the emergency button was satisfactory, except in the case of the Joystick, this being an analog signal sent by the physical Joystick to the manipulator, the emergency button manages to cut the communication of the control with the manipulator for a brief moment if the Joystick is still actuated in some direction after the emergency stop. The manipulator can continue its trajectory because the Joystick continues sending signals. In the other cases, this does not happen because the program does not continue sending signals.

5.6 Summary

In summary, this chapter presented the validation of safety skills, where each section proposes the method to be used for experimental development and the specific validation guideline for each safety skill. Being safety skills in charge of evaluating different aspects such as position, orientation, and relationship of the manipulator with the 3D hand model, each one requires a different experimentation and validation process. It can be observed within the results the degree of accuracy that each one of the safety skills has and the risk factor that they can avoid both together and individually. In this aspect, the position and orientation of the 3D hand model are approximated with the safety skills of a range of motion limit and alignment and orientation. The strength and safe manipulation of the phalanx is determined with the help of the limit of motion and physical energy. Finally in case of emergency and whenever the patient considers it, any of the actions or movements of the manipulator can be stopped at any time, with the help of the emergency button.

Chapter 6

Conclusions and Future Work

The increase in the demand for rehabilitation therapies for patients who have suffered a stroke has generated a great workload for the specialists in charge of this area. However, this has also led to the development of new rehabilitation systems, which allow the same health specialists to have a greater number of tools to address the problem of high demand for rehabilitation therapies. In such scenario, the robotic rehabilitation systems have proven to compete with traditional rehabilitation systems, improving their safety during the development of human-robot interaction.

During the development of this project, the general and specific objectives were achieved, obtaining a method to estimate the pose of the hand, planing the trajectory for fine control of robotic manipulators, conducting tests on 3D-printed models equipped with sensors and integrating a validation protocol to ensure the safety of the system. An experimental model was developed based on the proposed safety skills, which were employed to obtain satisfactory results for the validation of the safety protocol. Likewise, the different methods and scenarios proposed were explored to improve the results of positioning, orientation, and the relationship between the robotic manipulator and the 3D hand model. As a whole, each of the safety skills allowed exploring the different risk factors present in robotic rehabilitation systems and the importance of their implementation during the development of such systems that seek to carry out some type of rehabilitation using general-purpose manipulators.

6.1 Limitations

Some of the limitations of this project mainly include the lack of time to develop experiments on real patients and thus concretely determine the direct benefit on the range of mobility that stroke patients may have in their different degrees of musculo-nervous affectation. This system contemplates the use of a manipulator and only evaluates risk factors caused by the manipulator itself, without taking into account human error or misuse of the tools. The use of a specific manipulator during the development of this project limits the possibility of using tools or resources available for any other type of manipulator of general or specific use, making the development of this project focus only on the specifications and tools available for the Jaco robotic manipulator model manufactured by Kinova.

6.2 Future Work

In future work, the implementation of two robotic manipulators can be considered: one in charge of the task of fine manipulation of the hand and the other to mimic the interaction that a patient would normally have with the health specialist, by holding his arm, placing it in the position desired by the specialist and preventing from making any movement that would be harmful to the patient during the development of the fine manipulation of the hand or fingers. Extending the safety protocol to evaluate more risk factors, such as human error or direct failure of the physical components, such as the manipulator or the depth chamber. With this project, we hope to lay the foundations to extend the concept of safe human-robot interaction protocol in other robotic rehabilitation systems and in the same way continue proposing experimental scenarios with real patients, to obtain a more robust system capable of developing rehabilitation therapy in extremities in a general way.

References

Abdullahi, A. M., Haruna, A., & Chaichaowarat, R. (2024). Hybrid adaptive impedance and admittance control based on the sensorless estimation of interaction joint torque for exoskeletons: A case study of an upper limb rehabilitation robot. *Journal of Sensor and Actuator Networks*, *13*(2), 24.

Baek, S., Kim, K. I., & Kim, T.-K. (2019). Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 1067–1076).

Bertani, R., Melegari, C., De Cola, M. C., Bramanti, A., Bramanti, P., & Calabrò, R. S. (2017, May). Effects of robot-assisted upper limb rehabilitation in stroke patients: A systematic review with meta-analysis. *Neurological Sciences*, *38*(9), 1561–1569. doi: 10.1007/s10072-017-2995-5

Bessler, J., Prange-Lasonder, G. B., Schaake, L., Saenz, J. F., Bidard, C., Fassi, I., … Buurke, J. H. (2021, Mar). Safety assessment of rehabilitation robots: A review identifying safety skills and current knowledge gaps. *Frontiers in Robotics and AI*, *8*. doi: 10.3389/frobt.2021.602878

Bradski, G., & Kaehler, A. (2015). *Learning opencv.* O'Reilly Media.

Chen, W., Yu, C., Tu, C., Lyu, Z., Tang, J., Ou, S., … Xue, Z. (2020, Feb). A survey on hand pose estimation with wearable sensors and computer-vision-based methods. *Sensors*, *20*(4), 1074. doi: 10.3390/s20041074

Esquivel-Alvarado, R. J. (2018). Estimación de pose de manos bajo consideración de relaciones topológicas.

Fasoli, S. E., Krebs, H. I., Stein, J., Frontera, W. R., Hughes, R., & Hogan, N. (2004). Robotic therapy for chronic motor impairments after stroke: Follow-up results. *Archives of physical medicine and rehabilitation*, *85*(7), 1106–1111.

Gasparetto, A., & Scalera, L. (2019). A brief history of industrial robotics in the 20th century. *Advances in Historical Studies*, *08*(01), 24–35. doi: 10.4236/ahs.2019.81002

Grushko, S., Vysocky, A., Jha, V. K., Pastor, R., Prada, E., Miková, L., & BOBOVSKY, Z. (2020). Tuning perception and motion planning parameters for moveit! framework. *MM Sci. J*, *2020*, 4154–4163.

Han, X., Guo, N., Jie, Y., Wang, H., Wan, F., & Song, C. (2024). On flange-based 3d hand–eye calibration for soft robotic tactile welding. *Measurement*, *238*, 115376.

Kóczi, D., & Sárosi, J. (2022, May). *(pdf) the safety of collaborative robotics -a review.* Retrieved from https://www.researchgate.net/publication/363298471_THE_SAFETY_OF_COLLABORATIVE_ROBOTICS_-A_REVIEW

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., … others (2019). Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*.

Lum, P. S., Burgar, C. G., Loos, M. V., Shor, P. C., Majmundar, M., & Yap, R. (2006). Mime robotic device for upper-limb neurorehabilitation in subacute stroke subjects: A follow-up study. *The Journal of Rehabilitation Research and Development*, *43*(5), 631. doi: 10.1682/jrrd.2005.02.0044

Moran, M. E. (2007, May). Evolution of robotic arms. *Journal of Robotic Surgery*, *1*(2), 103–111. doi: 10.1007/s11701-006-0002-x

Narayan, J., Kalita, B., & Dwivedy, S. K. (2021). Development of robot-based upper limb devices for rehabilitation purposes: a systematic review. *Augmented Human Research*, *6*, 1–33.

Perry, J. C., Rosen, J., & Burns, S. (2007, Aug). Upper-limb powered exoskeleton design. *IEEE/ASME Transactions on Mechatronics*, *12*(4), 408–417. doi:

10.1109/tmech.2007.901934

Proietti, T., Crocher, V., Roby-Brami, A., & Jarrasse, N. (2016). Upper-limb robotic exoskeletons for neurorehabilitation: a review on control strategies. *IEEE reviews in biomedical engineering*, *9*, 4–14.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., … others (2009). Ros: an open-source robot operating system. In *Icra workshop on open source software* (Vol. 3, p. 5).

Ren, Y., Kang, S. H., Park, H.-S., Wu, Y.-N., & Zhang, L.-Q. (2012). Developing a multi-joint upper limb exoskeleton robot for diagnosis, therapy, and outcome evaluation in neurorehabilitation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *21*(3), 490–499.

Rusu, R. B., & Cousins, S. (2011). 3d is here: Point cloud library (pcl). In *2011 ieee international conference on robotics and automation* (p. 1-4). doi: 10.1109/ICRA.2011.5980567

Sheng, B., Zhang, Y., Meng, W., Deng, C., & Xie, S. (2016). Bilateral robots for upper-limb stroke rehabilitation: State of the art and future prospects. *Medical engineering & physics*, *38*(7), 587–606.

Thayabaranathan, T., Kim, J., Cadilhac, D. A., Thrift, A. G., Donnan, G. A., Howard, G., … et al. (2022, Sep). Global stroke statistics 2022. *International Journal of Stroke*, *17*(9), 946–956. doi: 10.1177/17474930221123175

Tölgyessy, M., Dekan, M., Chovanec, L., & Hubinskỳ, P. (2021). Evaluation of the azure kinect and its comparison to kinect v1 and kinect v2. *Sensors*, *21*(2), 413.

Xie, C., Yang, Q., Huang, Y., Su, S., Xu, T., & Song, R. (2021, Dec). A hybrid arm-hand rehabilitation robot with emg-based admittance controller. *IEEE Transactions on Biomedical Circuits and Systems*, *15*(6), 1332–1342. doi: 10.1109/tbcas.2021.3130090

Zohour, H. M., Belzile, B., & St-Onge, D. (2021). Kinova gen3-lite manipulator inverse kinematics: optimal polynomial solution. *arXiv preprint* *arXiv:2102.01217*.