METHODOLOGY FOR DESIGNING ASSISTIVE ROBOTS FOR ACTIVITIES OF DAILY LIVING ASSISTANCE

by

JAVIER DARIO SANJUAN DE CARO

A Dissertation Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

in Engineering

at

The University of Wisconsin-Milwaukee

August 2023

ABSTRACT

METHODOLOGY FOR DESIGNING ASSISTIVE ROBOTS FOR ACTIVITIES OF DAILY LIVING ASSISTANCE

by

Javier Dario Sanjuan De Caro

University of Wisconsin-Milwaukee, 2023
Under the Supervision of Professor Mohammad H Rahman

The growing prevalence of Upper or Lower Extremities Dysfunctions (ULED), often linked to central nervous disorders such as stroke, Spinal Cord Injury (SCI), and Multiple Sclerosis (MS), underscores the urgent need for innovative support solutions. Over 5.35 million Americans currently live with ULED, a situation that places a significant socioeconomic burden on families and society. Despite invaluable support from caregivers and family members, the need for more scalable, practical solutions persists.

Wheelchair-mounted assistive robots emerge as a promising alternative in this context. These devices, offering continuous and reliable assistance, significantly alleviate caregiver fatigue and enhance the independence and quality of life for individuals with ULED. Capable of ceaseless operation and providing assistance with various Activities of Daily Living (ADLs), these robots foster an unprecedented level of autonomy for those affected by ULED.

This dissertation initiates with the presentation of core design constraints for assistive robots, derived from a thorough analysis of user needs and requirements. These constraints include the necessity of custom workspace designs for facilitating essential ADLs, the impor-

tance of foldability in the design of assistive robots, and an assessment of the robot's weight handling capacity, set to a benchmark of 3 kg. These elements collectively underscore the delicate balance required in assistive robot design, which calls for improved maneuverability, efficient energy consumption, accommodation of ADLs, optimal space usage, and appropriate weight handling capacity.

With these constraints as a foundation, the dissertation advances a methodology for the optimal design of assistive robots. Central to this methodology is the maximization of workspace coverage, a vital aspect for effectively enabling ADLs. In service of this aim, ADLs are classified into seven distinct workspaces, creating a nuanced understanding of the diverse physical environments these activities entail.

To further augment the design methodology, the dissertation introduces HUNTER, an advanced inverse kinematics algorithm. Capable of handling singularities and providing solutions beyond the predefined workspace, HUNTER deepens the interaction between the robot and the user. This advancement, in turn, amplifies the versatility and safety of assistive robots and holds significant potential for improving the user experience.

Following the establishment of the methodology and the introduction of HUNTER, the dissertation details the testing of the optimal robot through the implementation of a robot control library in ROS2. This library incorporates the inverse kinematics of HUNTER, controls the motors using EtherCAT (a communication protocol that enhances speed and secures real-time implementation), and applies a collision detection algorithm for the user's safety.

Finally, the dissertation covers the testing of the robot, validating its performance under the previously established design constraints. This comprehensive approach to design, testing, and validation underscores the potential for more effective and versatile assistive robotic solutions for individuals living with ULED.

TABLE OF CONTENTS

LIST OF FIGURES . . . . . **xii**

LIST OF TABLES . . . . . **xiii**

1 INTRODUCTION **1**

INTRODUCTION **1**

1.1 Significance of the Problem **1**

1.2 Existing Solutions and Their Shortcomings **2**

1.3 Proposed solution **5**

1.4 Structure of the dissertation **6**

2 Assistive Robots and ADLs: Exploring Design Constraints **10**

2.1 Assistive Robotics: Unpacking Current Trends and Design Criteria . **11**

2.1.1 6-DOF 11

2.1.2 7-DOF 17

2.1.3 8-DOF 20

2.1.4 Identifying Challenges in Current Assistive Robot Design and Functionality . . 22

2.2 Establishing Design Constraints for Assistive Robots Based on ADL Requirements . . . 23
2.2.1 Using ADLs to Define Workspace in Assistive Robot Design . . . 24
2.2.2 Foldability: An Essential Design Constraint for Wheelchair-Mounted Assistive Robots . . . 28
2.2.3 Designing for a 3 kg Load: A Key Consideration in ARs . . . 30
2.3 Conclusions . . . 30
3 FORMULATING A NEW INVERSE KINEMATICS ALGORITHM: THEORETICAL FOUNDATIONS AND APPROACH . . . 32
3.1 State-of-the-art numerical IK . . . 34
3.1.1 Jacobian methods. . . . 35
3.1.2 Quasi-Newton methods . . . 40
3.2 HUNTER (Helping Users Navigate through Extended Reach): A Novel Algorithm for Inverse Kinematics in Assistive Robots. . . . 44
3.3 Conclusions . . . 49
4 ENHANCING MANIPULABILITY AND ENERGY EFFICIENCY: DEFINITION OF A NEW METHODOLOGY FOR DESIGNING ASSISTIVE ROBOTS . . . 50
4.1 Kinematic performance indices . . . 51
4.1.1 Kinematic Performance Based on the Jacobian Matrix . . . 51
4.1.2 Kinematic performance indices based on screw theory . . . 54
4.2 Revising Manipulability: Introduction of a New Index . . . 56
4.2.1 Rotation analysis of universal joints . . . 57

4.2.2 Experimental validation 60
4.2.3 Jacobian matrix calculation 68
4.2.4 New manipulability index 71
4.3 Energy consumption indices 72
4.3.1 Quadratic average torque (QAT) 72
4.3.2 Weighted Root Mean Square (WRMS) 72
4.3.3 The Absolute Sum of Torques (AST) 73
4.3.4 Selecting the proper ECIs 73
4.4 Methodology for the integration of manipulability and energy consumption indices in the design of assistive robots. 74
4.5 Conclusions 79
5 CRAFTING PERFECTION: THE OPTIMAL DESIGN OF AN ASSISTIVE ROBOT 82
5.1 Assistive robot description 83
5.1.1 Kinematics 85
5.1.2 Folding Criteria 87
5.1.3 Dynamic model 89
5.2 Algorithm for Collision Detection. 95
5.3 Scatter Search: The Optimal Choice for Robot Optimizatio 100
5.3.1 Scatter Search for Assistive Robot Link Length Optimization 102
5.4 Validation through Simulation 104
5.4.1 Workspace Coverage Verification 105
5.4.2 Foldability Verification 106

5.5 Dynamic Validation . . . 107
5.6 conclusions . . . 110
6 DEVELOPMENT OF THE SOEM-BASED LIBRARY FOR ASSISTIVE ROBOT MANIPULATION . . . 112
6.1 Review of Existing SOEM-based Libraries . . . 115
6.2 Overview of the SOEM-based library development . . . 116
6.3 Requirement identification . . . 117
6.4 Library Architecture . . . 118
6.4.1 Control interface GUI . . . 119
6.4.2 Robot control Libraries . . . 120
6.4.3 Kinematics library . . . 121
6.4.4 Collision avoidance library . . . 122
6.5 Conclusion . . . 124
7 FROM THEORY TO PRACTICE: TESTING THE ASSISTIVE ROBOT 128
7.1 Workspace Validation . . . 129
7.1.1 Workspace 1 . . . 130
7.1.2 Workspace 2 . . . 131
7.1.3 Workspace 3 . . . 132
7.1.4 Workspace 4 . . . 133
7.1.5 Workspace 5 . . . 134
7.1.6 Workspace 6 . . . 135
7.1.7 Workspace 7 . . . 137

7.2 Folding Mechanism Assessment 138

7.3 HUNTER Performance Evaluation 140

7.3.1 Picking an object from a shelf 140

7.3.2 Picking object from the floor 142

7.3.3 Picking object from the table 143

7.4 Load-Bearing Capacity Verification 144

7.5 Conclusions 145

References 147

Appendix 166

LIST OF FIGURES

2.1 KARES II, a wheelchair robot system. Image reproduced from [1] 12

2.2 Weston robot attached to Strom wheelchair. Image reproduced from [2] 14

2.3 MANUS: a wheelchair mounted rehabilitation robot. Image reproduced from [3] 15

2.4 JACO mounted on a power wheelchair. Image reproduced from [4] 16

2.5 Rehabilitation robot FRIEND II. Image reproduced from [5] 18

2.6 WMRA in a feeding pose, low-right reach, left side doorknob. Image reproduced from [6] 19

2.7 EDAN system including a closeup of the wheel-encoders, the 8-DOF robot, the head-switch, the RGB-D camera, and the tablet interface. Image reproduced as presented in [7]. 21

2.8 Experimental setup to measure workspace of the ADLs. 26

2.9 Representation of the workspaces of each ADLs relative to the robot and the user. 26

2.10 Dimensions and location of each ADL workspace, including the orientation of the end-effector relative to O. 27

2.11 Schematic Configuration Illustrating the Foldable State of the Assistive Robot Design. 29

3.1 Graphical User Interface (GUI) offering comprehensive control of the virtual assistive robot for evaluating the effectiveness of HUNTER in managing singularities. 47
3.2 Demonstration of HUNTER's Capability in Navigating Singular Positions and Boundaries of Workspace. 48

4.1 Link attached to an universal joint 57
4.2 Prototype for experimentation 61
4.3 Kinematic diagram of the prototype for experimentation presented in Fig 4.2 . . 63
4.4 Comparison between the trajectory used in the experiment and the lengths computed by inverse kinematic for: a) actuator 1, b) actuator 2, and c) actuator 3 66
4.5 a) Measured angles from the sensor. b) Equivalent U-joint angles. 67
4.6 Comparison between the sensed local angular velocity and the analytical local angular velocity in the a) x-direction, b) y-direction, and c) z-direction. 69
4.7 Methodology for Workspace Evaluation: Reducing Torque Consumption, Increasing Coverage, and Minimizing Object Collisions and Singularities. 75

5.1 wheelchair-mounted assistive robot (a) CAD (b) Joint Coordinate Definition. . . 83
5.2 Schematic Representation of the Assistive Robot in Folded State with Dimensional Parameters 88
5.3 Free body diagrams of the simplified links (a) Link 1 (b) Link 2 (c) Link 3. . . . 91
5.4 (a) OBB representation, (b) The distance between two OBBs is less than the total of their projected radii for some axis L 95

5.5 Representation of the robot with OBBs. Each link of the robot is encapsulated by a box that dynamically adjusts with the changes in the robot's configuration. 98
5.6 Optimized Assistive Robot: This figure illustrates the final design of the assistive robot with link lengths derived from the optimization process. The optimized design aims for maximum workspace while minimizing energy consumption. . . . 103
5.7 Comparison of the Accessible Workspace Covered by the Optimal Assistive Robot with the Total Designated Workspaces 106
5.8 CAD Model Illustrating the Folded State of the Optimal Assistive Robot 107
5.9 Simscape Multibody model of the optimal assistive robot 108
5.10 Torque workspace coverage of the optimal assistive robot 109

6.1 Library Architecture for the Assistive Robot. 126
6.2 The Graphical User Interface (GUI) developed for controlling the assistive robot. 127

7.1 Evaluation Results for Workspace 1 130
7.2 Evaluation Results for Workspace 2 132
7.3 Evaluation Results for Workspace 3 133
7.4 Evaluation Results for Workspace 4 134
7.5 Evaluation Results for Workspace 5 135
7.6 Evaluation Results for Workspace 6 136
7.7 Evaluation Results for Workspace 7 137
7.8 Evaluation Results of Folding Mechanism 139
7.9 Evaluation Results for Inverse Kinematics in Shelf Scenario 141
7.10 Evaluation Results for Inverse Kinematics in Floor Scenario 142

7.11 Evaluation Results for Inverse Kinematics in Table Scenario 143
7.12 Load Handling Evaluation 144

LIST OF TABLES

2.1 Timeline of the assistive robots classified according to the DOF. 22

2.2 Categorization of the listed essential ADLs. 25

4.1 Values of θ_x, θ_y, and time for the experimentation trajectory. 62

4.2 Coefficients for quintic function 63

5.1 DH Parameters of the Assistive Robot 84

5.2 Definition of each Oriented Bounding Box (OBB) as a function of its corresponding link length. Each box definition includes the HTM representing the box's orientation and position relative to the link's coordinate frame, as well as the box's dimensions 99

5.3 Optimization Parameters for Scatter Search: This table lists the parameters, their ranges, and constraints used in the scatter search optimization process for determining the optimal link lengths of the assistive robot. 102

5.4 Results of the optimization 104

5.5 Joint limits of the purposed assistive robot 104

5.6 Motor References for Each Joint of the Robot 108

CHAPTER 1

INTRODUCTION

1.1 Significance of the Problem

Upper or Lower Extremities Dysfunctions (ULED) are frequently the outcome of central nervous disorders such as stroke, Spinal Cord Injury (SCI), and Multiple Sclerosis (MS). These conditions impact 5.35 million Americans, contributing to ULEDs [8, 9]. Stroke is the most common culprit, accounting for 33.7% of ULED cases in the US, followed by SCI (27.3%), and MS (18.6%). SCI-related ULEDs primarily arise from motor vehicle accidents (27%), physical labor (21%), and falls (11%) [9]. Merely 15% of individuals with ULED remain employed, with over 40% deemed incapable of working [8]. Furthermore, approximately 3.6 million individuals with ULED use a wheelchair, a statistic that has doubled in the past decade, with the population increasing at an alarming annual rate of 5.9%. The treatment of central nervous disorders leading to ULED burdens the healthcare system with billions of dollars annually, with SCI alone accounting for 40 billion dollars [8]. Hence, the long-term support required by most individuals with ULED has profound social and economic implications.

Existing aids for individuals with ULED primarily consist of caregivers and assistive devices.

However, it has been revealed that 41 million caregivers provide more than 30 billion care hours [10], incurring substantial costs for the families of those affected by ULED. Moreover, despite the majority of individuals with ULED relying on wheelchairs, over 90% of wheelchair users report limitations in their Activities of Daily Living (ADL) [11]. The capacity to independently take care of oneself is vital for those with ULED, motivating researchers to develop assistive devices targeting specific tasks like mobility [12], feeding [13], and transferring [14]. Among these devices, wheelchair mounted Assistive Robots are prominent due to their versatility in satisfying a broad range of ADLs [15]. Examples include the JACO Arm by Kinova Robotics [16], and the EDAN system by DLR Robotics [7]. These assistive robots can facilitate numerous ADLs and foster user interaction with their surroundings. Nonetheless, these robots necessitate high degrees of freedom (DOF) to manipulate objects, particularly near the patient, to ensure gripper manipulability and prevent collisions with the robot and the user [17]. For instance, the JACO Arm has 7-DOFs, while the EDAN system has 8-DOFs. Other systems like FRIEND IV [18] and the University of Florida's assisitive robot also employ 7-DOFs. However, a high DOF count implies an increased number of actuators, which escalates energy consumption and weight of the manipulator [19], consequently limiting operating time and payload capacity. Therefore, the design and development of new Wheelchair-Mounted Assistive Robots capable of executing equivalent tasks with fewer DOFs is of pressing necessity.

1.2 Existing Solutions and Their Shortcomings

The design of a Wheelchair-Mounted assistive robot with fewer Degrees of Freedom (DOF), ensuring effective and efficient execution of movements—known as kinematic performance—within

its workspace, is an intricate optimization problem. A common approach to tackle this challenge involves developing an objective function rooted in the Jacobian matrix [20]. As the Jacobian matrix maps the output speed and the wrench of the end effector to the actuation joints, researchers have found that the measurement of the condition number of the Jacobian matrix indicates the kinematic capacities of the robot at a specific position and orientation [21, 22]. This methodology has been implemented in diverse studies such as the design of a Delta robot [23], the optimization of a 3-UPR parallel mechanism for a robotic leg [24], and the trajectory optimization of serial robots for drilling tasks [25]. Nevertheless, these methods necessitate a square matrix to compute the condition number. One approach to overcoming this obstacle is to use the Jacobian matrix's eigenvalues, a strategy successfully implemented in the optimization of a serial spherical mechanism for robotic surgery [26]. Critiques of this methodology, however, point out the issue of the Jacobian matrix's inhomogeneity, where columns may have different units leading to an inconsistent condition number [27]. Possible solutions include the use of screw theory with the dot product with the twist to ensure homogeneity of the Jacobian matrix [28], the normalization of the Jacobian using the characteristic length [29], or the application of an algebraic operator to eliminate the directions at which no velocity can be achieved from the Jacobian [30]. The precise formulation of the Jacobian matrix remains an active area of research.

Further manipulability measures employ dynamics to minimize the joint torques [31] and energy consumption [32]. Minimizing joint torques helps reduce end-effector vibrations [33, 34] while decreasing energy consumption can enhance device autonomy [35, 36]. Consequently, several authors have introduced various dynamic indices for torque and energy optimization. Despite the benefits, the application of these dynamic indices is largely limited to trajectory

optimization, and does not currently extend to workspace optimization.

Another major concern in workspace optimization is the avoidance of link collisions. Multiple design methodologies have been proposed to minimize object collision, such as limiting the assembly modes of the mechanism [37], increasing the robot's DOFs to ensure sufficient mobility for object avoidance [17], or clever positioning of the robot links to prevent collisions within the workspace [38]. Despite these advancements, collision avoidance remains an open research area. Currently, a comprehensive methodology that integrates collision detection into the kinematic design process is yet to be implemented.

Inverse kinematics (IK) poses another layer of complexity to the optimization problem. At certain positions, known as singularities, robots lose one or more degrees of freedom (DOF), which traditional IK methodologies struggle to handle. Consequently, users of assistive robots lose control at these singular positions, impacting the device's manipulability—the robot's ability to execute a broad spectrum of movements that enhances its functionality and versatility. Thus, devising an algorithm that consistently resolves the IK issue is crucial for the improved manipulability of assistive robots.

IK solution methods are primarily categorized into analytical and numerical. Analytical methods hinge on geometric or algebraic approaches and, despite their speed, lack a universal methodology to find the IK for all types of robots [39]. The IK-fast algorithm proposed by Rosen offers a semblance of a generalized IK solution, but it is unable to compute the IK of overly complex, redundant robots, or those with numerous constraints [40]. As a result, researchers often simplify the robot design to facilitate the calculation of IK. However, this approach often inadvertently increases the presence of singularities within the workspace [41]. On the other hand, numerical methods can apply to any robot and accommodate various types

of constraints, but they come with their own set of challenges, such as slow convergence speed and the accuracy of the solution.

To sum up, while different strategies to formulate the Jacobian matrix and solve the IK problem offer potential solutions, they often bring with them new challenges that must be overcome. This underscores the complexity of designing a wheelchair-mounted assistive robot that can deliver efficient and effective motion in its workspace, with minimized DOF, while avoiding singularities and ensuring consistent control. This multifaceted challenge remains a crucial area of ongoing research.

1.3 Proposed solution

The goal of this research is to establish a methodology for the optimal design of Assistive Robots. The proposed methodology encompasses the development of a novel inverse kinematics algorithm that functions effectively in real time, operates seamlessly in singularities, and computes solutions for points beyond the workspace. The approach also includes the design of a low DOF Wheelchair-Mounted assistive robot that can execute essential ADLs while minimizing energy consumption. The principal workspace is identified through customer discovery to include critical ADLs [42]. In addition, the methodology optimizes the link dimensions to minimize collisions and singularities within the workspace. The effectiveness of this proposed approach is validated by designing a 6-DOF assistive robot. The specific research objectives are as follows:

- *Aim 1:* To develop a novel inverse kinematics algorithm for assistive robots that enhances their capability

to perform ADL, with the distinctive features of effectively managing singularities and ensuring the algorithm's speed is sufficient for real-time application.

- *Aim 2:* To define a new integrated objective function for the design of assistive robots with the explicit goals of minimizing energy consumption, increasing workspace coverage, enhancing user safety, and enabling compact folding of the robot in its home position.

- *Aim 3:* To design an assistive robot with optimal link length, ensuring collision avoidance, compact folding, capable of handling a 3 kg load within a maximized workspace, and safeguarding motor capability.

- *Aim 4:* To develop an assistive robot, adhering to all design specifications, to fulfill ADL.

- *Aim 5:* To develop a SOEM-based library that integrates with ROS2 for EtherCAT motor control and assistive robot manipulation, which incorporates safety features such as object collision avoidance and implements the proposed real-time inverse kinematics solution.

- *Aim 6:* Implement a comprehensive evaluation of the optimal assistive robot and the SOEM-based library, verifying extensive workspace coverage, successful inverse kinematics implementation, compact foldability, and 3 kg load handling capacity.

1.4 Structure of the dissertation

To fulfill the proposed aims this research is organized as follows:

Chapter 2: Assistive Robots and ADLs: Exploring Design Constraints

This chapter conducts an in-depth review of Assistive Robots (ARs), exploring their progression, innate design complexities, and their essential function in aiding ADLs. It subsequently pivots towards the design requisites of assistive robots, focusing on tailoring the workspace to cater to crucial ADLs, underlining the significance of a foldable design in assistive robots, and providing a meticulous analysis of the assistive robots' weight handling capacity, with a set benchmark at 3 kg. These design specifications form the bedrock of this research, directly influencing Aim 2 and Aim 4. This guides the design optimization process of the assistive robot, driving the development towards more effective and adaptable solutions for individuals with ULED.

Chapter 3: FORMULATING A NEW INVERSE KINEMATICS ALGORITHM: THEORETICAL FOUNDATIONS AND APPROACH This chapter delves into the domain of inverse kinematics, particularly its application in assistive robotics. It initiates with a detailed exploration of established numerical techniques for inverse kinematics, encompassing Jacobian-based methods as well as Quasi-Newton methods. Subsequent to the dissection of these numerical strategies, the chapter unveils a novel algorithm named HUNTER (Helping Users Navigate through Extended Reach). This algorithm is specifically formulated to overcome the shortcomings prevalent in current methodologies. Consequently, this chapter corresponds directly with Aim 1 of this research.

Chapter 4: ENHANCING MANIPULABILITY AND ENERGY EFFICIENCY: DEFINITION OF A NEW METHODOLOGY FOR DESIGNING ASSISTIVE ROBOTS This chapter focuses on performance indices, crucial scalar functions derived from matrix models, which represent unique features of parallel or serial robots. Essential in the

design of assistive robots, these indices help enhance manipulability by quantifying kinematic performance. The chapter introduces a novel manipulability index to overcome past limitations. It also emphasizes the importance of minimizing energy consumption, detailing specific indices to calculate energy usage within a workspace. The chapter concludes with the introduction of a novel approach that synergizes kinematic performance and energy consumption indices and integrates a collision algorithm to detect link collisions. This exploration directly corresponds to Aim 2 of this research.

Chapter 5: CRAFTING PERFECTION: THE OPTIMAL DESIGN OF AN ASSISTIVE ROBOT This chapter offers a thorough step-by-step implementation of the comprehensive methodology for designing an optimal assistive robot. Employing a Scatter Search-based optimization and a meticulous validation process, it successfully produces a design that meets stringent efficiency and reliability criteria. The implications of this endeavor extend to achieving Aim 3 and Aim 4 of this research, underlining the link between sound methodology and the creation of an efficient, reliable assistive robot that meets the necessary specifications for supporting daily activities.

Chapter 6: DEVELOPMENT OF THE SOEM-BASED LIBRARY FOR ASSISTIVE ROBOT MANIPULATION this chapter delves into the development of a Simple Open EtherCAT Master (SOEM) based library that forms a seamless bridge between EtherCAT and ROS2. This library will empower us to control the proposed assistive robot in real-time, signifying a crucial step towards our assistive robotics goal, specifically pertaining to Aim 5.

Chapter 7: FROM THEORY TO PRACTICE: TESTING THE ASSISTIVE ROBOT

In accordance with the sixth aim of this research, this chapter is devoted to the evaluation of the assistive robot's performance and reliability in its physical form, utilizing the SOEM-BASED library developed during the course of this study.

CHAPTER 2

Assistive Robots and ADLs: Exploring Design Constraints

The study of assistive robotics has been ongoing since the 1960s, fueled by the need to aid individuals with upper and lower extremities dysfunctions (ULEDs) [43, 44]. The primary purpose of assistive robots is to restore the autonomy of individuals with ULEDs by replicating the functionalities of the human arm [45].

In the first section of this chapter, a comprehensive review of assistive robots mounted on wheelchairs is presented. The assistive robots are differentiated according to their number of Degrees of Freedom (DOF), excluding additional features of the gripper or the wheelchair, such as lifting, which are not counted as extra DOFs. The robots within each classification are carefully scrutinized to identify trends and design criteria of assistive robots. From this review, two key design objectives emerge: the enhancement of manipulability and the reduction in energy consumption. A significant challenge in assistive robots, the computation of inverse kinematics (IK), is also highlighted, which can be complicated by potential singularities within the workspace. These findings provide a comprehensive understanding of the current status of

AR.

The second section of the chapter presents design constraints for assistive robots based on the findings of the study and the design aims of the research. These constraints indicate that the robot should be capable of folding to minimize space usage, its workspace should correspond to 19 ADLs to accommodate a wide range of human activities, and it should be able to handle a weight of 3 kg within the workspace. The constraints outlined in this section provide a basis for the future design and development of AR, aligning with the overall research objectives.

2.1 Assistive Robotics: Unpacking Current Trends and Design Criteria

Assistive robots, characterized by different degrees of freedom (DOF) - namely 6-DOF, 7-DOF, and 8-DOF - bring distinct strengths and weaknesses in their design, functionality, energy consumption, and overall operation efficiency.

This section conducts a thorough review of ARs, primarily emphasizing their DOF, and outlines the corresponding design considerations and implications. The objective is to provide insights that can guide the design and application of ARs in various contexts, taking into account factors such as task complexity, environment, design resources, control complexity, energy efficiency, and operational costs.

2.1.1 6-DOF

6-DOF assistive robots bring distinct advantages due to their relative simplicity and a balanced approach between functionality and energy efficiency. These robots can adeptly perform a vari-

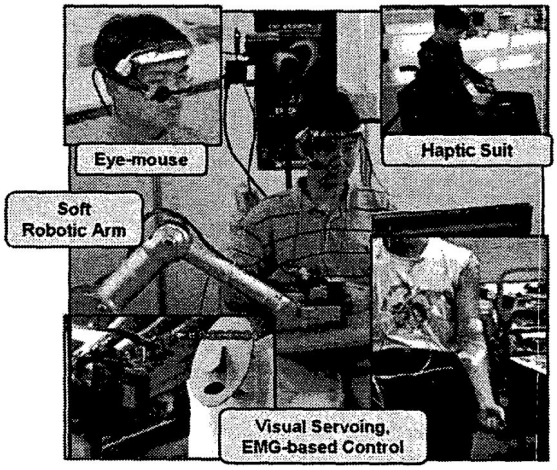


Figure 2.1: KARES II, a wheelchair robot system. Image reproduced from [1]

ety of tasks with a high degree of precision within a practical workspace. The manageability of their control systems potentially contributes to a more robust and cost-effective design. However, certain limitations may surface, particularly their restricted versatility in arm positioning and orientation, especially for more intricate tasks or environments. Obstacle avoidance could pose a challenge, and they may not be able to reach all points within their workspace with every possible orientation.

KARES II: a wheelchair robot system

The robot KARES II, which is depicted in Figure 2.1 and detailed in [1], incorporates several distinct features:

- Composed of a powered wheelchair and a mobile-platform-based robotic arm, KARES II broadens its workspace through the mobility of its base.
- The robot adopts the Denavit-Hartenberg parameters of the PUMA, thereby simplifying the computation of inverse kinematics.

- A pivotal difference between KARES and KARES II is evidenced in their control methods. KARES II integrates myoelectric sensors and eye-gaze sensors, in contrast to KARES, which uses a joystick for control as discussed in [46].
- KARES II is capable of handling a maximum payload of 500 grams.

Despite these features, the moving platform introduces certain constraints. Although it expands the workspace, it simultaneously enlarges the overall size of the robot. This increase could potentially limit the number of ADLs that the robot can perform, and could also impact its autonomy negatively.

Weston

The Weston wheelchair-mounted assistive robot, displayed in Figure 2.2 and described in [2], is constructed based on a set of essential requirements:

- It is crucial that the manipulator doesn't interfere with the control of the wheelchair by obstructing the user's vision or destabilizing the wheelchair.
- The manipulator's control should be seamlessly integrated with the control system of the wheelchair.
- A joystick should be the primary means of controlling the robot arm.

Upon testing, researchers identified several design challenges. The width of the robot emerged as a critical constraint, as it restricted both visibility and control. Additionally, issues related to the robot's stability were reported, particularly when the robot was fully extended. Consequently, the authors suggest that reducing the robot's inertia could address these stability concerns.

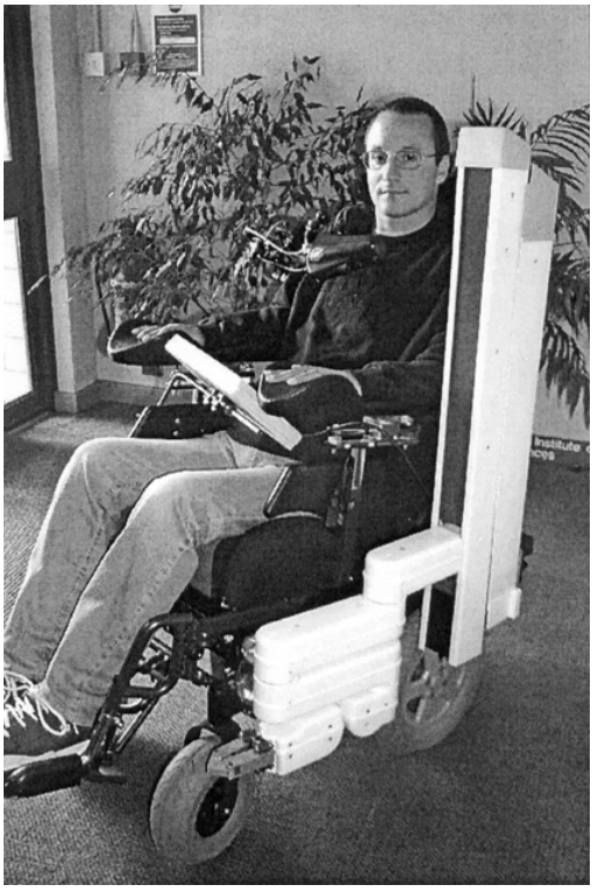


Figure 2.2: Weston robot attached to Strom wheelchair. Image reproduced from [2]

MANUS: a wheelchair-mounted rehabilitation robot

The MANUS assistive robot, as illustrated in Figure 2.3 and discussed in [3], was designed with a distinct set of requirements in mind:

- The robot, when attached to the wheelchair, should maintain a slim profile to easily pass through doorways.
- The control of MANUS should be facilitated through a joystick.

Figure 2.3: MANUS: a wheelchair mounted rehabilitation robot. Image reproduced from [3]

- The robot should be directly connected to the wheelchair.
- The robot's control system should be seamlessly integrated with the wheelchair's control.
- The robot should strive for minimal power consumption.
- The robot's arm should be capable of lifting up to 1.5 kg of weight.

In order to minimize the impact in the event of a collision, the designers made a conscious effort to reduce the inertia of each link in the robot. They achieved this by positioning all the motors at the base of the robot and utilizing belts and gears to transmit power to each joint.

This approach, however, had the side effect of increasing control difficulties due to the backlash at each joint. The extensive use of gears also increased friction, leading to an increased power consumption.

A subsequent iteration of MANUS, known as Iarm, included CAN communication, which facilitated the robot's communication with a PC [47].

JACO

Figure 2.4: JACO mounted on a power wheelchair. Image reproduced from [4]

JACO, a commercially available assistive robots that can be mounted on a wheelchair, is depicted in Figure 2.4. Its key features are:

- The robot is equipped with a joystick that is integrated directly into the wheelchair.

- With a slim design, JACO reduces the inertia of each link. As a result, even in case of a collision, the robot does not exert any harmful force on the user.
- A torque limit is implemented in JACO's controller to prevent the user from inflicting self-harm.
- JACO incorporates a spasm filter to mitigate the effects of discontinuities in its kinematics.

Despite these advantages, JACO's primary limitation lies in dealing with singularities. In proximity to singularities, JACO may produce unforeseen movements, causing potential operational challenges.

2.1.2 7-DOF

7-DOF assistive robots offer an additional degree of freedom, providing superior flexibility. These robots are adept at performing more complex manipulative tasks and navigating around obstacles more efficiently. However, the energy consumption of these robots escalates due to the additional mechanical joint. The complexity in the mechanical design and the control algorithm development increases, potentially leading to a rise in costs and maintenance needs. Stability and robustness might also be compromised compared to 6-DOF systems.

FRIEND II

FRIEND II, the successor to the rehabilitation robot FRIEND, is acknowledged as an intelligent wheelchair-mounted manipulator by its creators. The robot, as depicted in Figure 2.5, is elaborated upon in [5].

Figure 2.5: Rehabilitation robot FRIEND II. Image reproduced from [5]

- FRIEND II represents a more advanced version of FRIEND I. The robot includes an expanded set of sensors to facilitate interaction between the objects and the robot, thereby enhancing the abstraction level of the controller. In this context, the robot can predict the objects with which the user is likely to interact [48].
- The robot's 7-DOF configuration mirrors the joint assignment of the human arm, thereby aiming to achieve comparable manipulability.

There are two additional versions of the robot, FRIEND III and FRIEND IV, but the enhancements in these versions are predominantly focused on controller improvements, including collision detection and the addition of a chin-joystick [18]. Therefore, these subsequent versions do not present substantial design advancements compared to FRIEND II.

WMRA I

Figure 2.6: WMRA in a feeding pose, low-right reach, left side doorknob. Image reproduced from [6]

WMRA, a wheelchair-mounted robot, was developed by the University of Florida. The robot is displayed in Figure 2.6 as shown in [6]. The following design criteria were utilized in the construction of this robot:

- The robot's weight is constrained to a maximum of 14 kg to enhance the robot's autonomy.
- The robot is mounted as far forward and upward on the wheelchair as possible. This positioning increases manipulability while minimizing obtrusiveness.
- The payload capacity of the robot is capped at 2 kg.
- The base of the robot is designed to be adaptable, allowing the robot to be installed on different wheelchair models.
- The robot is controlled via a joystick, simplifying user control.
- The robot requires 7-DOF to have full pose control, especially in complex areas of the workspace such as reaching around the wheelchair.
- The control of the robot and wheelchair should be integrated [49].

A successor to WMRA I, known as WMRA II, was introduced in [50]. The key difference in WMRA II is its construction material; instead of using aluminum, it employs carbon fiber, reducing the robot's weight to 11.5 kg.

2.1.3 8-DOF

8-DOF assistive robots provide the greatest degree of flexibility and versatility among the three types of ARs. These robots can undertake a wider array of tasks, especially in complex surroundings. Their design enables them to perform intricate manipulations and bypass obstacles more effectively. However, these advantages also come with inherent challenges. The increase in DOFs enhances the complexity of the system, leading to higher energy consumption, making

the robot's design, control, and maintenance more demanding and costly. The elevated flexibility may require advanced safety mechanisms to prevent undesired or unsafe movements. In this review, the only 8-DOF assistive robot found in the state-of-the-art is EDAN.

EDAN

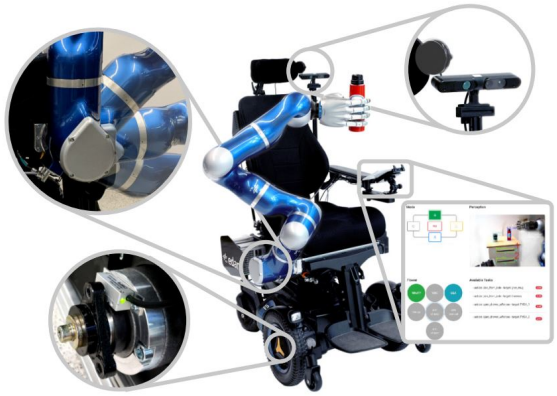


Figure 2.7: EDAN system including a closeup of the wheel-encoders, the 8-DOF robot, the head-switch, the RGB-D camera, and the tablet interface. Image reproduced as presented in [7].

EDAN is a fully integrated wheelchair-based manipulation robot. Figure 2.7 presents EDAN as introduced in [7]. The design consideration of the robotic arm are summarized as follows:

- EDAN can be controlled by a joystick or via electromyographic signals.
- EDAN uses a lightweight robot named LWR-III.
- The robot has 8-DOF to provide additional reachability in the complete surroundings of the wheelchair.
- EDAN has a payload capacity of 1 kg

- EDAN has a total weight of 30 kg.

EDAN is also the first wheelchair-mounted assistive robots that implemented EtherCAT communication. The implementation of EtherCAT allows the robot to handle an increased number of signals in real-time.

Table 2.1: Timeline of the assistive robots classified according to the DOF.

		Year								
		1999	2000	2002	2003	2005	2006	2009	2012	2020
DOF	6-DOF	KARES I	MANUS	Weston	KARES II			JACO	IARM	
	7-DOF				FRIEND I	FRIEND II	WRMA I	FRIEND III WRMA II	FRIEN IV	
	8-DOF									EDAN

2.1.4 Identifying Challenges in Current Assistive Robot Design and Functionality

In this section, the core findings from the review of assistive robots were examined. As displayed in Table 2.1, it became evident that over time, assistive robots designs trended towards incorporating more DOFs. This trend reflected a growing requirement for robots with greater range of movement and improved safety measures. Increased DOFs provided redundancy in IK, resulting in multiple configurations for identical position and orientation. Consequently, the robots could more easily avoid collisions with both the user and themselves.

A second conclusion drawn from the review highlighted the necessity for lighter ARs, driven by the need to extend assistive robots autonomy. As ARs are typically mounted on battery-operated wheelchairs, a lighter robot design would naturally lead to reduced energy consumption. Lighter robots also offered increased payload capacity, since the dynamic torque produced

by the robot's weight on the motors was lessened. However, a trade-off was identified: lighter robots exhibited decreased stiffness, which could lead to increased vibrations and decreased accuracy. Yet, in the context of assistive applications, precision was not considered a primary concern.

A key challenge noted in the quest for lighter robots was their contradiction with the trend towards increased DOFs. More DOFs implied more actuators, which in turn added to the robot's total mass. This conflicting scenario underscored the need for optimization strategies in assistive robotsdevelopment, balancing high manipulability with low energy consumption.

2.2 Establishing Design Constraints for Assistive Robots Based on ADL Requirements

This section delves into the crucial aspects of designing an efficient assistive robot to cater to the needs of the physically challenged. The section is structured into three vital subsections, each dedicated to a primary design constraint that emerged from the study findings. These constraints represent not only key areas of focus for this research, but they are also integral to the overall performance and efficiency of the assistive robot.

The first subsection focuses on the ADLs and their significant influence on the design and functionality of assistive robots. To address a wide variety of daily tasks, the robots' workspace was tailored to accommodate 19 essential ADLs identified through comprehensive customer discovery. Using the Lawton IADL scale and Bristol ADL scale, the positions of objects associated with ADLs were examined, leading to the categorization of these activities into seven groups. The workspace dimensions for each ADL category, including the orientation of the end effector,

were carefully considered, ensuring a versatile and user-friendly solution for individuals with upper extremity dysfunctions.

The second subsection sheds light on the concept of foldability, an attribute that holds significant importance in minimizing the space utilization of the robot. This discussion is central to creating an assistive robot that is not only functional but also practical in daily environments.

Lastly, the third subsection revolves around the robot's ability to handle weights, with the benchmark set at 3 kg within the workspace. Understanding and accommodating this aspect is crucial for the robot to efficiently assist users in their daily tasks.

These subsections provide an in-depth exploration of the three central constraints that guide the design and development of assistive robots in line with the research objectives.

2.2.1 Using ADLs to Define Workspace in Assistive Robot Design

In the process of customer discovery [51], more than 220 interviews were conducted with a diverse group. This group included elderly individuals residing in assisted living centers, people with upper and/or lower extremities dysfunctions, stroke survivors, and educators/researchers in the field of rehab/assistive robotics. It also involved occupational therapists (OTs), recreational therapists (RTs), physiotherapists (PTs), clinicians, caregivers, family members, and representatives from wheelchair and assistive device manufacturing companies, distributors, assistive technology professionals (ATPs), and insurance providers. These interviews aimed to explore the needs, market opportunities, and trends concerning wheelchair-bound individuals with upper extremity dysfunction, as well as propose a minimum viable solution to address their needs.

Table 2.2: Categorization of the listed essential ADLs.

Categories	**Essential ADLs**
CATEGORY 1	Picking/ Placing objects from table Holding spoon/fork Maneuvering spoon/fork to take food from bowl to plate Holding credit cards, Swipe credit cards at the shop, ATM booth Opening/ closing lid of a jar, box, paper box, cap of bottles Holding printed books, turning pages
CATEGORY 2	Picking/ Placing objects from ground
CATEGORY 3	Picking/ Placing objects from man reach, upper shelf
CATEGORY 4	Opening closing drawers
CATEGORY 5	Push-pull, Swing open-close doors Opening/ closing Refrigerator, Oven
CATEGORY 6	Holding cup near the mouth Gradual upward positioning of the cup while drinking Maneuvering spoon/fork to take food from bowl to plate Maneuvering spoon/fork to put food to the mouth Putting pills/medicine in the mouth Holding the phone near the ear or put in speaker mode
CATEGORY 7	Holding Pen, Maneuvering on paper or surfaces

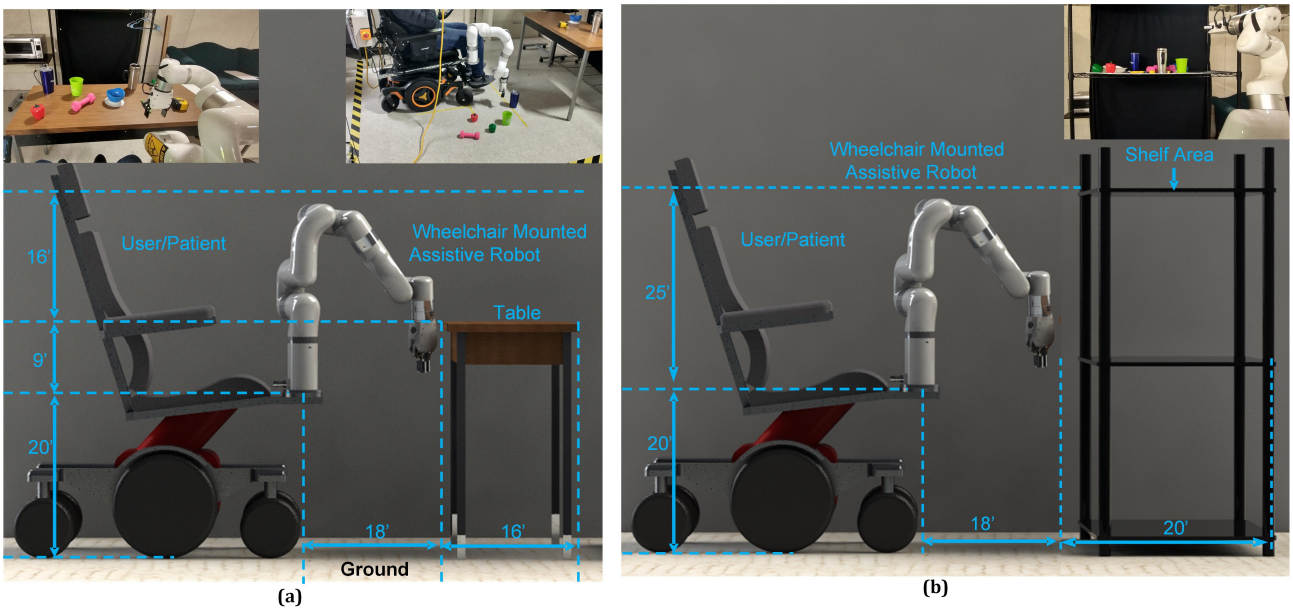


Figure 2.8: Experimental setup to measure workspace of the ADLs.

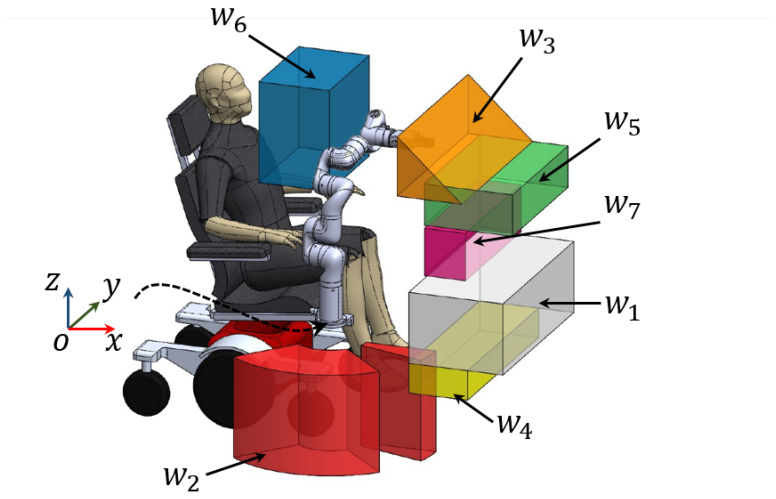


Figure 2.9: Representation of the workspaces of each ADLs relative to the robot and the user.

From these discussions, 19 essential ADL were identified for investigation in this study. ADLs serve as assessment tools to determine an individual's functional status and the level of

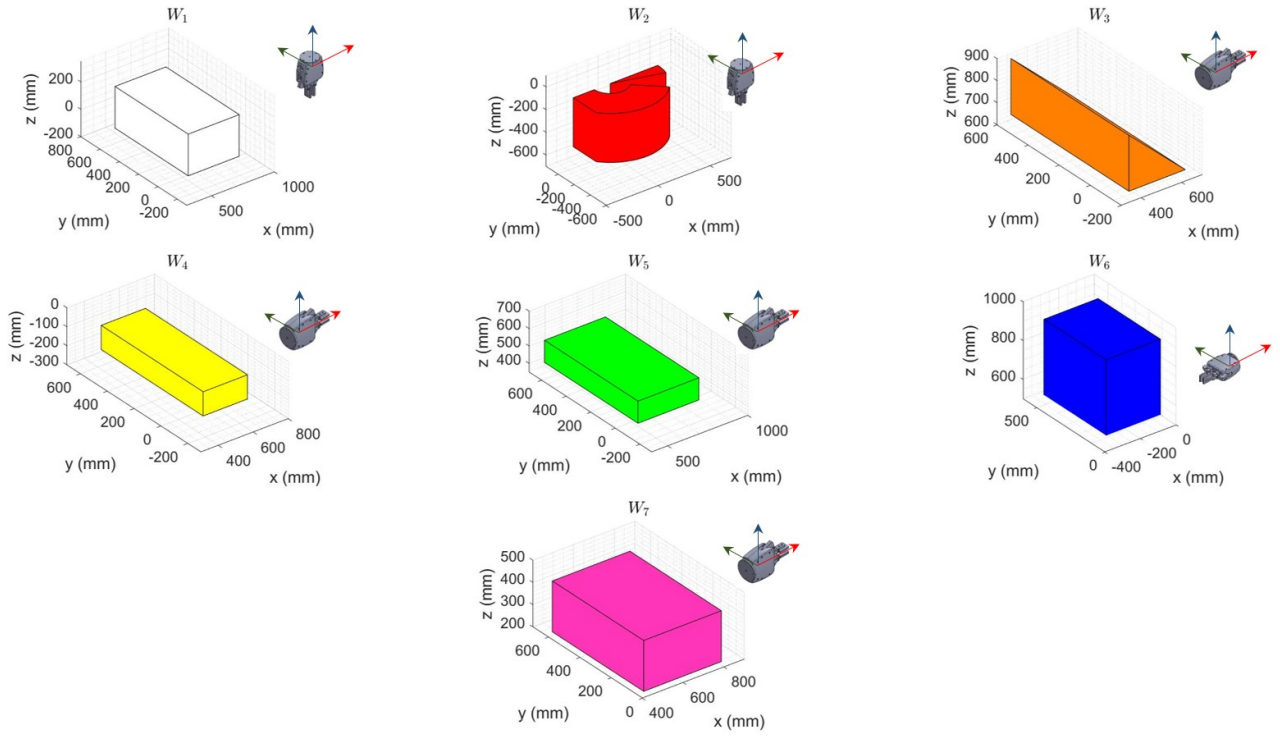


Figure 2.10: Dimensions and location of each ADL workspace, including the orientation of the end-effector relative to O.

care services required for maintaining a good quality of life. The Lawton IADL scale [52] and Bristol ADL scale [53] were used throughout the customer discovery process. After examining the positions of the objects associated with the ADLs, these 19 ADLs were grouped into seven different categories, as shown in Table 2.2, with each category listing its corresponding ADLs.

Figure 2.8 illustrates a few of the ADLs experimental setup used for measuring the workspace. All ADLs accounted for a clearance relative to the robot, ensuring collision-free movement of the end-effector. Although the user's blind spots were excluded from the workspace, it was ensured that the end-effector could still cover these areas. The measured workspace relative

to the wheelchair is represented in Figure 2.9, while Fig. 2.10 depicts the dimensions of each ADL workspace category, including the end-effector's orientation. It should be noted that the end-effector orientation emulates the orientation of a human hand approaching specific objects for manipulation [54].

2.2.2 Foldability: An Essential Design Constraint for Wheelchair-Mounted Assistive Robots

This subsection delves into the critical importance of foldability as a design consideration for wheelchair-mounted AR. Foldability, which allows for the reduction in size and form of the robot when not in use, is a feature that combines practicality with functionality, enhancing the user's experience in everyday environments.

- **Space Efficiency:** Efficient utilization of space is crucial for wheelchair users, given the limited spatial capacity inherent to most indoor environments. A foldable assistive robot takes up minimal space when inactive, thus reducing potential collisions or obstructions, and enhancing the user's mobility [55].

- **Portability:** Foldability also contributes to the portability of the assistive robot. It facilitates the transportation of the wheelchair through narrow passages and during vehicle transportation [56].

- **User Convenience:** A robot that can fold away when not in use minimizes interference with the user's daily activities. This characteristic promotes ease of use, which is vital for user acceptance of assistive technology [57].

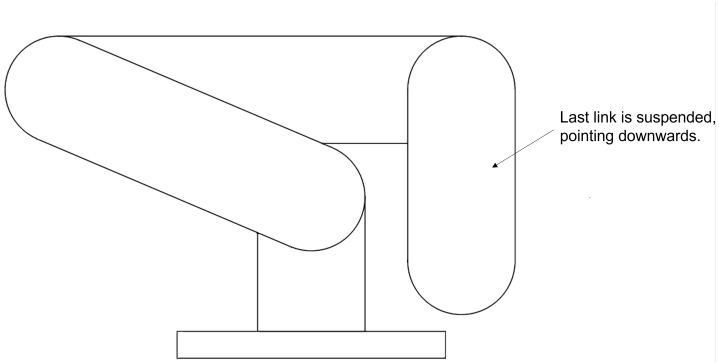


Figure 2.11: Schematic Configuration Illustrating the Foldable State of the Assistive Robot Design.

- **Safety:** When the assistive robot is folded, the risk of unintended contact or injury decreases. This safety measure is particularly important for users with limited mobility or impaired visibility [58].
- **Aesthetics:** Foldability can also offer aesthetic benefits. A folded robot can integrate more seamlessly into various living environments, presenting a less obtrusive and more appealing appearance [59].
- **Energy Conservation:** The ability of the assistive robot to fold may also contribute to energy conservation. Certain parts of the robot might not require power when in a folded state, thereby extending the battery life of the robot and the wheelchair [60].

In essence, the concept of foldability in ARs mounted on wheelchairs brings substantial advantages. It not only amplifies user experience and safety but also boosts the overall efficiency of the assistive device. Figure 2.11 showcases the folding configuration that will guide the design of the assistive robots in this study. In this layout, the final links are depicted as suspended,

leading to a foldable design that aligns well with the requirements of ARs.

2.2.3 Designing for a 3 kg Load: A Key Consideration in ARs

The capacity of assistive robots to manage a load of up to 3 kg within their operational scope is a crucial design element. This requirement is grounded in the fact that 3 kg is the approximate average weight of items people frequently interact with in their daily routines [61]. Therefore, this weight handling ability is not only a practical but a vital specification for an assistive robot, enabling it to provide beneficial assistance across a broad spectrum of everyday tasks.

2.3 Conclusions

In conclusion, this chapter has offered a thorough examination of ARs, presenting insights into their evolution, design challenges, and the key aspects of their functionality in supporting ADLs.

The chapter begins by analyzing the growing trend of increasing DOFs in assistive robots design, noting the enhanced maneuverability and safety this provides, in addition to the redundancy offered in Inverse Kinematics. Despite these advantages, it is acknowledged that more DOFs present challenges such as increased robot mass and energy consumption. The discussion then moves to the necessity of lightweight assistive robots designs, emphasizing their role in extending autonomy and enhancing payload capacity. However, the pursuit of lighter robots often finds itself at odds with the aim for more DOFs, thus underlining the need for strategic optimization in assistive robots development.

Then, the chapter delved into the design requirements of ARs, stressing the necessity of

customizing the workspace to facilitate essential ADLs, the significance of foldability in assistive robots design, and an analysis of the AR's weight handling capacity, with a benchmark set at 3 kg. Collectively, these considerations emphasize the intricate balance required in assistive robots design, entailing improved maneuverability, efficient energy consumption, accommodation of ADLs, optimal space usage, and appropriate weight handling capacity. These design constraints will act as the cornerstone for the upcoming research, guiding the design optimization of the assistive robot in pursuit of more effective and versatile solutions for individuals with ULED.

CHAPTER 3

FORMULATING A NEW INVERSE KINEMATICS ALGORITHM: THEORETICAL FOUNDATIONS AND APPROACH

Inverse kinematics (IK) is the problem of computing the joint angles for a kinematic chain, skeleton, or mechanism such that an end effector will reach a prescribed goal. IK is a highly used tool for the generation of trajectories of industrial robots. In the case of assistive robots, the inverse kinematics is mainly for the joystick control of the end-effector/gripper, which acts as the handling tool to perform ADL [62]. Thus, the IK of assistive robots comes with additional requirements and limitations compared to industrial robots. For example, in industrial robots, the design of trajectory is done offline, i.e., which facilitates the detection and avoidance of singularities within the workspace. Contrarily, assistance robots are controlled online due to is difficult to generate a specific trajectory for each ADL. This condition creates issues with online control because an untrained user can not identify and avoid singularities within the

robot workspace [63].

Singularities are positions where the robot loses one or more degrees of freedom (DOF). Singularity avoidance is paramount because the conventional IK methods cannot solve IK for singular positions. Consequently, the user of assistive robots lost control of the robot at singular positions. Thus, an algorithm that can solve the IK problem in a reliable way for assistive robots will increase the manipulability of such robots. This means the robot would be able to perform a wider range of movements, improving its functionality and versatility.

There are two kinds of IK solution methods, analytical and numerical. Analytical methods are either based on geometric or algebraic approaches [39]. Although analytical methods yield the fastest solution for the IK, there is no general methodology for finding the IK for any robot. The closest attend of a general IK solution is presented by Rosen [40], who presented the algorithm IK-fast. Although this algorithm computes the IK of most non-redundant robots, if the robot is to complex, redundant, or has too many constraints the algorithm fails to find the IK. Thus, researchers usually prefer to simplify the robot design to facilitate the calculation of the IK, at expenses of increasing the singularities within the workspace [41]. Opposite to analytical methods, Numerical methods apply to any robot and allows the implementation of different kinds of constraints. However, they come with other issues, such as convergence speed and accuracy of the solution.

In the upcoming chapter, a comprehensive understanding of the approach to enhance the capability and efficiency of assistive robots through a novel inverse kinematics method is laid out. This chapter is systematically structured into three main sections, serving to clarify the progression from conceptualization to realization. The first section delves into a thorough examination of existing state-of-the-art numerical IK methods, spotlighting their inherent strengths

and pinpointing their limitations. This section also includes a critical review of the methodologies and accomplishments of current IK methods, providing a solid foundation for subsequent discussion. In the second section, the proposed method for computing inverse kinematics, explicitly designed to augment the performance of assistive robots, is unveiled. This segment walks the reader through the conceptual framework, design, and practical implementation of the novel algorithm.

3.1 State-of-the-art numerical IK

In this section, the current state-of-the-art numerical inverse kinematics (IK) methods are examined. IK is the mathematical process of calculating the variable joint parameters needed to place the end-effector (the device at the end of a robotic arm) of a robot in a certain position and orientation.

To study these numerical methods, the position and orientation of a robot's end-effector can be represented by a function P of the joint angles θ:

$$P(\theta) = \begin{bmatrix} P_x(\theta) \\ P_y(\theta) \\ P_z(\theta) \\ \theta_x(\theta) \\ \theta_y(\theta) \\ \theta_z(\theta) \end{bmatrix} \tag{3.1}$$

In this equation, P_x, P_y, and P_z are the positions of the end-effector along the x, y, and z axes, respectively, and θ_x, θ_y, and θ_z are the orientations of the end-effector about these axes.

θ is a vector representing the joint angles.

Assuming there is a goal position and orientation g for the end-effector, the IK problem can be represented by the following residual vector:

$$R(\theta) = P(\theta) - g \tag{3.2}$$

Here, the residual vector R is the difference between the current and goal positions and orientations of the end-effector. The numerical solution to the IK problem is found when $R = 0$, that is, when the end-effector is at the goal position and orientation.

In the realm of robotics, numerical methods for achieving this goal are primarily classified into two groups: Jacobian methods and quasi-Newton methods. These techniques will be scrutinized in the subsequent subsections. The ultimate aim of these methods is to devise efficient ways of making $R = 0$, thereby accurately positioning the end-effector as per the desired goal.

3.1.1 Jacobian methods

The Newton-Raphson method is a widely recognized numerical method used in solving inverse kinematics problems [64]. This method utilizes an iterative algorithm that updates θ using the following equation:

$$\theta^k = \theta^{k-1} - J\left(\theta^{k-1}\right)^{-1} R\left(\theta^{k-1}\right) \tag{3.3}$$

Here, J represents the Jacobian matrix, which is a mathematical construct that describes the change in the end-effector's position and orientation concerning changes in the joint angles.

The k and $k-1$ denote the current and previous iterations of the algorithm, respectively.

However, the Newton-Raphson method comes with its share of challenges. One major issue is its computational expense, primarily attributed to the requirement of computing the inverse of the Jacobian matrix. Moreover, when the robot is in a singularity, the Jacobian matrix also becomes singular (meaning it is not invertible), which prohibits the computation of the inverse kinematics.

Furthermore, Newton-Raphson does not control the step size, the increment of change in joint angles in each iteration. If the initial guess of joint angles is too far from the actual solution, the method might diverge to unfeasible solutions. Lastly, this method is limited to 'square' problems, where the number of joint angles equals the size of $R(\theta)$.

To address these issues, a collection of methods, known collectively as Jacobian methods, have been developed. These methods essentially involve modifications to the Jacobian matrix and its utilization to yield more efficient and robust solutions to the inverse kinematics problem.

Jacobian transpose

A method that addresses the singularity issues of the Newton-Raphson approach is the Jacobian transpose method. This technique circumvents the need for the inverse of the Jacobian matrix by utilizing its transpose. Hence, the previous equation 3.3 is adapted as follows:

$$\theta^k = \theta^{k-1} - \alpha J\left(\theta^{k-1}\right)^T R\left(\theta^{k-1}\right) \tag{3.4}$$

In this equation, α denotes a step size parameter, which regulates the increment of change in joint angles during each iteration. This parameter is determined using:

$$\alpha = \frac{< R, JJ^T R >}{< JJ^T R, JJ^T R >} \tag{3.5}$$

Here, the angle brackets $<,>$ indicate the inner product.

Though the Jacobian transpose method overcomes the singularity challenge present in the Newton-Raphson approach, it has its set of drawbacks. For instance, it generally requires a significant number of iterations to converge on a solution. The number of iterations can be decreased by enhancing the value of α, but this solution isn't without its problems. Studies have reported instances of oscillations and discontinuities in the resulting solution when using a higher α value [65].

Jacobian pseudo-inverse

Another technique that has been employed to solve the inverse kinematics problem is the use of the Jacobian pseudo-inverse. The pseudo-inverse, often referred to as the Moore-Penrose inverse, is calculated as follows:

$$J^{\dagger} = J^T (JJ^T)^{-1} \tag{3.6}$$

Using this calculation, the iterative equation for updating joint angles 3.3 becomes:

$$\theta^k = \theta^{k-1} - J^{\dagger} \left(\theta^{k-1}\right)^T R \left(\theta^{k-1}\right) \tag{3.7}$$

The pseudo-inverse calculation is applicable for matrices of any size, meaning that the Jacobian pseudo-inverse method isn't limited to square problems, expanding its usefulness in a broader range of scenarios.

However, while the pseudo-inverse is computationally feasible even for singular matrices, there are noted issues when approaching singularities. In these situations, large changes in joint angles have been reported, leading to high oscillations in the end-effector's position and potentially causing instability [66].

Damped least squares

The Damped Least Squares method employs the Levenberg-Marquardt algorithm, which can mitigate the oscillations caused by the Jacobian pseudo-inverse when approaching singularities. This technique adjusts the calculation of the inverse Jacobian with a damping factor, as shown in the following equation:

$$J^{\star} = J^{T}(JJ^{T} + \lambda^{2}I)^{-1} \tag{3.8}$$

In this equation, λ is a non-zero damping constant, and I represents the identity matrix. The careful selection of the damping constant, λ, can enhance the solution behavior of the inverse kinematics (IK) near singularities, providing a more controlled response.

However, the introduction of this damping constant introduces a trade-off. While it can improve the stability of the robot's movements near singularities, it can decrease the accuracy of the IK solutions at non-singular positions, leading to less precision in the robot's movements. This is a critical consideration in the application of this method [67].

Singular value decomposition

The Singular Value Decomposition (SVD) method is used to address the high computational cost associated with the pseudo-inverse calculation, especially in the presence of singularities

[68]. This method simplifies the calculation of the Jacobian pseudo-inverse, decomposing the matrix into its eigenvalues and two orthogonal matrices. The pseudo-inverse is then computed as follows:

$$J^{\dagger} = \sum_{i=1}^{r} \sigma_i^{-i} v_i u_i^T \tag{3.9}$$

In this equation, r denotes the number of rows in the Jacobian matrix. The σ_i values represent the singular values or eigenvalues of the matrix, while u_i and v_i are columns from the two orthonormal matrices U and V, derived from the original Jacobian matrix through the decomposition process.

While the Singular Value Decomposition (SVD) method is a significant advancement, it is not without its limitations. One of the major concerns is its computational complexity. SVD is computationally intensive, especially for large matrices, which might be a limiting factor in real-time applications where quick computation is crucial [69].

Furthermore, SVD can encounter difficulties when the Jacobian matrix contains near-zero singular values. The computation of the pseudo-inverse using SVD in these situations can lead to large magnitudes for the joint velocities due to the inversion of these small singular values. This can result in unpredictable and erratic robot motions [70].

Another challenge with SVD is the potential for numerical instability when the singular values of the matrix are not distinct, a situation known as an ill-conditioned matrix. This can result in inaccurate results and compromise the reliability of the inverse kinematics solutions [71].

Finally, the implementation of SVD is more complex compared to some other methods, which could add to the difficulty in achieving correct and efficient implementation [69].

Pseudo-inverse damped least square

This method extends the damped least square method using the SVD, obtaining the following expression:

$$J^T(JJ^T + \lambda^2 I)^{-1} = \sum_{i=1}^{r} \frac{\sigma_i}{\sigma_i^2 + \lambda^2} v_i u_i^T \tag{3.10}$$

The use of SVD facilitates the identification of singularites, this way if $\sigma_i \approx 0$ the damping constant is used to force the existance of the inverse. Consequently, the damping constant is only employed near singularities, thereby avoiding the jerking motion of the end-effector often observed in these regions [72]. However, it's important to note that the implementation of SVD introduces an additional layer of computational complexity. While SVD can significantly assist with the inverse computation and singularity detection, it concurrently increases the computational cost of the method, particularly in scenarios involving robots with a large number of degrees of freedom. This increase in computational demand is a crucial consideration when evaluating the feasibility of this method in real-time applications.

3.1.2 Quasi-Newton methods

Quasi-Newton methods offer an alternative approach to the Inverse Kinematics problem by treating the solution of equation (3.2) as an optimization problem:

$$L(\theta) = \mathbf{min}\ R(\theta)^T R(\theta) \tag{3.11}$$

This implies that the objective is to minimize the magnitude of the residual vector, $R(\theta)$, by adjusting the joint angles (θ).

The optimization problem can be solved by considering the second-order Taylor series ap-

proximation of equation (3.11):

$$L(\theta + \Delta\theta) = L(\theta) + \Delta L^T(\theta)\Delta\theta + \frac{1}{2}\Delta\theta\nabla^2 L(\theta + \Delta\theta)\Delta\theta \tag{3.12}$$

Upon differentiation with respect to $\Delta\theta$, the following result is obtained:

$$\nabla L(\theta + \Delta\theta) = \nabla L(\theta) + \nabla^2 L(\theta)\Delta\theta \tag{3.13}$$

Setting $\nabla L(\theta + \delta\theta) = 0$ (the stationary point), yields:

$$\Delta\theta = -(\nabla L^2(\theta))^{-1}\nabla F(\theta) \tag{3.14}$$

The solution to equation (3.14) is known as the Newton's method, where $\nabla^2 L$ is the Hessian matrix. Despite its similarity to the Newton-Raphson approach, Newton's method doesn't restrict the problem to be square [73]. However, it introduces an additional constraint: if the eigenvalues of the Hessian matrix are negative, the method will not converge.

To overcome this issue, a class of methods called quasi-Newton methods were introduced. These methods ensure that the Hessian matrix is always positive definite, thus providing a more robust solution. The two primary quasi-Newton methods employed in robotics are the Levenberg-Marquardt (LM) and the Broyden-Fletcher-Golfarb-Shanno (BFGS) methods.

Levenbger-Marquardt

The Levenberg-Marquardt (LM) method offers a different approach to dealing with negative eigenvalues in the Hessian matrix. To understand how the LM method works, let's first introduce the concept of eigen decomposition. In essence, the Hessian matrix can be represented

as:

$$\nabla^2 L = V D V^T \tag{3.15}$$

Here, D is a diagonal matrix whose elements are the eigenvalues of $\nabla^2 L$, and V is a matrix composed of the corresponding eigenvectors. The LM method utilizes this representation to manage the issue of negative eigenvalues.

Specifically, the LM method checks if the minimum eigenvalue is negative ($\min \lambda_i < 0$). If that's the case, the method modifies the Hessian matrix using the following equation:

$$M = V(D + (\beta - \min \lambda_i)I)V^T \tag{3.16}$$

In this equation, M is the modified Hessian matrix and β is a constant selected in such a way that the eigenvalues of $\nabla^2 L$ become positive. With this modification, the update equation (3.14) becomes:

$$\Delta\theta = -M^{-1}\nabla F(\theta) \tag{3.17}$$

This approach effectively avoids the issue of non-convergence due to negative eigenvalues.

Despite its advantages in addressing negative eigenvalues in the Hessian matrix, the LM method presents certain challenges. Foremost, the method demands the computation of the eigenvalues of the Hessian matrix during each iteration, which significantly raises computational costs [74]. This computational complexity may hinder real-time or rapid applications, particularly when dealing with high-dimensional systems [75]. Further, the method necessitates a careful selection of the constant β that ensures the positivity of the modified eigenvalues. The

performance and convergence rate of the LM method may greatly depend on this choice, adding another layer of complexity in its implementation [76].

Broyden-Fletcher-Golfarb-Shanno (BFGS) Method

The BFGS method offers a unique approach in dealing with the inverse kinematics problem. Rather than directly computing the inverse Hessian matrix, the BFGS method applies a series of n-rank updates to the Hessian approximation. This n-rank update is inherently positive definite, hence if the Hessian is positive definite at the outset, it remains so throughout the iterations.

The BFGS update is given by:

$$M^{k+1} = M^k + \frac{yy^T}{s^T y} - \frac{M^k s s^T M^k}{s^T M^k s} \tag{3.18}$$

Here, $s = x^{k+1} - x^k$ and $y = \nabla f\,(k+1) - \nabla f\,(k)$.

What makes the BFGS method stand out is its superlinear convergence. The method provides a solution even at singular positions, typically the problematic points for most inverse kinematics methods.

In order to control the step size, the BFGS method typically incorporates a line search step. Here, the chosen step size must satisfy the Wolfe conditions, which are given by:

$$\begin{aligned} f\left(\theta^k + \alpha^k \Delta\theta\right) &\leq f\left(x^k\right) + c_1 \alpha^k \nabla f^T\left(\theta^k\right) \Delta\theta \\ \nabla f\left(\theta^k + \alpha^k \Delta\theta\right) &\geq c_2 \nabla f^T\left(\theta^k\right) \Delta\theta \end{aligned} \tag{3.19}$$

The Wolfe conditions ($0 < c_1 < c_2 < 1$) are key to ensuring the step size progressively

reduces the objective function at each iteration.

By effectively solving an optimization problem, the BFGS method delivers solutions for inverse kinematics problems even when the initial guess is far from the actual solution. Despite its complexity, the BFGS method is noted for its robustness, making it applicable to any serial robot. Furthermore, this optimization-based approach enables the easy implementation of constraints, such as joint limits, making it particularly suitable for designing and controlling anthropomorphic robots [77].

3.2 HUNTER (Helping Users Navigate through Extended Reach): A Novel Algorithm for Inverse Kinematics in Assistive Robots

The design of assistive robots for wheelchair users is an intricate process that centers on facilitating the most positive and efficient user experience possible. Among the various factors that can enhance this experience is the ability of the robot to compute inverse kinematics, even for positions outside of its typical workspace [78].

Let's consider a scenario where the user instructs the robot to retrieve an item that is slightly beyond its standard reach. In most conventional robotic systems, this would likely result in task failure due to the physical limitations of the robot's range[79]. This failure could potentially lead to user confusion about why the robot could not complete the task [80].

However, if the robot is capable of performing inverse kinematics for positions beyond its regular workspace, it allows the user to gain a more accurate understanding of the system's

limitations[81]. For instance, if the user requests an object that is out of reach, the robot could compute the inverse kinematics for the closest possible position to the desired object. Consequently, the user might intuitively respond by slightly adjusting the wheelchair to bring the object within the robot's workspace.

Such an interactive feedback mechanism significantly enhances the user experience. Instead of merely observing the robot failing to complete a task, the user becomes part of the problem-solving process. The feeling of shared control and partnership adds a layer of autonomy and agency that significantly enriches the overall experience of using an assistive robot [82].

The BFGS method, however, struggles with handling inverse kinematics for positions outside the workspace. The issue lies in that its n-rank approximation might fail when the position is outside the workspace, yielding a Hessian matrix that may be negative. This results in a limitation to the BFGS method's capabilities.

That's where our novel algorithm, HUNTER (Helping Users Navigate through Extended Reach), comes in. HUNTER augments the BFGS method with an additional step: a check for any negative elements in the Hessian matrix diagonal. If such elements exist, HUNTER incorporates the Levenberg-Marquardt method, ensuring a positive definite Hessian matrix. This adjustment significantly enhances the BFGS method, allowing for effective computation of inverse kinematics even for positions outside the workspace.

Algorithm 1 below illustrates the HUNTER algorithm in detail:

To evaluate the proficiency of HUNTER in managing singularities, a Graphical User Interface (GUI) and a virtual model of the six-DoF assistive robot were employed, as illustrated in Figure 3.1. This intuitive GUI provides comprehensive control over the virtual robot, thereby enabling the identification of singular positions. Utilizing a Cartesian control system, the inter-

Algorithm 1 Inverse Kinematics

```
1: procedure InverseKinematics(p, θ_0)
2:     Tol ← Tolerance
3:     MaxIter ← MaximumIteration
4:     Iter ← 0
5:     Hessian ← Levenbger-Marquardt(θ_0)
6:     gradient ← Gradient(θ_0)
7:     dx ← −inv(Hessian)gradient
8:     α ← LineSearch(x, dx)
9:     s ← αdx
10:    while norm(s) > tol  and  Iter < MaxIter do
11:        θ_0 ← θ_0 + s
12:        g1 ← Gradient(θ_0)
13:        y ← g_1 − gradient
14:        H_i ← BFGS(s, y)
15:        Hessian ← Hessian + H_i
16:        if min(diag(hessian)) < 0 then
17:            Hessian ← Levenbger-Marquardt(θ_0)
18:        end if
19:        dx ← −inv(Hessian)g_1
20:        α ← LineSearch(θ_0, dx)
21:        s ← α ∗ dx
22:        Iter ← Iter + 1
23:    end while
24:    return θ_0 + s
25: end procedure
```

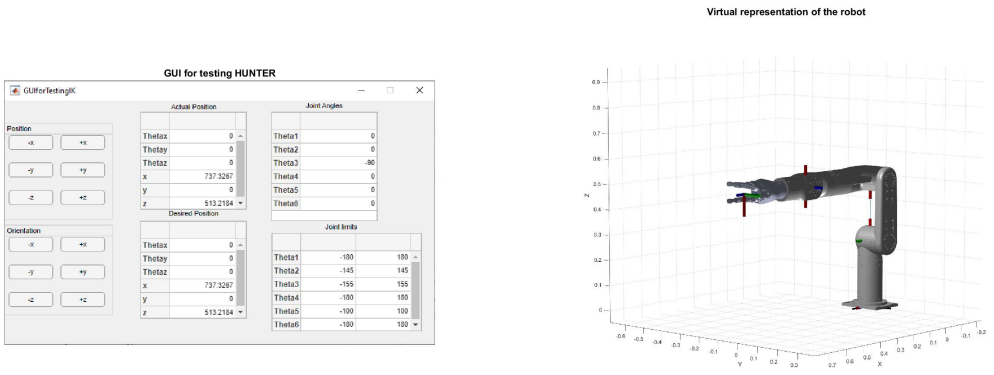


Figure 3.1: Graphical User Interface (GUI) offering comprehensive control of the virtual assistive robot for evaluating the effectiveness of HUNTER in managing singularities.

face facilitates precise manipulation of the x, y, and z coordinates, as well as the Euler angles of the end effector.

The results of the HUNTER testing are presented in Figure 3.2. This figure traces the robot's trajectory at the boundary of the workspace. As evidenced in the data, the robot first reaches its maximum position on the x-axis, then the y-axis, and lastly the z-axis.

Upon reaching the limit on the x-axis and attempting to extend to the y-axis limit, the robot traces a circular trajectory. This is because, at the x-axis limit, the robot cannot further extend in the y direction, necessitating a compensatory adjustment in its x-axis position to align with the desired y-axis direction. A similar compensation occurs when extending to the z-axis limit, with HUNTER adjusting the robot's orientations to maintain alignment with the desired position.

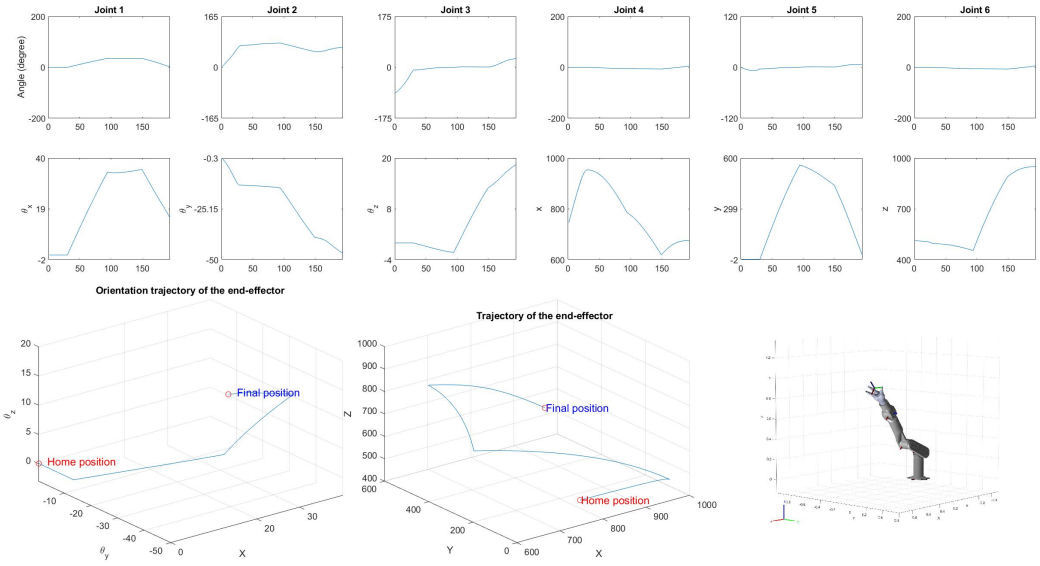


Figure 3.2: Demonstration of HUNTER's Capability in Navigating Singular Positions and Boundaries of Workspace.

This test demonstrates HUNTER's ability to navigate trajectories both outside the workspace and at singular positions. Notably, the transition throughout this process remains smooth, with no discontinuity in any of the joint angles. This suggests HUNTER's effectiveness in real-time adaptation and trajectory planning, reinforcing its potential as a solution for inverse kinematics problems in assistive robotics.

In conclusion, the ability of assistive robots to compute inverse kinematics for positions beyond their workspace, and the proposed enhancement to the BFGS method, can greatly enhance the user experience. It fosters an environment of collaboration and understanding, crucial elements in the design of user-friendly assistive robots [83].

3.3 Conclusions

This chapter explored the field of inverse kinematics with a focus on its role in assistive robotics. It began with an examination of the existing numerical techniques for inverse kinematics, including Jacobian-based methods and Quasi-Newton methods. Each technique was analyzed to understand its strengths and weaknesses. Jacobian methods are efficient and easy to implement but face challenges such as singularity. Quasi-Newton methods, specifically the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method, are robust and fast-converging, but can struggle with positions outside the typical workspace of the robot.

Following the examination of numerical techniques, the chapter introduced HUNTER (Helping Users Navigate through Extended Reach), a new algorithm designed to address the limitations of the BFGS method. HUNTER enhances the ability of the BFGS method to handle inverse kinematics for positions outside the workspace. It achieves this by adding a step that integrates the Levenberg-Marquardt (LM) method when the Hessian matrix becomes negative, thereby ensuring it remains positive definite.

The development of HUNTER is a step forward for assistive robotics. It increases the versatility and safety of these systems and promotes a greater level of interaction between the robot and the user. This increase in interaction has the potential to improve the user's experience with the assistive robot.

CHAPTER 4

ENHANCING MANIPULABILITY AND ENERGY EFFICIENCY: DEFINITION OF A NEW METHODOLOGY FOR DESIGNING ASSISTIVE ROBOTS

Performance indices, scalar functions synthesized from matrix models, depict distinct characteristics of either parallel or serial robots. As previously noted, the design of assistive robots demands an enhancement in robot manipulability, quantified through kinematic performance indices. Section 4.1 delves into these indices and section 4.2 presents a novel manipulability index to rectify the shortcomings of previous definitions. Minimizing energy consumption is another vital requirement in assistive robots design. Accordingly, section 4.3 details the indices that calculate energy consumption within a workspace. Lastly, a new methodology is presented that harmonizes the kinematic performance and energy consumption indices while integrating an object collision algorithm determining link collisions.

4.1 Kinematic performance indices

Kinematic performance indices are primarily classified based on their foundational theory: they are either based on the Jacobian matrix or on screw theory.

1. **Jacobian Matrix – Based Indices** : These indices utilize the Jacobian matrix, a crucial tool in robotics that defines the relationship between joint velocities and the linear and angular velocities of the robot's end-effector. These types of indices typically employ the concept of manipulability.

2. **Screw Theory – Based Indices** : On the other hand, indices built upon screw theory offer a robust understanding of the rigid body motion and forces involved. These indices encompass a wider range of kinematic performance attributes as they consider both motion and force transmission concurrently.

In the subsequent sections, both Jacobian matrix-based and screw theory-based indices will be thoroughly reviewed and discussed. This will provide a comprehensive overview of their respective applications, advantages, and limitations in the context of kinematic performance evaluation.

4.1.1 Kinematic Performance Based on the Jacobian Matrix

Kinematic performance indices based on the Jacobian matrix capitalize on its ability to relate output speed and differential end-effector motion with actuation joints, as well as map actuation wrench to output wrench. This mapping and relation is critical in understanding the robot's performance and potential limitations.

One widely used index is the condition number of the Jacobian matrix using the 2-norm [84], defined as:

$$\kappa_2 = \frac{\sigma_{max}(J)}{\sigma_{min}(J)} \tag{4.1}$$

Here, σ_{min} and σ_{max} denote the minimum and maximum eigenvalues respectively, and J stands for the Jacobian matrix.

Other approaches compute the condition number using the Frobenius norm [85]:

$$\kappa_F = \|\sqrt{trace(JJ^T)}\| \; \|\sqrt{trace(J^{-1}J^{-T})}\| \tag{4.2}$$

and the weighted Frobenius norm [23]:

$$\kappa_{wF} = \|\sqrt{trace(JWJ^T)}\| \; \|\sqrt{trace(J^{-1}WJ^{-T})}\| \tag{4.3}$$

In these expressions, W is a weighted matrix.

The first application of the condition number was on serial mechanisms [86]. It has been used to evaluate the dexterity and the distance to singularities of the robot [23, 84, 87, 88]. Despite its widespread use as a kinematic performance measure, its implementation has been called into question due to certain controversies.

The principal issue revolves around the inconsistencies in the dimensions of the Jacobian matrix of a mechanism with mixed Degrees of Freedom (DOFs). This muddles the physical meaning of the condition number. A common strategy to handle this issue is to divide the elements of the Jacobian matrix that have units by a characteristic or intrinsic length to render the Jacobian matrix dimensionless [89, 90]. However, this characteristic length is often geomet-

rically ambiguous. As a solution, Khan et al. [85] introduced the concept of a homogeneous space, a Euclidean space with dimensionless coordinates, and computed the condition number of the homogeneous Jacobian. In a similar vein, Pond et al. [91] proposed a dimensionless Jacobian matrix using velocities of three points on different lines of the end-effector.

In addition to condition number, the Jacobian matrix is employed to compute the manipulability of a mechanism, which is a measure of the force/velocity transmission and the distance to singularities. Yoshikawa [92] was the first to define the velocity manipulability index as:

$$\mathbf{W} = \sqrt{det(JJ^T)} \tag{4.4}$$

This definition considers the eigenvectors of J forming an ellipsoid whose axes are scaled by the eigenvalues, with the determinant of J representing the product of the eigenvalues. The determinant of the Jacobian is thus proportional to the volume of the ellipsoid, which implies that the volume signifies the robot's ability to move in any direction. If the volume equals zero, the robot is at a singularity. Thus, the larger the volume of the ellipsoid, the farther the robot's position from a singularity. Equation 4.4 does not have inconsistencies in units, making it a reliable tool for finding singularities at a given position. However, it does not provide the type of singularity, i.e., whether the singularity is rotational or translational.

Wu et al. proposed another manipulability definition that considers the minimum eigenvalue of the matrix $J^{-T}J^{-1}$ [93]:

$$\mathbf{W} = \sqrt{min(\|\lambda_{Vi}\|)} \tag{4.5}$$

According to this proposal, the larger the minimum output velocity, the better the ve-

locity performance of a manipulator. Nonetheless, the definition in equation 4.5 suffers from inconsistencies in the eigenvalues of the Jacobian matrix, as all the eigenvalues have different units.

4.1.2 Kinematic performance indices based on screw theory

Ball introduced the concept of virtual coefficient using screw theory [94]. Suppose a wrench is acting upon a rigid body on a given screw β. By the action of this wrench, the rigid body generates a twist around a screw α. This way, the virtual coefficient expresses the energy required to develop this displacement. Consequently, Yuan et al. used the virtual coefficient to relate the transmission wrench screw (TWS) with the output twist screws (OTS) to propose the transmission factor to measure the transmission quality of robots [95]. However, the transmission factor ranges between $-\infty$ to ∞. Thus, Sutherland et al. introduced a normalized version of the transmission factor called the transmission index (TI) [96]:

$$TI = \frac{|VC|}{|VC_{max}|} \tag{4.6}$$

Where $|VC|$ represents the virtual coefficient between the TWS and OTS and $|VC_{max}|$ is the maximum virtual coefficient. Another index was introduced by Tsai et al. who proposed the total transmission index (TTI) [97]. This index combines the ability of a mechanism to generate output (Transmissivity) and the capacity of an input motion to be transmitted into the TWS (Manipulability):

$$TTI = TI \times MI \tag{4.7}$$

where TI and MI are the transmissivity and manipulability indices.

Other indices consider the pressure and transmission angle at the connection between the input and output links, as presented by Takeda et al. They defined the TI based on the virtual power transmitted from the input to the output link [98] as follows:

$$TI = \min(|\cos\alpha_1|, |\cos\alpha_2|, \ldots, |\cos\alpha_n|) \tag{4.8}$$

where α_i represents the pressure angle between the output wrench and output twist generated by each actuated limb. TI is dimensionless and bounded between 0 and 1 where 0 is a singularity. Hence, TI also measures the distance to a singularity. However, this approach is only valid for wrenches with zero moments [99] or zero force [100], but not a combination of both.

Based on the concept of the virtual coefficient and the index proposed by Takeda, Wang et al. proposed the input transmission index. This index is a procedure for nonredundant spatial parallel manipulators, where the force transmission quality is expressed by input and output transmission. This way, the index is defined as follows:

$$\eta_{\text{input}} = \min(\eta_{i,i}) \tag{4.9}$$

where $\eta_{i,i}$ represents the ratio of actual power to maximal power of the i^{th} input member. Similarly, the output transmission index is given by:

$$\eta_{\text{output}} = \min(\eta_{o,i}) \tag{4.10}$$

where $\eta_{o,i}$ represents the ratio of actual power to maximal power of the i^{th} output member.

Furthermore, the force transmission index is defined as:

$$\eta = \min(\eta_{\text{input}}, \eta_{\text{output}}) \tag{4.11}$$

this index is bounded between 0 and 1, where 1 indicates a better transmission and 0 is a singularity.

Screw theory, while powerful and systematic, is not without its limitations. Specifically, it is based on an assumption that the U-joint prevents any redundant finite rotation of a link about its axis, implying that the infinitesimal rotation of the link should have a zero component along its axis [101]. This leads to the belief that the angular velocity generated by the U-joint's rotation is perpendicular to the link's axis. As a result, the Jacobian matrix computed under this assumption may not accurately represent the kinematics of the system. To date, no performance indices, to the best of available knowledge, have considered the rotation of the link along its principal axis, which signals a significant gap in the current understanding of the field.

4.2 Revising Manipulability: Introduction of a New Index

In this section, a new manipulability index is introduced, addressing the limitations highlighted in the review of chapter 4. It is proposed and substantiated that the angular velocity created by the rotation of a link attached to a U-joint is not necessarily perpendicular to the link. To validate this proposition, experimental testing of the theoretical results is conducted. While the

analytical model is adapted from the work of Carricato et al. [101], the experimental results mark a significant advancement in the understanding of this phenomenon. A novel method for computing the Jacobian matrix is then suggested, one that includes a precise calculation of angular velocity. Finally, a new manipulability index for the computation of manipulability is presented, integrating this exact Jacobian matrix, thus opening new pathways in the design and analysis of assistive robots.

4.2.1 Rotation analysis of universal joints

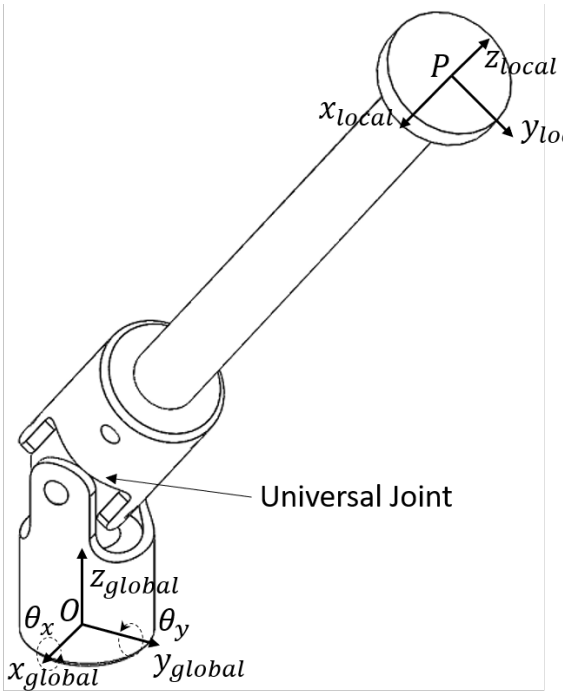


Figure 4.1: Link attached to an universal joint

To compute the components of the angular velocity generated by the actuation of universal joints, consider a link attached to a universal joint as presented in figure (4.1). In this figure,

the global and local coordinate systems are attached to the points O and P, respectively. Since the universal joint allows the rotation in two perpendicular axes, the orientation of the local coordinate system relative to the global coordinate system is given by doing two successive rotations in the X and Y, as presented in the following equation:

$$\begin{aligned} {}^{global}R_{Local} &= R_x R_y \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \\ &= \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ \sin\theta_x \sin\theta_y & \cos\theta_x & -\cos\theta_y \sin\theta_x \\ -\cos\theta_x \sin\theta_y & \sin\theta_x & \cos\theta_x \cos\theta_y \end{bmatrix} \end{aligned} \tag{4.12}$$

The matrix ${}^{global}R_{Local}$ is orthogonal, therefore it satisfies the following property:

$${}^{global}R_{Local}\ {}^{global}R^T_{Local} = I_{3x3} \tag{4.13}$$

Where I_{3x3} represents the identity matrix. The term T stands for the transpose. Then differentiating equation (4.13) with respect to time, the following expression is obtained:

$$\frac{d({}^{global}R_{Local})}{dt}\ {}^{global}R^T_{Local} + {}^{global}R_{Local}\ \frac{d({}^{global}R^T_{Local})}{dt} = 0 \tag{4.14}$$

Note that in the left side of equation (4.14) the first term is equivalent to the transpose of the second term. Then, since by adding those two terms the result is zero and one expression is the

transpose of another, both matrices must be skew-symmetric. The terms of the generated skew-symmetric matrix corresponds to the components of angular velocity of the link, as presented below:

$$\begin{aligned}\frac{d({}^{global}R_{Local})}{dt}\ {}^{global}R_{Local}^{T} &= [\omega_{global}]_{\times} \\ &= \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}\end{aligned} \tag{4.15}$$

Where ω_x, ω_y, and ω_z are the components of the term ω_{global}, which corresponds to the angular velocity measured in the global system. Then, after differentiating equation (4.15) and organizing the terms, the expression for the angular velocity in the global coordinate system due to the rotations of the universal joint is obtained:

$$\omega_{global} = \begin{bmatrix} \dot{\theta}_x \\ \dot{\theta}_y \cos\theta_x \\ \dot{\theta}_y \sin\theta_x \end{bmatrix} \tag{4.16}$$

Finally, this angular velocity is computed in the local coordinate system by applying the transpose of the rotation matrix, as follows:

$$\begin{aligned}\omega_{Local} &= {}^{global}R_{Local}^{T}\ \omega_{global} \\ &= \begin{bmatrix} \dot{\theta}_x \cos\theta_y \\ \dot{\theta}_y \\ \dot{\theta}_x \sin\theta_y \end{bmatrix}\end{aligned} \tag{4.17}$$

Note that term ω_{Local} has a component along the axis of the link (Z_{Local}). This result demonstrates that the angular velocity is not perpendicular to the axis of the link. A similar result was obtained by Carricato et al. in [101]. Note that this result is not only limited to U-joints, but it is a applicable to any combination of pair of perpendicular joints. Thus, it is a limitation of the screw theory that affects the calculation of the angular velocity and all the derivative measures [102].

4.2.2 Experimental validation

The results of equation (4.17) were obtained analytically, in the following, these results are contrasted with the experimental results. The validation was carried out using the prototype for experimentation presented in figure (4.2). This prototype is actuated by the utilization of cables that control the orientation of the moving platform. As can be seen, the moving platform is attached to a fixed base by the use of a U-joint. To control the motion of the moving platform, the inverse kinematics are considered as presented below:

$$L_i = \sqrt{\Delta L_{xi}^2 + \Delta L_{yi}^2 + \Delta L_{zi}^2} \tag{4.18}$$

Where the terms ΔL_{xi}, ΔL_{yi}, and ΔL_{zi} are presented as follows:

$$\begin{aligned}
\Delta L_{xi} &= \mathrm{R_a}\cos(\alpha_\mathrm{i}) - \mathrm{R_b}\cos(\alpha_\mathrm{i})\cos(\theta_\mathrm{y}) - \mathrm{h_b}\cos(\theta_\mathrm{x})\sin(\theta_\mathrm{y}) - \mathrm{R_b}\sin(\alpha_\mathrm{i})\sin(\theta_\mathrm{x})\sin(\theta_\mathrm{y}) \\
\Delta L_{yi} &= \mathrm{R_a}\sin(\alpha_\mathrm{i}) + \mathrm{h_b}\sin(\theta_\mathrm{x}) - \mathrm{R_b}\sin(\alpha_\mathrm{i})\cos(\theta_\mathrm{x}) \\
\Delta L_{zi} &= \mathrm{R_b}\cos(\alpha_\mathrm{i})\sin(\theta_\mathrm{y}) - h - \mathrm{h_b}\cos(\theta_\mathrm{x})\cos(\theta_\mathrm{y}) - \mathrm{R_b}\sin(\alpha_\mathrm{i})\cos(\theta_\mathrm{y})\sin(\theta_\mathrm{x})
\end{aligned} \tag{4.19}$$

The coefficients of equation (4.19) are presented in figure (4.3). The angles θ_x and θ_y represent the angular orientation of the moving platform, while the angles α_i represent the position of

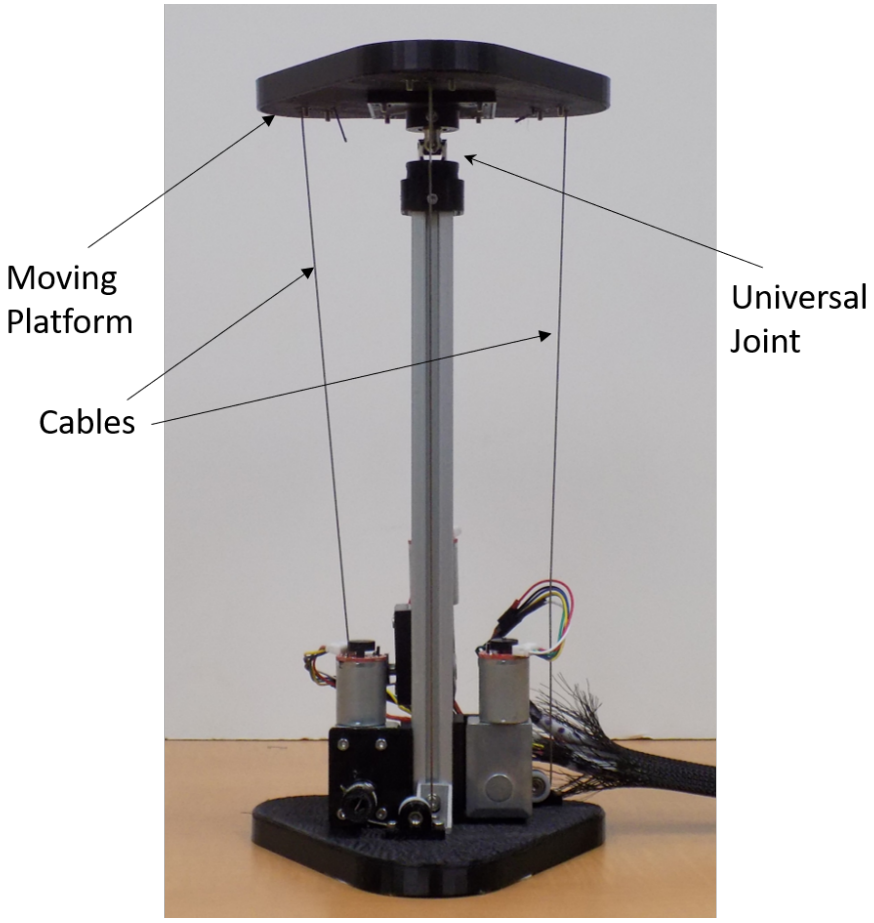


Figure 4.2: Prototype for experimentation

the A_i and B_i points.

To generate the trajectory of the moving platform, the quintic polynomial trajectory is considered as represented in the following equation:

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{4.20}$$

Assuming initial and final values of zero for the velocity and acceleration, the coefficients of

equation (4.20) are computed by solving the following linear problem:

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^4 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} \theta_0 \\ \dot{\theta}_0 \\ \ddot{\theta}_0 \\ \theta_f \\ \dot{\theta}_f \\ \ddot{\theta}_f \end{bmatrix} \tag{4.21}$$

Then, table (4.1) presents the values of θ_x, θ_y, and time for the experimentation trajectory. Note that the angular values presented in table (4.2) are in degree and equation (4.18) uses the values in radians. To measure the orientation and angular velocity of the moving platform, a Life-

Table 4.1: Values of θ_x, θ_y, and time for the experimentation trajectory.

Time (s)	θ_x (deg)	Time (s)	θ_y (deg)
2.0	-10.0	2.0	5.0
3.6	-18.0	3.6	16.3
5.0	-8.3	4.1	13.0
6.7	2.6	5.0	4.6
9.0	16.0	6.3	-8.6
10.0	6.8	7.4	-4.5
11.3	-9.3	8.4	1.5
13.0	-13.2	9.2	4.2
15.0	9.6	9.7	0.7
15.7	5.3	10.2	-6.7
16.5	-5.5	11.0	-13.8
16.9	-11.5	11.3	-14.7
17.6	-13.9	13.4	14.0
18.2	-12.0	14.1	8.9
18.5	-9.0	14.4	6.8
19.1	-2.5	14.6	4.4
		14.9	1.5
		15.7	-3.0
		16.3	-7.8
		17.3	1.0
		18.2	8.2
		19.1	7.0

Table 4.2: Coefficients for quintic function

c_5	c_4	c_3	c_2	c_1	c_0
1.99×10^3	-4.97×10^3	3.32×10^3	0	0	0

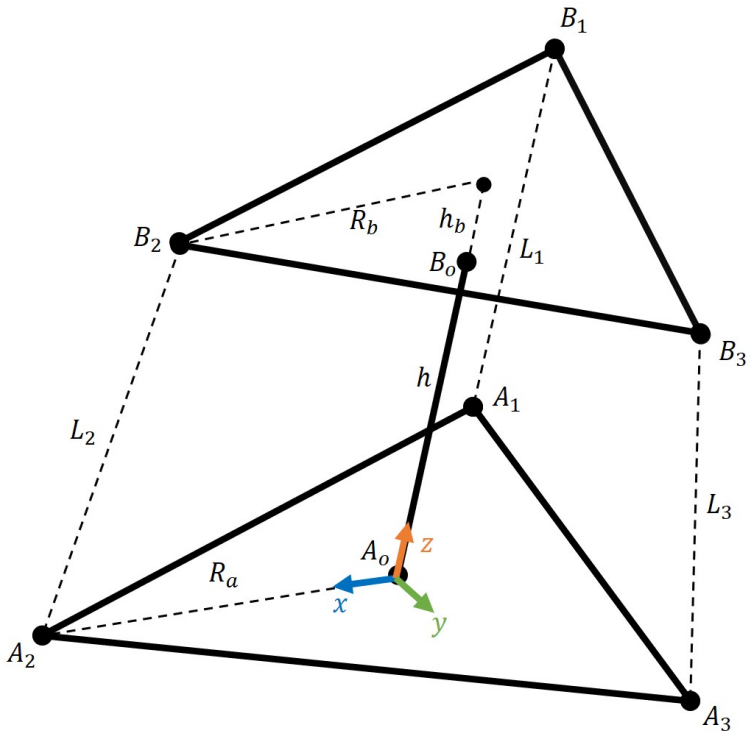


Figure 4.3: Kinematic diagram of the prototype for experimentation presented in Fig 4.2

Performance-Research (LP-Research) motion sensor [103] was installed. This sensor measures the orientation using Euler angles with the rotation sequence ZYX with a resolution of 0.01

deg. Then, the equivalent rotation matrix is obtained by using the following equation:

$$
\begin{aligned}
R_{Sensor} &= R_z R_y R_x \\
&= \begin{bmatrix}
\cos\hat{\theta}_y \cos\hat{\theta}_z & \sin\hat{\theta}_x \sin\hat{\theta}_y \cos\hat{\theta}_z - \cos\hat{\theta}_x \sin\hat{\theta}_z & \sin\hat{\theta}_x \sin\hat{\theta}_z + \cos\hat{\theta}_x \sin\hat{\theta}_y \cos\hat{\theta}_z \\
\cos\hat{\theta}_y \sin\hat{\theta}_z & \cos\hat{\theta}_x \cos\hat{\theta}_z + \sin\hat{\theta}_x \sin\hat{\theta}_y \sin\hat{\theta}_z & \cos\hat{\theta}_x \sin\hat{\theta}_y \sin\hat{\theta}_z - \sin\hat{\theta}_x \cos\hat{\theta}_z \\
-\sin\hat{\theta}_y & \cos\hat{\theta}_y \sin\hat{\theta}_x & \cos\hat{\theta}_x \cos\hat{\theta}_y
\end{bmatrix}
\end{aligned}
\tag{4.22}
$$

Where $\hat{\theta}$ is used to represent the angles obtained from the sensor. The values of R_{Sensor} were compared with the rotation matrix R from equation (4.12) to obtain the values of θ_x and θ_y using the following equations:

$$
\begin{aligned}
\tan\theta_y &= \frac{\sin\hat{\theta}_x \sin\hat{\theta}_z}{\cos\hat{\theta}_y \cos\hat{\theta}_z} \\
\tan\theta_x &= \frac{\cos\hat{\theta}_y \sin\hat{\theta}_x}{\cos\hat{\theta}_x \cos\hat{\theta}_z + \sin\hat{\theta}_x \sin\hat{\theta}_y \sin\hat{\theta}_z}
\end{aligned}
\tag{4.23}
$$

Equation (4.23) represents the equivalent angles of the U-joint, since, as stated in equation (4.12), only two angles are required to describe the rotations of a U-joint. Then, the values of the angles obtained from the sensor and the angles from applying equation (4.23) are presented in figure (4.5). Those angles were obtained by following the trajectory presented in figure (4.4) applying equation (4.18) to compute the cables lengths. Note that the sensed angles $\hat{\theta}_x$ and $\hat{\theta}_y$ have almost identical values to θ_x and θ_y, respectively. Note also that the zones in which the differences $\|\theta_x - \hat{\theta}_x\|$ and $\|\theta_y - \hat{\theta}_y\|$ increase coincides with the higher values of $\hat{\theta}_z$. This error occurs because the U-Joint has a backslash of ± 4 deg, which adds an additional DOF to the U-joint at some positions. This error is better perceived when comparing the trajectory used to control the position of the prototype for experimentation and the lengths computed by inputting the sensed angles into equation (4.18), as shown in Figure (4.4). This figure

presents the comparison between the experimental path and the inverse kinematics; as can be seen, the difference between the two graphs increases at the changes in the trajectory direction.

However, besides the existence of the backslash, the maximum absolute error between the sensed angles and the analytical angles is less than 3 deg, which is considered acceptable for this application.

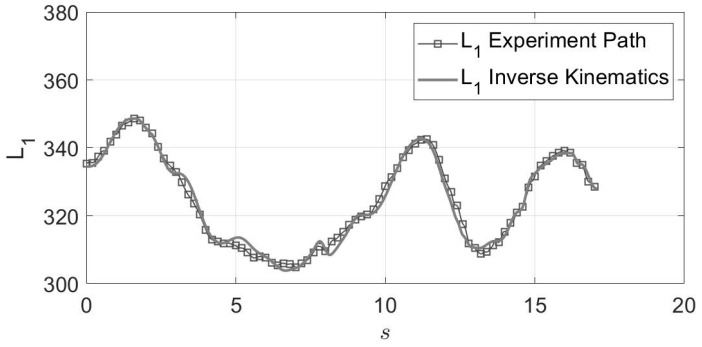
In the case of the angular velocity, the sensor has a range of ±125 deg/s with a resolution of 16 bits, and measures with respect to the global coordinate system. The angular values from the sensor were used in equation (4.16), to obtain the values of $\dot{\theta}_x$ and $\dot{\theta}_y$, as presented in the following equations:

$$\begin{aligned} \dot{\theta}_x &= \hat{\omega}_x \\ \dot{\theta}_y &= \frac{\hat{\omega}_z}{\cos\theta_x} \end{aligned} \tag{4.24}$$

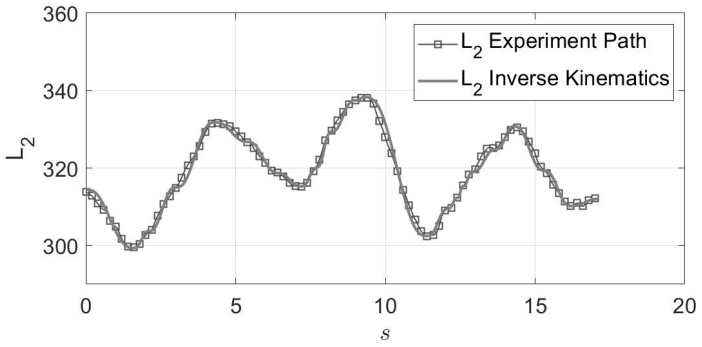
The term $\hat{\omega}_x$ represents the value of the sensed global angular velocity. The results from equations (4.24) and (4.23) were used in equation (4.17) to compute ω_{Local}, which is considered the analytical local angular velocity. Otherwise, the sensed local angular velocity is computed using the following equation:

$$\hat{\omega}_{Local} = R^T_{Sensor}\hat{\omega} \tag{4.25}$$

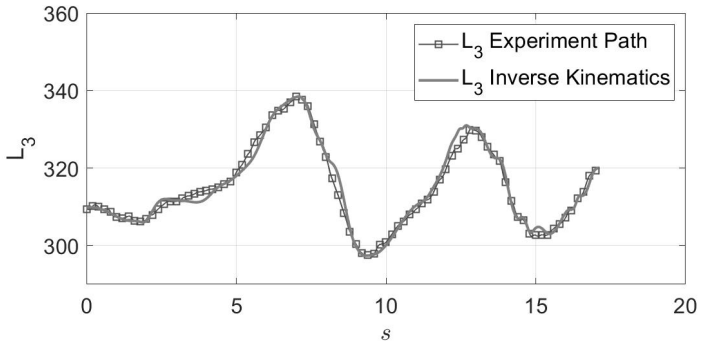
Figure (4.6) presents the comparison between each of the components of $\hat{\omega}_{Local}$ and ω_{Local}. As can be seen, both angular velocities describe almost the same behavior in the x and y directions. However, in the case of the z direction, although both graphs show similar behavior, the error is considerably bigger than the found in the x and y directions. The reason for this remains to be the backslash of the U-joint described earlier, noting that the additional DOF along the z axis



(a)

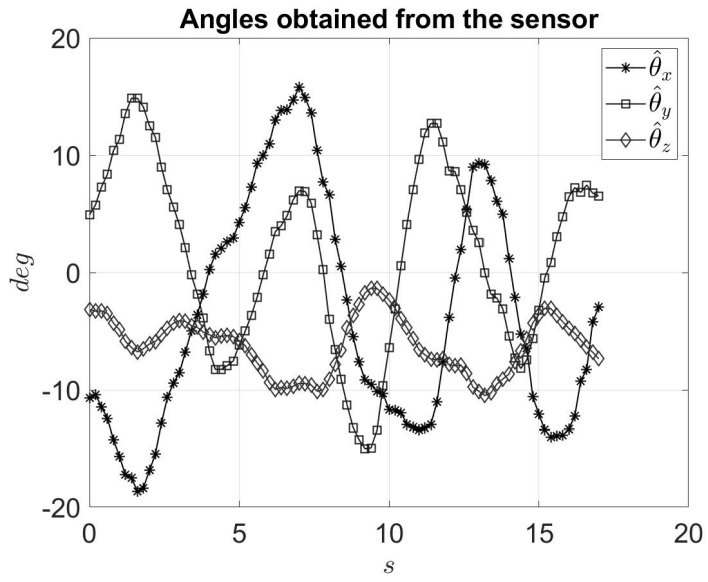


(b)

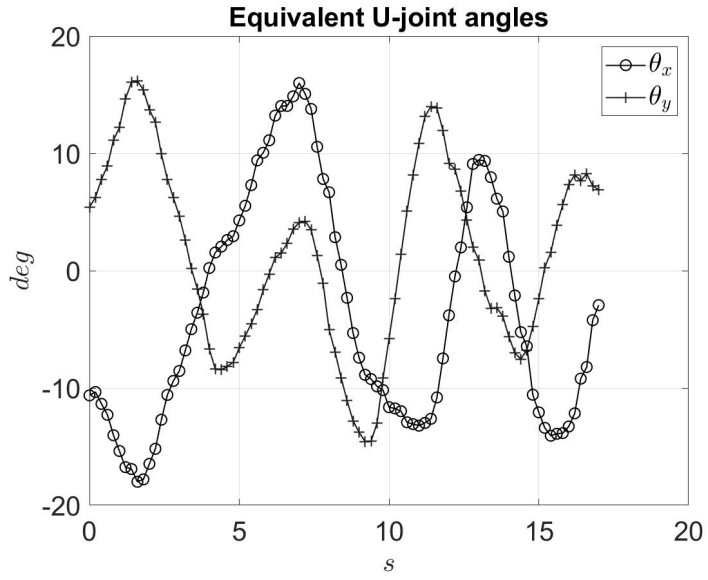


(c)

Figure 4.4: Comparison between the trajectory used in the experiment and the lengths computed by inverse kinematic for: a) actuator 1, b) actuator 2, and c) actuator 3



(a)



(b)

Figure 4.5: a) Measured angles from the sensor. b) Equivalent U-joint angles.

makes the projection of the angular velocity in this direction smoother for the sensed values. Still, besides this error, the graph shows that along the z direction of the moving platform there is a component of the angular velocity, proving that the components of the angular velocity in the local coordinate system are not perpendicular. This result also proves that equation (4.17) is a reliable tool to describe the angular velocity of a link attached to a U-joint.

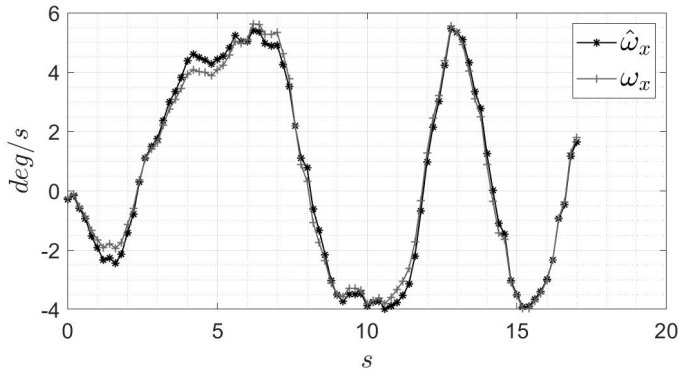
4.2.3 Jacobian matrix calculation

The calculation of the Jacobian matrix follows the result from equation (4.17), but instead of considering only two rotations, the calculation of the Jacobian is generalized for multiple degrees of freedom (DOF). Thus, to generalize the calculation of the Jacobian matrix, consider a homogeneous transformation matrix (HTM):

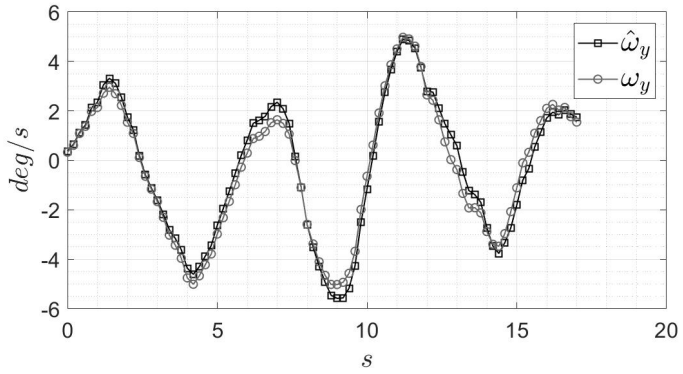
$$ {}^{0}T_{ee} = \begin{bmatrix} {}^{0}R_{ee} & {}^{0}P_{ee} \\ 0 & 1 \end{bmatrix} \tag{4.26} $$

Where ${}^{0}R_{ee}$ is a 3×3 rotation matrix that represents the orientation of the end-effector with respect to the global coordinate frame (frame 0). Also, ${}^{0}P_{ee}$ represents the position of the end effector relative to frame 0. Note that in a serial robot the HMT of the end-effector is a function of all the joints. Thus, using equation (4.26), the following equation presents the proposed method for the calculation of the Jacobian matrix:

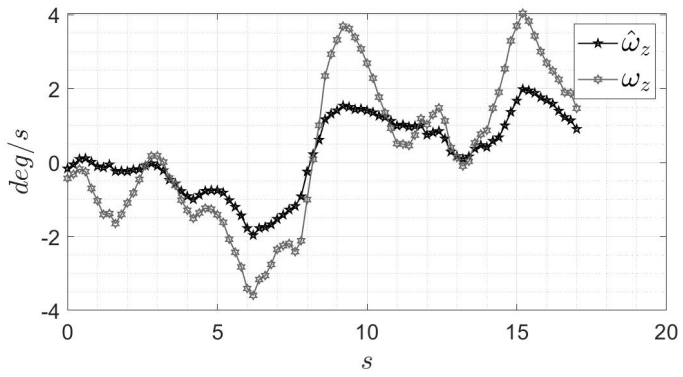
$$ J = \begin{bmatrix} J_t \\ J_\theta \end{bmatrix} \tag{4.27} $$



(a)



(b)



(c)

Figure 4.6: Comparison between the sensed local angular velocity and the analytical local angular velocity in the a) x-direction, b) y-direction, and c) z-direction.

Where J_t is the represents the translational Jacobian matrix, which relates the change in position of the end-effector with the change in position of the joint angles. J_T is computed using the traditional equation:

$$J_t = \begin{bmatrix} \frac{\delta\ {}^0P_{ee}}{\delta\theta_1} & \frac{\delta\ {}^0P_{ee}}{\delta\theta_2} & \dots & \frac{\delta\ {}^0P_{ee}}{\delta\theta_n} \end{bmatrix} \tag{4.28}$$

In the case of J_θ, the Jacobian matrix is computed for the rotation along the three principal directions, as follows:

$$J_\theta = \begin{bmatrix} J_{\theta_x} \\ J_{\theta_y} \\ J_{\theta_z} \end{bmatrix} \tag{4.29}$$

Where the terms J_{θ_x}, J_{θ_y}, and J_{θ_z} are obtained using the results of equation (4.17) as follows:

$$\begin{aligned} J_{\theta_x} &= \begin{bmatrix} {}^0R_6(3,1) & {}^0R_6(3,2) & {}^0R_6(3,3) \end{bmatrix} \begin{bmatrix} \frac{\partial\ {}^0R_6^T}{\partial\theta}(2,1) \\ \frac{\partial\ {}^0R_6^T}{\partial\theta}(2,2) \\ \frac{\partial\ {}^0R_6^T}{\partial\theta}(2,3) \end{bmatrix} \\ J_{\theta_y} &= \begin{bmatrix} {}^0R_6(1,1) & {}^0R_6(1,2) & {}^0R_6(1,3) \end{bmatrix} \begin{bmatrix} \frac{\partial\ {}^0R_6^T}{\partial\theta}(3,1) \\ \frac{\partial\ {}^0R_6^T}{\partial\theta}(3,2) \\ \frac{\partial\ {}^0R_6^T}{\partial\theta}(3,3) \end{bmatrix} \\ J_{\theta_z} &= \begin{bmatrix} {}^0R_6(2,1) & {}^0R_6(2,2) & {}^0R_6(2,3) \end{bmatrix} \begin{bmatrix} \frac{\partial\ {}^0R_6^T}{\partial\theta}(1,1) \\ \frac{\partial\ {}^0R_6^T}{\partial\theta}(1,2) \\ \frac{\partial\ {}^0R_6^T}{\partial\theta}(1,3) \end{bmatrix} \end{aligned} \tag{4.30}$$

Equation (4.27) is used to compute the Jacobian matrix from the HTM.

4.2.4 New manipulability index

Using the outcomes from Equation (4.27), two positive definite matrices are established as follows:

$$\begin{aligned} J_t^+ &= J_t J_t^T \\ J_\theta^+ &= J_\theta J_\theta^T \end{aligned} \tag{4.31}$$

It is worth highlighting that matrices J_t^+ and J_θ^+ are positive definite in regions excluding singularities; in regions with singularities, they are positive semi-definite. The eigenvalues of these matrices are used to formulate the new manipulability index, denoted by $\mathcal{K}$.

$$\mathcal{K} = \frac{\sigma_{t,min}\sigma_{\theta,min}}{\sigma_{t,max}\sigma_{\theta,max}} \tag{4.32}$$

In this equation, σ_{min} and σ_{max} are the smallest and largest eigenvalues of matrices J_t^+ and J_θ^+, respectively. This updated manipulability index circumvents inconsistencies linked to the eigenvalue units, while preserving the range from 0 to 1. It is equally applicable to parallel and serial robots.

Another valuable attribute of this new manipulability index is its capability to detect the nature of singularities. For example, if the smallest eigenvalue of J_t^+ is zero, it signals a direction in which the robot is unable to move. Likewise, if the smallest eigenvalue of J_θ^+ is zero, it suggests a direction where the robot cannot execute a rotation.

4.3 Energy consumption indices

Energy consumption indices (ECIs) are scalar functions that determine the power required to perform a given trajectory. ECIs usually express energy consumption by measuring the torque used by the robot. The most common ECIs available are the quadratic average torque (QAT), the weighted root mean square (WRMS), and the absolute sum of torques (AST).

4.3.1 Quadratic average torque (QAT)

The QAT is defined as the quadratic average of torques using the following equation:

$$QAT = \int_0^T \sum_{\text{joint}=1}^{n_j} \left(\frac{\tau_{\text{joint}}(t)}{\tau_{joint}^{max}} \right)^2 dt$$

where T is the trajectory time. In this context, τ_{joint} represents the torque of the jointth of the robot, and n_j represents the total number of joints. However, as explained in [104], this measure is used for trajectory generation.

4.3.2 Weighted Root Mean Square (WRMS)

The weighted sum as presented in [105] requires the calculation of each individual joint torque within the workspace. Then, the following equation is applied to compute the WRMS:

$$WRMS = \sum_{joint=1}^{n_j} w_{joint} \tau_{joint,rms} \tag{4.33}$$

where w_{joint}'s is a weight assigned to each joint torque. The weights are selected such that their sum is equal to 1, and, according to [105], each torque uses the same weight for each joint

torque τ_{joint}. Note that Equation (4.33) requires to know the torque of the jointth of the robot through the trajectory.

4.3.3 The Absolute Sum of Torques (AST)

In [106], the authors presented an index for measuring the energy consumption in a trajectory. The index is expressed as follows:

$$E_{robot} = \int_{o}^{t} W_r(t)dt \tag{4.34}$$

where W_r is the robot energy consumption, obtained by the sum of the individual energy consumption of each joint, as follows:

$$W_r(t) = \sum_{joint=1}^{6} W_{m,joint}(t) \tag{4.35}$$

The main limitation of these indices is that they are implemented for trajectory optimization.

4.3.4 Selecting the proper ECIs

In the cited work [107], a comparative investigation of three different torque indices is conducted, with the focus on the optimization of assistive robots. The indices were modified to facilitate workspace optimization, rather than trajectory optimization, thus making them more suitable for assistive robots design. The optimal solutions derived from these indices are compared, illustrating that solutions from the AST and the WRMS are equivalent, leading to an expansive workspace but higher torque consumption. On the other hand, the optimal solution using the

QAT results in a more compact workspace with lesser torque consumption. Thus, all three torque measures produce valid optimal solutions. The study concludes by asserting that the algorithm presented is a reliable tool for designing not only assistive robots but also serial robots. Consequently, in this research, the selected energy index for torque minimization is the QAT, which for workspace optimization is defined as follows:

$$QAT_i = \sum_{\text{joint}=1}^{n_j} \left(\frac{\tau_{\text{joint}}}{\tau_{joint}^{max}} \right)^2 \tag{4.36}$$

In this equation, the subindex i indicates that the index is computed locally, at each position within the workspace. Thus, the global computation of the index is performed using the subsequent equation:

$$QAT_{global} = \sum_{i=1}^{N} QAT_i \tag{4.37}$$

4.4 Methodology for the integration of manipulability and energy consumption indices in the design of assistive robots.

The novel manipulability index introduced in equation (4.32) and the global QAT, as outlined in equation (4.37), are integrated using the methodology shown in figure (4.7). The steps of this methodology are explained as follows:

1. **Inputs** In this step the algorithm asks the user for three variables, W, m_{obj}, and $\vec{L}$. Where W represents the whole workspace, which is the sum of the seven workspaces described

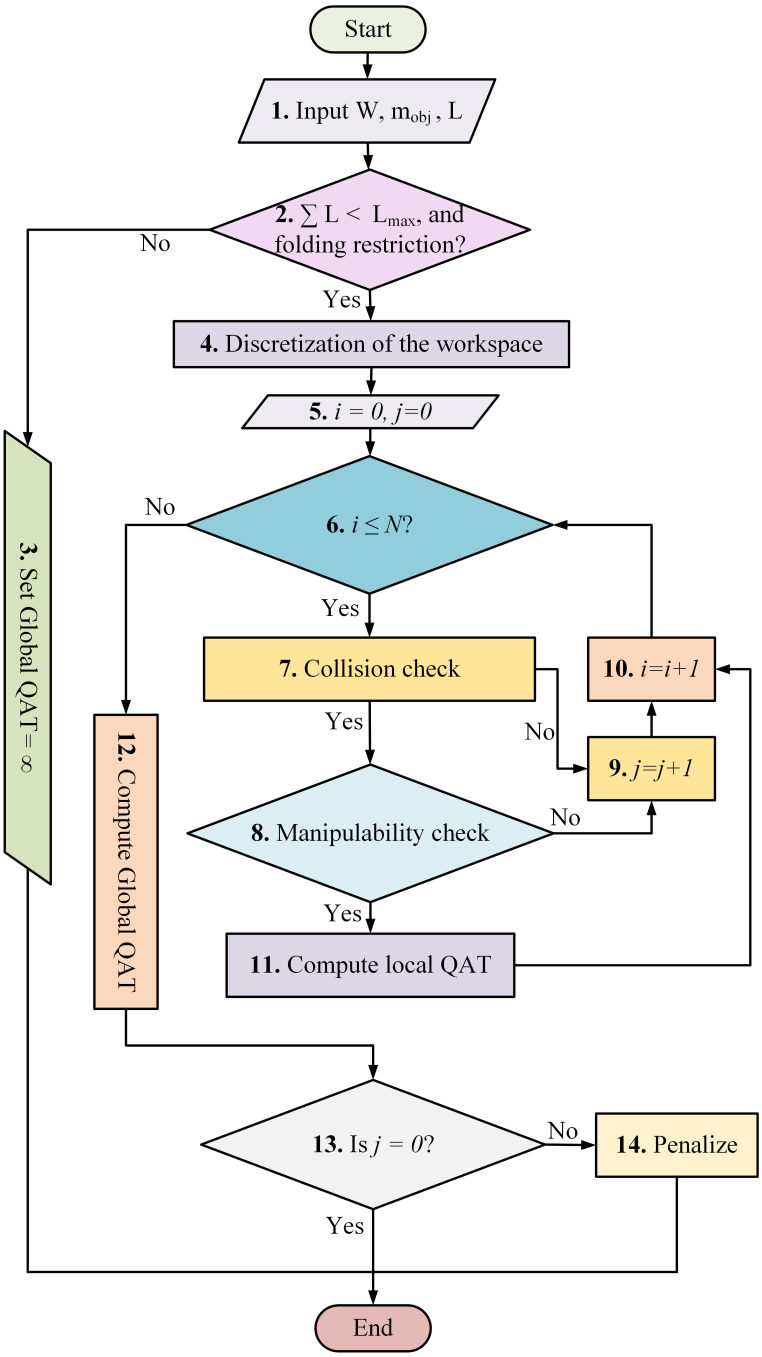


Figure 4.7: Methodology for Workspace Evaluation: Reducing Torque Consumption, Increasing Coverage, and Minimizing Object Collisions and Singularities.

in section (2.2.1). Thus, the W is expressed as follows:

$$\{W_1, W_2, W_3, W_4, W_5, W_6, W_7\} \in W \tag{4.38}$$

lastly, m_{obj} represents the weight of the carried object.

2. **Dimensional Constraints** Two dimensional constraints exist, one being that the cumulative length of the links must not exceed the maximum length. This is mathematically represented as:

$$\sum_{i}^{n} L_i = L_{max}$$

The second constraint takes into account the foldability, an element that depends on the configuration of the link. Details of this constraint will be explored in chapter 4 where the optimization is implemented.

3. **Gobal QAT to infinity** In this step, if the dimensional constraints are not met, a value of ∞ is assigned. This symbol represents the largest possible value that can be stored in a floating-point data type, essentially indicating that the configuration fails the constraints.

4. **Discretization of the Workspace**: The seven workspaces discussed in section 2.2.1 are discretized, with the total number of points from each discretized workspace being summed up to yield N. By examining each i member of the discretization, it is possible to evaluate the entire workspace comprehensively.

5. **Initialize Workspace Counter and Penalization Counter**: In this stage, two counters are initialized. The first counter tracks the current i position within the workspace,

while the second counter is for positions that require penalization (initialized with $j = 0$). Together, these counters aid in monitoring the progression of workspace exploration.

6. **Establish While Loop Condition**: In this context, a 'while' loop is implemented to ensure the comprehensive exploration of the workspace. Specifically, this loop continues to run until every point in the workspace has been checked, i.e., until i equals N.

7. **Collision detection** This step involves verifying if the robot is in a self-collision state. The specifics of implementing the collision detection algorithm depend on the particular robot under study. A more in-depth explanation of this process is provided in Chapter 4.

8. **Kinematic checker**: At this stage, the methodology checks if the local position i is in a singularity by using the new manipulability checker as outlined in equation (4.32). This step ensures that each position in the workspace does not result in a robotic configuration that could lead to undefined or problematic robot behavior.

9. **Penalization Counter Increment** If the robot is either colliding or positioned at a singularity at the local position 'i', the penalization counter is incremented by one. This step ensures that configurations that do not satisfy the non-collision or kinematic checks are appropriately accounted for and penalized in the optimization process.

10. **Workspace Counter Increment**: This counter increases by one during each iteration of the while loop. Its purpose is to track the progression through the local positions within the workspace, ensuring that every position is accounted for in the process.

11. **Compute local QAT** Hence, if the robot is not in a state of self-collision at the current local position and passes the kinematic check, the QAT is calculated as depicted

in Equation (4.36). This ensures that the performance index used for optimization is only evaluated for valid configurations that meet both the non-collision and kinematic conditions

12. **Global QAT Computation** The final stage involves the calculation of the Global QAT, which is achieved by implementing Equation (4.37). This calculation takes into account all the local QAT computations made for each position in the workspace, culminating in an overall torque assessment for the assistive robot operation.

13. **Penalization Counter Check** This stage assesses if there is a need for applying any penalizations based on the outcomes of the previous stages. If any singular positions or collisions were detected during the exploration of the workspace, it signals that penalizations are necessary.

14. **Penalization** This stage applies penalization to the global QAT using the equation:

$$\overline{QAT_{global}} = \left(\frac{N}{N-j}\right)^2 QAT_{global} \tag{4.39}$$

This equation ensures that any detected singular positions or collisions within the workspace are taken into account, adjusting the global QAT accordingly. The corrected global QAT, denoted as $\overline{QAT_{global}}$, provides a more accurate representation of the system's performance taking into consideration its operational constraints.

In conclusion, this section presents an in-depth methodology that facilitates the integration of a novel manipulability index with the global QAT for the purpose of optimizing the design of assistive robots. This process takes into account crucial aspects such as workspace discretiza-

tion, collision detection, singularity avoidance, and constraint satisfaction, among others. The comprehensive nature of this approach makes it a versatile and robust tool for evaluating the performance of assistive robots under various conditions. The implementation of this methodology will be further explored in Chapter 4, providing a practical demonstration of its efficacy and adaptability in the design of assistive robots.

4.5 Conclusions

In conclusion, performance indices are essential tools that provide insight into the functionality and efficiency of both parallel and serial robots. Section 4.1 provides an in-depth discussion on the development and understanding of these indices, mainly focusing on two primary theories: Jacobian matrix-based indices and screw theory-based indices.

However, it's important to acknowledge the distinct limitations of both these theories. Jacobian matrix-based indices grapple with inconsistencies arising from the units of the Jacobian matrix, which can complicate their practical use and interpretation. On the other hand, indices grounded on screw theory rely on an assumption that a U-joint restricts any unnecessary finite rotation of a link around its axis. This assumption infers that the infinitesimal rotation of the link would lack any component along its axis. This concept can potentially lead to a skewed representation of the system's kinematics, underscoring the need for a more accurate and nuanced understanding of these mechanisms.

The introduction of a novel manipulability index, as discussed in Section 4.2, marks a significant stride towards addressing limitations inherent in previous models. This newly proposed index incorporates a more precise representation of the kinematics involved in the operation of

a robot, particularly one fitted with a U-joint. One of the key insights offered by this work is the understanding that the angular velocity generated by a U-joint's rotation is not necessarily perpendicular to the link's axis.

In addition to providing a more accurate depiction of kinematics, another salient feature of this new manipulability index lies in its ability to discern the type of singularities within the system. It does this by examining the eigenvalues of the respective matrices $J^{+}t$ and $J^{+}\theta$. These additional insights from the new manipulability index further its usefulness in the design and operation of robots, making it a promising tool for future research and application in robotics.

Furthermore, the necessity of minimizing energy consumption in the design of assistive robots has been a crucial aspect of the discussion, as examined in detail in the energy consumption section (4.3). Various indices that compute energy consumption within a robot's workspace were explored, providing a comprehensive analysis of their functionalities and applications.

An interesting study investigated the comparative performance of three different torque indices, focusing specifically on their optimization for assistive robots. These indices were innovatively modified to focus on workspace optimization rather than trajectory optimization, thereby enhancing their suitability for the design of assistive robots. The comparative study yielded intriguing results: the solutions derived from the AST and WRMS were identical, resulting in a larger workspace but at the cost of increased torque consumption. In contrast, the solution using the QAT resulted in a more confined workspace but with the advantage of reduced torque consumption.

All three torque measures yielded valid optimal solutions, illustrating the versatility and effectiveness of these indices. Notably, the QAT was selected as the preferred energy index for

torque minimization in this research due to its capacity to balance workspace optimization with torque consumption. This index was calculated locally at each position within the workspace, with the global computation of the index performed by aggregating the local computations.

The work has also yielded a new methodology that integrates the aspects of kinematic performance and energy consumption, while incorporating a collision algorithm that successfully detects and avoids potential link collisions.

CHAPTER 5

CRAFTING PERFECTION: THE OPTIMAL DESIGN OF AN ASSISTIVE ROBOT

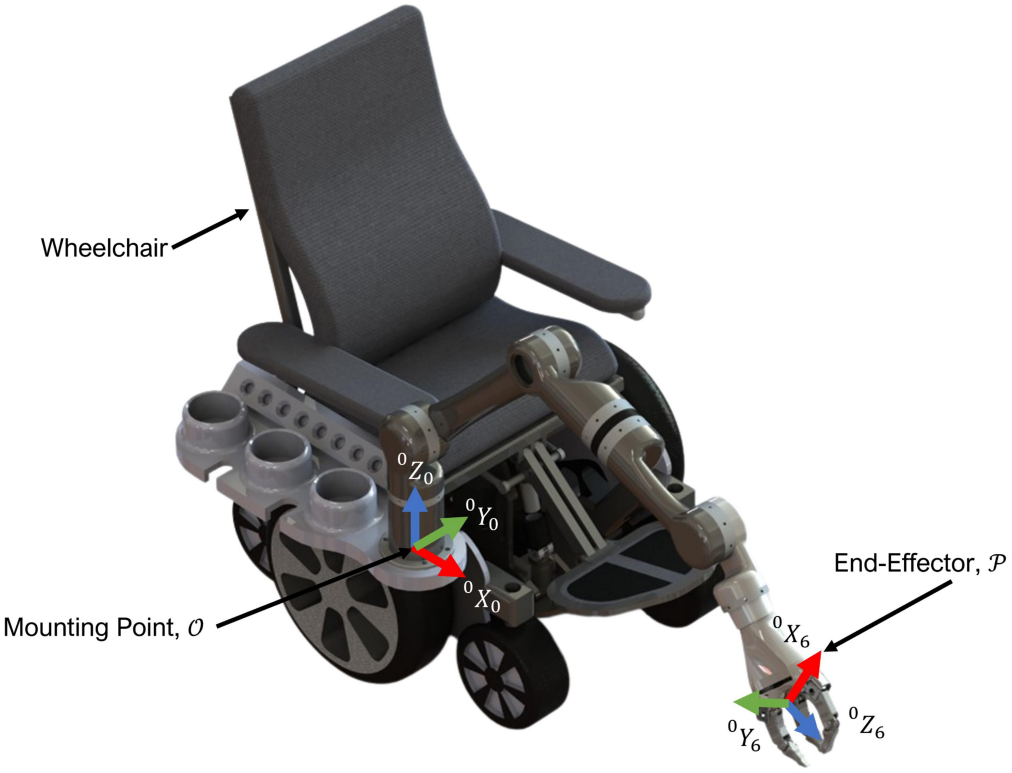
In this chapter, the hands-on application of designing an assistive robot is explored. This exercise employs an integrative methodology that combines manipulability and energy consumption indices, originally presented in Chapter 4.

Initially, the chapter outlines the essential characteristics and specifications that shape the robot's design. These parameters include the Denavit-Hartenberg representation, the kinematics, criteria for considering folding, the dynamic model, and the collision avoidance algorithm.

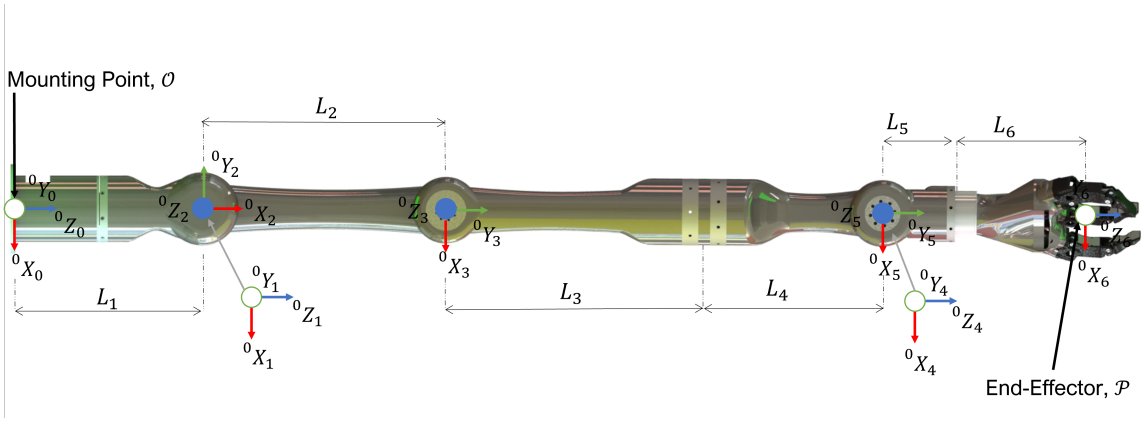
With these foundational elements defined, the exploration proceeds into the design phase, formulating an objective function for optimization. This step leverages the Scatter Search algorithm, a potent tool that aids in achieving an optimal design for the assistive robot.

Upon reaching a design, the effectiveness is validated through comprehensive simulation tests. These tests verify if the robot can cover the required workspace, demonstrate the ability to fold, and support a 3kg weight midrange within the workspace. This exhaustive approach ensures that the design meets the desired efficiency and reliability criteria for an assistive robot.

5.1 Assistive robot description



(a)



(b)

Figure 5.1: wheelchair-mounted assistive robot (a) CAD (b) Joint Coordinate Definition.

This research primarily focuses on a wheelchair-mounted robot with 6-DOF that is designed to assist individuals with ULED in ADL. The robot in question is a serial manipulator, consisting of six revolute joints.

The robot's configuration, as depicted in Fig. 5.1b, is defined by the Denavit Hartenberg (DH) parameters. These parameters, as shown in Table-5.1, describe the relationship between neighboring links in the robot arm.

Table 5.1: DH Parameters of the Assistive Robot

	a_{i-1}	α_{i-1}	d_i	θ_i
Joint 1	0	0	L_1	θ_1
Joint 2	0	$\frac{\pi}{2}$	0	$\theta_2 + \frac{\pi}{2}$
Joint 3	L_2	0	0	$\theta_3 - \frac{\pi}{2}$
Joint 4	0	$-\frac{\pi}{2}$	$L_3 + L_4$	θ_4
Joint 5	0	$\frac{\pi}{2}$	0	θ_5
Joint 6	0	$-\frac{\pi}{2}$	$L_5 + L_6$	θ_6

The homogeneous transformation matrix (HTM), derived from these DH parameters, provides a compact representation of the robot's configuration. The HTM, expressed as equation 5.1, illustrates the transformation from one joint frame to the next along the robot's kinematic chain.

$$
{}^{i-1}T_i = \begin{bmatrix} 1 & 0 & 0 & a_{i-i} \\ 0 & c\alpha_{i-i} & -s\alpha_{i-i} & 0 \\ 0 & s\alpha_{i-i} & c\alpha_{i-i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.1}
$$

Here, α_{i-1} is the link twist, a_{i-1} is the link length, θ_i is the joint variable of the revolute joint, and d_i is the link offset and/or the joint variable of the prismatic joint.

Fig. 5.1a indicates that the robot's base frame $\mathcal{O}$ is located at point O, on the mounting

plate of the robot. This well-defined framework lays the groundwork for the subsequent analysis and optimization.

5.1.1 Kinematics

Kinematics refers to the study of motion without considering the forces that cause the motion. For robotic manipulators, kinematics deals with the relationship between the joint parameters of the robot and the position and orientation of its end-effector. This subsection presents a comprehensive overview of the kinematics of the assistive robot, including the direct kinematics, inverse kinematics, and the Jacobian.

Direct kinematics

The pose of the end-effector, which comprises both position and orientation, is obtained from the HTM. The matrix can be expressed as given in equation 5.2::

$$ {}^{0}T_{6} = \begin{bmatrix} & & & \\ & {}^{0}R_{6} & & {}^{0}P_{6} \\ & & & \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.2} $$

Here, the column vector ${}^{0}P_{6}$ represents the position of the end-effector, while the rotation matrix ${}^{0}R_{6}$ describes the end-effector's orientation using Euler angles (Roll, Pitch, and Yaw). Each sub-matrix comprises the following components:

$$
{}^{0}R_{6} = \begin{bmatrix} c\theta_y c\theta_z & s\theta_y s\theta_x c\theta_z - c\theta_x s\theta_z & s\theta_x s\theta_z + c\theta_x s\theta_y c\theta_z \\ c\theta_y s\theta_z & -s\theta_y s\theta_x s\theta_z + c\theta_x c\theta_z & -s\theta_x c\theta_z + c\theta_x s\theta_y s\theta_z \\ -s\theta_y & c\theta_y s\theta_x & c\theta_y c\theta_x \end{bmatrix}
$$

$$
{}^{0}P_{6} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}
$$

In this context, θ_x denotes the roll angle, θ_y refers to the pitch angle, and θ_z indicates the yaw angle. Furthermore, c and s represent cos and sin, respectively.

In order to compute the HTM that represents the pose of the end-effector, the successive multiplication of the HTM for each joint was performed, as shown below:

$$
{}^{0}T_{1}\ {}^{1}T_{2}\ {}^{2}T_{3}\ {}^{3}T_{4}\ {}^{4}T_{5}\ {}^{5}T_{6} =\ {}^{0}T_{6} \tag{5.3}
$$

In conclusion, the matrix ${}^{0}T_{6}$ represents the direct kinematics of the end-effector, thus providing a detailed overview of its position and orientation.

Inverse kinematics

To compute the inverse kinematics, this work employs the HUNTER algorithm, a novel approach that was presented earlier in Chapter 3.2. This algorithm is a modified version of the well-known Broyden-Fletcher-Goldfarb-Shanno (BFGS) method. The enhancement brought by HUNTER lies in its capability to handle situations where the BFGS method might fail due to a non-positive definite Hessian matrix.

One of the key advantages of the HUNTER algorithm is its ability to compute inverse

kinematics even for positions outside the workspace. This feature makes it an excellent tool for this work, where the robot's maneuverability and flexibility are crucial.

Jacobian matrix computation

The Jacobian matrix plays a pivotal role in robot kinematics as it provides a relationship between the joint velocities and the linear and angular velocities of the robot's end-effector. In other words, it allows us to understand how changes in the joint angles will impact the position and orientation of the end-effector. This makes it an essential tool for controlling the motion of the robot.

The Jacobian matrix for the assistive robot is computed using the methodology discussed in Chapter 4 and applying equation (4.27).

After computing the Jacobian matrix, there is a more profound understanding of the kinematic behavior of the assistive robot. This essential kinematic insight proves critical for the calculation of the novel manipulability index introduced in Chapter 4, which, in turn, becomes indispensable for optimizing the robot's functionality.

5.1.2 Folding Criteria

This folding criteria for the assistive robot takes into account the round shape of the robot's links. The folding requirements, as detailed in Chapter 2.2.2, are guided by the dimensions of the robot as depicted in Figure 5.2. Here, R_1 denotes the radius of link 1, R_2 refers to the radius of link 2, and R_3 stands for the combined radius of link 3 and link 4, as these links align with each other when the robot folds.

The parameter h, representing the height, can be expressed as the sum of the radii of the

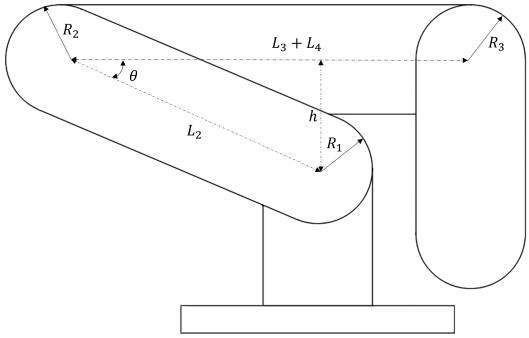


Figure 5.2: Schematic Representation of the Assistive Robot in Folded State with Dimensional Parameters

first two links:

$$h = R_1 + R_2 \tag{5.4}$$

The angle θ is then computed based on the sine rule involving the height h and the length L_2 of the second link:

$$\theta = \arcsin \frac{h}{L_2} \tag{5.5}$$

The computed distance h and the angle θ provide the position of link 3 and link 4 resting over link 1 when the robot is folded.

Finally, to ensure that link 5 and link 6 do not collide with link 1 during the folding process, the following relation should be satisfied:

$$L_2 \cos\theta + R_1 + R_3 * 2 < L_3 + L_4 \tag{5.6}$$

This condition ensures safe and efficient folding of the assistive robot without any risk of collision.

5.1.3 Dynamic model

In the study of robotic manipulators, the dynamic model holds a crucial role. It elucidates the relationship between the external forces and torques acting on the robot and the corresponding actuator torques. The dynamic model for the proposed assistive robot is derived by invoking the principles of virtual work and D'Alembert [108, 109].

D'Alembert's principle provides a robust framework to understand the dynamics, considering the inertia forces acting upon a body as if they were external forces. This perspective facilitates the analysis, under the assumption of the robot being in a quasi-static state. Consequently, the virtual work done by these external forces amounts to zero. Implementing this methodology yields the following representation of the dynamic model:

$$\boldsymbol{\tau} = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) \tag{5.7}$$

Here, the vector $\tau \in \mathbb{R}^{6\times1}$ signifies the torque of each actuator. The matrix $M \in \mathbb{R}^{6\times6}$ encapsulates the mass and inertia of each link. The matrix $C \in \mathbb{R}^{6\times6}$ illustrates the Coriolis and centrifugal forces, and $G \in \mathbb{R}^6$ symbolizes the gravity vector.

To utilize the virtual work method with efficiency, calculating the Jacobian matrix of each link is essential. For this purpose, considering a link j, whose pose depends on k joints, the Homogeneous Transformation Matrix (HTM) that describes the center of gravity of link j is expressed as follows:

$$ {}^{0}T_{1}\ {}^{1}T_{2} \dots\ {}^{k-1}T_{k}\ {}^{k}T_{j_m} = {}^{0}T_{j_m} \tag{5.8} $$

In this context, ${}^{k}T_{j_m}$ is an HTM that explains the pose of each link j's center of gravity with respect to the coordinate system k. Accordingly, for each link, the previously detailed methodology is employed on the HTM from Eq. 5.8, yielding J_j.

J_j is understood as the Jacobian matrix at the center of gravity for each corresponding link.

The focus of the discussion now transitions to the dynamic model of the proposed assistive robot, incorporating results from Chapters (4.2.3) and (5.1.3). In an effort to simplify the analysis, the inertial effects are disregarded. This decision is motivated by the impracticality of considering all potential robot motions at every point within the workspace, a consideration that would be necessary for an optimization problem. Consequently, matrices M and C of Eq. 5.7 are omitted, leading to the following equation:

$$ \tau = G(\theta) \tag{5.9} $$

Eq. 5.9 illustrates the torque in the context of the robot's static configuration. Thus, the matrix G is calculated, identifying the Jacobian matrices for each coordinate frame attached to the center of gravity of link i, and distinguishing the wrenches impacting each link.

To simplify the analysis, several assumptions are made. It is considered that the actuator at joint 1 is located at the base of link 2, thereby excluding link 1. Furthermore, given that the actuator at joint 4 does not shift the center of gravity of link 4 relative to link 3, links 3 and 4 are regarded as a single link. Similarly, since the actuator at joint 6 doesn't alter the center of gravity of link 6 relative to link 5, links 5 and 6 are treated as a single link. Assuming

uniform density across links, the center of gravity is placed in the middle of each link. Fig. 5.3 illustrates the free-body diagram of each link.

Finally, due to these assumptions reducing the number of links, the resulting links are referred to as 'simplified links'. The subsequent section provides details about each link, such as the HTM and the vector of external forces or wrench ($\mathcal{W}$), that are necessary for computing the dynamic model.

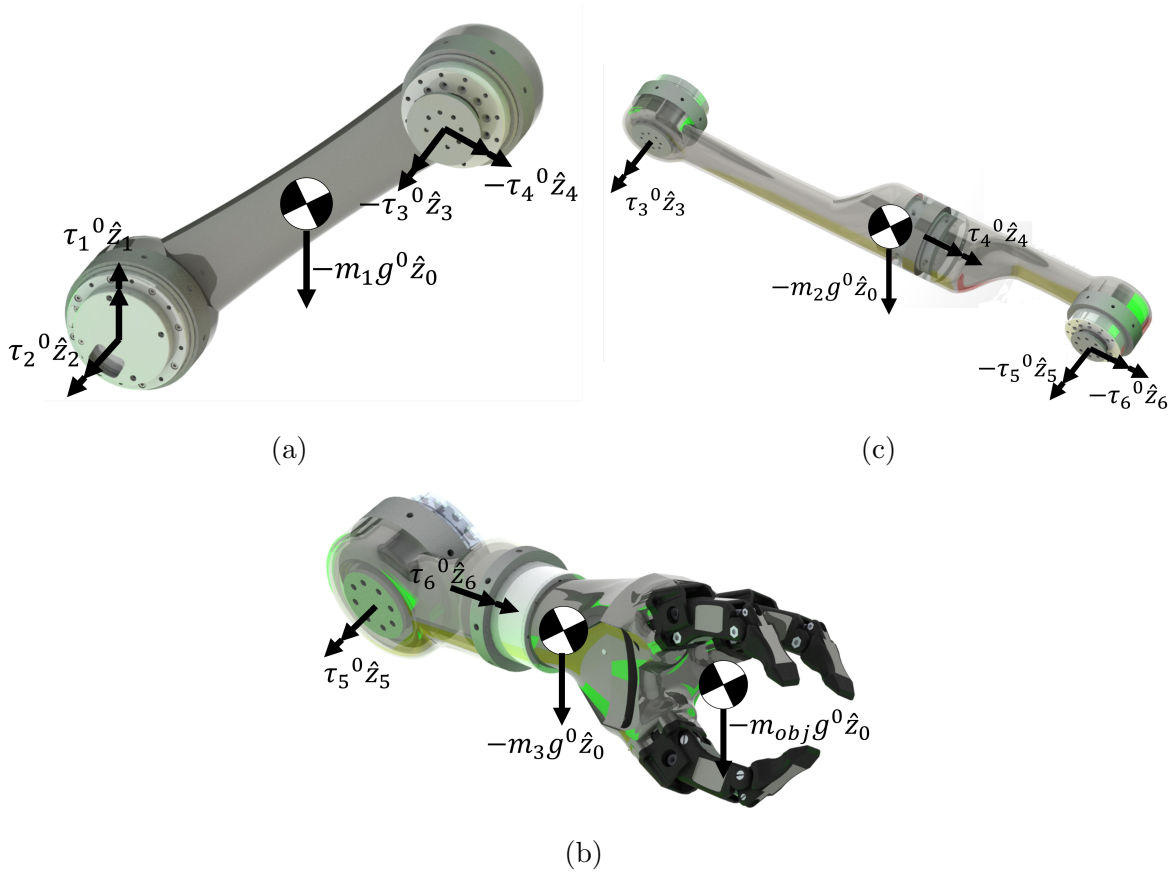


Figure 5.3: Free body diagrams of the simplified links (a) Link 1 (b) Link 2 (c) Link 3.

Simplified link 1

Fig. 5.3a presents the free body diagram (FBD) of the simplified link 1 . The HTM used to represent the pose of the center of gravity of this link is presented below:

$$ {}^{0}T_{1_2} = {}^{0}T_1 \, {}^{1}T_2 \, {}^{2}T_{1_2} \tag{5.10} $$

The wrench ($\mathcal{W}$) of the simplified link 1 is:

$$ \mathcal{W}_1 = \begin{bmatrix} -m_1 g \, {}^{0}\hat{z}_0 \\ \tau_1 \, {}^{0}\hat{z}_1 + \tau_2 \, {}^{0}\hat{z}_2 - \tau_3 \, {}^{0}\hat{z}_3 - \tau_4 \, {}^{0}\hat{z}_4 \end{bmatrix} \tag{5.11} $$

Where g is the gravity and m_1 is the mass of simplified link 1, given by the following equation:

$$ m_1 = \rho_l L_2 \tag{5.12} $$

where ρ_l is the linear density of the robot.

Simplified link 2

Analogously, Fig. 5.3b presents the FBD of the simplified link 2. The HTM used to represent the pose of the center of gravity of this link is presented below:

$$ {}^{0}T_{2_4} = {}^{0}T_1 \, {}^{1}T_2 \, {}^{2}T_3 \, {}^{3}T_4 \, {}^{4}T_{2_4} \tag{5.13} $$

The wrench of the simplified link 2 is presented as follows:

$$\mathcal{W}_2 = \begin{bmatrix} -m_2 g \ {}^0\hat{z}_0 \\ \tau_3 \ {}^0\hat{z}_3 + \tau_4 \ {}^0\hat{z}_4 - \tau_5 \ {}^0\hat{z}_5 - \tau_6 \ {}^0\hat{z}_6 \end{bmatrix} \tag{5.14}$$

The mass m_2 is given by the following equation:

$$m_2 = \rho_l(L_3 + L_4) \tag{5.15}$$

Simplified link 3

Analogously, Fig. 5.3c presents the FBD of the simplified link 3. The HTM used to represent the pose of the center of gravity of this link is presented below:

$${}^0T_{3_6} = \ {}^0T_1 \ {}^1T_2 \ {}^2T_3 \ {}^3T_4 \ {}^4T_5 \ {}^5T_6 \ {}^6T_{3_6} \tag{5.16}$$

The wrench of the simplified link 3 is presented below:

$$\mathcal{W}_3 = \begin{bmatrix} -m_3 g \ {}^0\hat{z}_0 \\ \tau_5 \ {}^0\hat{z}_5 + \tau_6 \ {}^0\hat{z}_6 \end{bmatrix} \tag{5.17}$$

The mass m_3 is given by the following equation:

$$m_3 = \rho_l(L_5 + L_6) \tag{5.18}$$

By applying the principle of virtual work on the wrenches of the three simplified links, the

following equation is obtained:

$$J_1^T \mathcal{W}_1 + J_2^T \mathcal{W}_2 + J_3^T \mathcal{W}_3 = 0 \tag{5.19}$$

This equation, (5.19), represents a linear system for τ. Solving this system leads to the derivation of the dynamic model as shown below:

$$\begin{aligned} \tau &= \begin{bmatrix} 0 \\ \kappa_6 \\ \kappa_7 \\ \kappa_8 \\ \kappa_9 \\ 0 \end{bmatrix} \\ &= \hat{G} \end{aligned} \tag{5.20}$$

The κ_1 to κ_4 terms are detailed in Appendix (7.5). Equation (5.20) exemplifies the gravitational influence on each joint attributed to the mass of each link.

To incorporate the impacts of a payload (M_{obj}) or a moving object at the end-effector of the robot, the Jacobian matrix at the end-effector (J) is computed and the virtual work done by the payload is added. This provides an updated dynamic model that includes the effects of the payload at the end-effector:

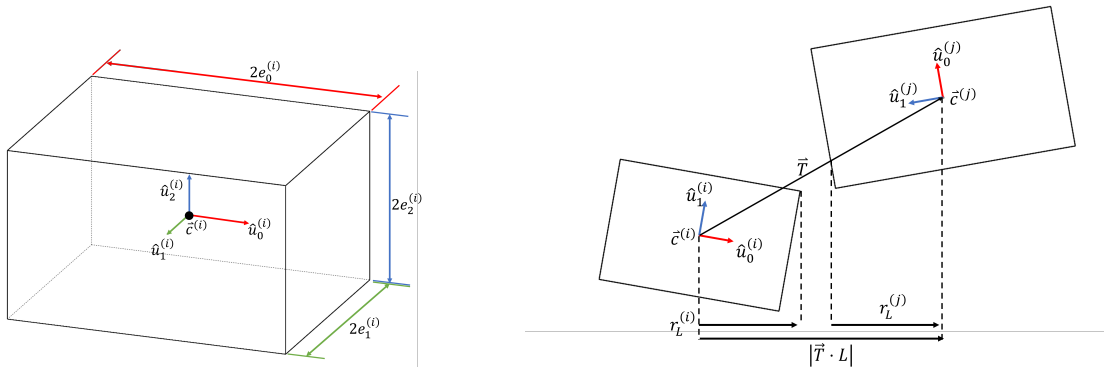
$$\tau = \hat{G} + J^T \mathcal{W}_{ef} \tag{5.21}$$

Here, $\mathcal{W}_{ef}$ denotes the wrench generated by the moving object at the end-effector. $\mathcal{W}_{ef}$ is defined as:

$$\mathcal{W}_{ef} = \begin{bmatrix} 0 \\ 0 \\ M_{obj}g \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.22}$$

Equation (5.21) represents the static model of the assistive robot, encompassing both the gravitational and payload effects.

5.2 Algorithm for Collision Detection



(a) (b)

Figure 5.4: (a) OBB representation, (b) The distance between two OBBs is less than the total of their projected radii for some axis L

The purpose of a collision detection algorithm in robotic systems is to identify potential col-

lisions between different links of the robotic manipulator, as well as between the robot and other objects in its operational environment. This is a vital aspect of robot safety and effectiveness, particularly for robots interacting closely with humans or in cluttered environments.

In this study, the problem is approached by representing each link of the robot as a combination of Oriented Bounding Box (OBB) objects. OBB is a compact and flexible geometric representation that is widely used in collision detection for its efficiency. The idea is to enclose each link within an oriented bounding box and then inspect for potential collisions between these boxes.

The detection of collision between two OBB objects is accomplished via a collision algorithm as elaborated in [110]. The OBB objects are oriented bounding boxes encapsulating the physical bodies in the 3D space of the robot's environment, as demonstrated in Figure 5.4(a).

The OBB object i has its center represented by vector $\vec{c}^{(i)}$, and three half-dimensions denoted as $e_k^{(i)}$ along their respective directions ${u_k}^i$. The rotation of the OBB is defined by the three orthonormal vectors $u_k^{(i)}$ that compose the following rotation matrix:

$$R^{(i)} = \begin{bmatrix} \hat{u}_0^{(i)} & \hat{u}_1^{(i)} & \hat{u}_2^{(i)} \end{bmatrix} \tag{5.23}$$

Figure 5.4(b) illustrates the distance vector between the centers of two OBB objects i and j:

$$\vec{T} = \vec{c}^{\,(j)} - \vec{c}^{\,(i)} \tag{5.24}$$

This vector $\vec{T}$ is projected onto 15 separate axes. These axes include the three coordinate axes of both OBB i and j and the nine axes perpendicular to an axis from each OBB. The vectors forming the separating axes are from $R^{(i)}$ and $R^{(j)}$ as defined in equation (5.23), along with the

cross products:

$$P_k = \left[\hat{u}_k^{(i)} \times \hat{u}_0^{(j)} \quad \hat{u}_k^{(i)} \times \hat{u}_1^{(j)} \quad \hat{u}_k^{(i)} \times \hat{u}_2^{(j)} \right] \text{ for } k = 0, 1, 2 \tag{5.25}$$

The projection of $\vec{T}$ onto each separating axis is therefore:

$$T_L = \begin{bmatrix} R^{(i)^t} \\ R^{(j)^t} \\ P_0^t \\ P_1^t \\ P_2^t \end{bmatrix} \vec{T} \tag{5.26}$$

Where the t superscript represents the transpose. Additionally, the projection of each OBB edge onto each separating axis is given by:

$$r_L^{(i)} = \begin{bmatrix} I \\ |R^{(j)^t} R^{(i)}| \\ |P_0^t R^{(i)}| \\ |P_1^t R^{(i)}| \\ |P_2^t R^{(i)}| \end{bmatrix} \begin{bmatrix} e_0^{(i)} \\ e_1^{(i)} \\ e_2^{(i)} \end{bmatrix} \tag{5.27}$$

where I represents a 3×3 identity matrix. In order to verify the separation between the two objects, the following condition is checked:

$$|T_L| > r^{(i)} + r^{(j)} \tag{5.28}$$

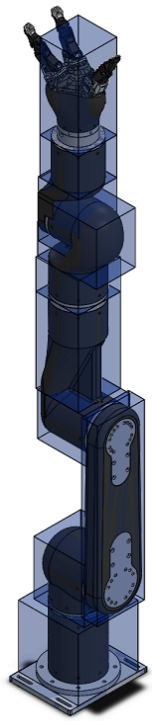


Figure 5.5: Representation of the robot with OBBs. Each link of the robot is encapsulated by a box that dynamically adjusts with the changes in the robot's configuration.

This condition is evaluated for each element in the 15×1 vector T_L. If at least one of the inequalities in equation (5.28) holds true, the two OBB objects are determined to be separate, indicating the absence of collision. On the contrary, if none of the inequalities are satisfied, it signifies a collision between the OBBs.

Table 5.2: Definition of each Oriented Bounding Box (OBB) as a function of its corresponding link length. Each box definition includes the HTM representing the box's orientation and position relative to the link's coordinate frame, as well as the box's dimensions

$OBBi$	${}^0T_i$	${}^i\vec{c}^{(i)\ T}$	$2 * \vec{e}^{(i)\ T}$
OBB 1	${}^0T_0$	$\begin{bmatrix}0 & 0 & 5\end{bmatrix}$	$\begin{bmatrix}141.4 & 180 & 10\end{bmatrix}$
OBB 2	${}^0T_0$	$\begin{bmatrix}0 & 0 & \frac{L_1-88}{2}+10\end{bmatrix}$	$\begin{bmatrix}140 & 140 & L_1-88\end{bmatrix}$
OBB 3	${}^0T_1$	$\begin{bmatrix}0 & 9.93 & -8.75\end{bmatrix}$	$\begin{bmatrix}110 & 118.15 & 127.5\end{bmatrix}$
OBB 4	${}^0T_2$	$\begin{bmatrix}\frac{L_2-5}{2} & 0 & 81.3\end{bmatrix}$	$\begin{bmatrix}L_2+105 & 110 & 37\end{bmatrix}$
OBB 5	${}^0T_3$	$\begin{bmatrix}0 & 0 & -12.75\end{bmatrix}$	$\begin{bmatrix}100 & 100 & 116.5\end{bmatrix}$
OBB 6	${}^0T_3$	$\begin{bmatrix}0 & \frac{L_3+50}{2} & 0\end{bmatrix}$	$\begin{bmatrix}99 & L_3-50 & 99\end{bmatrix}$
OBB 7	${}^0T_4$	$\begin{bmatrix}0 & 0 & -76.5-\frac{L_4-76}{2}\end{bmatrix}$	$\begin{bmatrix}99 & 99.3 & L_4-75\end{bmatrix}$
OBB 8	${}^0T_4$	$\begin{bmatrix}0 & 19.27 & -15.25\end{bmatrix}$	$\begin{bmatrix}91 & 137.83 & 121.5\end{bmatrix}$
OBB 9	${}^0T_5$	$\begin{bmatrix}-0.65 & -20 & -1.92\end{bmatrix}$	$\begin{bmatrix}90 & 131.3 & 94.15\end{bmatrix}$
OBB 10	${}^0T_5$	$\begin{bmatrix}-0.15 & -85.65-\frac{L_5-85.65}{2} & 0\end{bmatrix}$	$\begin{bmatrix}90 & L_5-85.65 & 90.3\end{bmatrix}$
OBB 11	${}^0T_6$	$\begin{bmatrix}-0.15 & 0 & -\frac{L_6}{2}\end{bmatrix}$	$\begin{bmatrix}90 & 90.13 & L_6\end{bmatrix}$

The implementation of the OBB collision detection algorithm requires the definition of the boxes for each link to be functionally related to the length of the link. This is crucial because the optimization algorithm will iteratively adjust the link lengths, and for each length, the collision detection algorithm must function correctly.

As depicted in Figure (5.5), the robot is represented with associated bounding boxes. Additionally, Table (5.2) provides the specific definition of each box as a function of the corresponding link length. Importantly, the boxes are defined in relation to the coordinate frame of each link, i.e., the HTMs are employed to define the orientation and position of each box. This approach ensures that the boxes remain correctly aligned with their corresponding links during all robot movements and manipulations, enabling accurate collision detection.

Algorithm 2 Collision checker

```
1: procedure CollisionChecker
2:     for i = 1; i ≤ n; i++ do
3:         for j = i + 1; j ≤ n; j++ do
4:             if Any box of Link i collides with Any box of Link j then
5:                 return True
6:             end if
7:         end for
8:     end for
9:     return False
10: end procedure
```

Algorithm 2 outlines the implementation of the collision checker. The algorithm iteratively checks each box of each link for potential collisions with the boxes of every other link. If any collision is detected, the algorithm returns true; otherwise, it returns false, indicating that no collisions exist in the current configuration.

5.3 Scatter Search: The Optimal Choice for Robot Optimizatio

Scatter Search is a metaheuristic algorithm recognized for its effectiveness in solving multi-objective optimization problems. The algorithm's strength lies in its unique methodology, which generates a diverse set of solutions and combines them to form new potential solutions. This approach, underpinned by a strategic combination of intensification (focusing on promising areas of the search space) and diversification (exploring a wide range of solution possibilities), enables the algorithm to converge to the Pareto front of the problem efficiently [111].

The versatility and demonstrated efficacy of scatter search and path relinking make them

well suited to handle a wide variety of optimization problems. Their applications encompass a broad range of sectors and complex structures, as discussed in the literature [111]. Computational results have consistently demonstrated the high-quality solutions produced by scatter search, all within a reasonable computational time frame, positioning it as one of the top-performing metaheuristics [112].

Further validation of scatter search's performance comes from comprehensive computational comparisons. In studies involving 16 leading methods for multimodal function optimization, tested on a set of 64 widely used benchmark problems, a new Scatter Tabu Search (STS) procedure emerged as a highly competitive algorithm. In terms of the average optimality gap achieved, STS compared favorably with other state-of-the-art methods, affirming its potential in tackling complex optimization problems [113].

The superiority of Scatter Search is supported by extensive experiments and associated statistical tests. In these evaluations, the most effective variant of Scatter Search consistently outperformed other state-of-the-art methods. This strong performance demonstrates the remarkable adaptability of Scatter Search, with its modifications and variations holding significant potential for enhanced problem-solving capabilities [114].

Given these attributes, scatter search was chosen as the optimal algorithm to tackle the multi-objective optimization problem encountered in this study – the optimization of the assistive robot. The ensuing sections will detail the application of scatter search in this specific context, shedding light on the nuances of the optimization problem and the benefits brought forth by this powerful algorithm.

Table 5.3: Optimization Parameters for Scatter Search: This table lists the parameters, their ranges, and constraints used in the scatter search optimization process for determining the optimal link lengths of the assistive robot.

Required constants	Values
L_1 limits	$[234,\ 500.0]\ mm$
L_2 limits	$[214.0,\ 500.0]\ mm$
L_3 limits	$[240.0,\ 500.0]\ mm$
L_4 limits	$[102.0,\ 500.0]\ mm$
L_5 limits	$[146.0,\ 500.0]\ mm$
L_6 limits	$[199.0,\ 500.0]\ mm$
L_{max}	$1260.4\ mm$
m_{obj}	$3\ kg$
ρ_l	$1.0116\ kg/m$
N	3360
R_1	$55.15\ mm$
R_2	$50\ mm$
R_3	$45\ mm$

5.3.1 Scatter Search for Assistive Robot Link Length Optimization

This section delves into the application of scatter search for the optimization of our proposed assistive robot. The objective function slated for optimization was introduced in Chapter 4.4. This methodology takes advantage of the comprehensive description of the assistive robot outlined in this chapter, allowing us to compute an objective function that harmoniously combines workspace maximization and energy consumption reduction.

Constants crucial for the execution of this algorithm, along with the constraints governing optimization, are presented in Table 5.3. It is crucial to remember that the aim of the optimization is to find the link lengths that result in the minimization of the objective function. The lower bounds depicted in the table for each link were strategically determined based on the minimum space required for motor assembly. These motors were also pivotal in defining

Figure 5.6: Optimized Assistive Robot: This figure illustrates the final design of the assistive robot with link lengths derived from the optimization process. The optimized design aims for maximum workspace while minimizing energy consumption.

the structure of each link.

The outcomes of the optimization process are detailed in Table 5.4, and the acquired dimensions have been employed to construct the robot, as illustrated in Figure 5.6. The robot features Brushless DC motors with harmonic drives, underpinned by a lightweight structure crafted from carbon fiber and aluminum. This robot, which weighs approximately 13 kg, has joint limits as specified in Table 5.5, demonstrating its operational range and versatility. Its gripper operates via a compliant mechanism controlling three fingers, providing precise manipulation capabilities. Communication with the robot is facilitated through either CAN or

Table 5.4: Results of the optimization

Variable	Optimum value (mm)	$\sum L_i$ (mm)	$\overline{QAT_{global}}$ $((Nm)^2)$
L_1	267.76	1250.5	1.589×10^6
L_2	245.46		
L_3	259.12		
L_4	130.64		
L_5	148.23		
L_6	199.34		

ETHERCAT protocols, ensuring versatile and robust control. Following this chapter, we delve into the validation of the optimization process, which substantiates the effectiveness and applicability of our optimized design. The next stage will examine the robot's performance and reliability under a variety of operating conditions.

Table 5.5: Joint limits of the purposed assistive robot

Joint	Limits (units in degree)
Joint 1	$[-180,\ 180]$
Joint 2	$[-145,\ 145]$
Joint 3	$[-155,\ 155]$
Joint 4	$[-180,\ 180]$
Joint 5	$[-100,\ 100]$
Joint 6	$[-180,\ 180]$

5.4 Validation through Simulation

This section aims to verify the optimization results by employing a thorough simulation process. The objective of this validation is to affirm that the optimal robot adheres to all the key design constraints that have been outlined in this study.

Primarily, the simulation is designed to validate that the robot can effectively traverse across all seven designated workspaces associated with the 19 ADL. This is crucial in ensuring that the robot is capable of efficiently performing the tasks within the defined areas, affirming the accuracy of the optimized link lengths.

Additionally, it is important that the robot can compactly fold when not in use. This functionality not only allows for easier storage but also promotes seamless integration within a living environment without causing unnecessary obstructions. The simulation process has been instrumental in validating this particular design aspect.

Lastly, the simulation verifies whether the robot can support a payload of 3 kg. This weight threshold is indicative of the demands the robot will typically encounter while assisting with ADLs. The simulation will therefore confirm this capacity, ensuring that the robot can fulfill its intended roles without risk of mechanical failure or instability.

The meticulous nature of this validation process allows us to assert the effectiveness of our optimization procedure and the capabilities of the resulting robot design. The subsequent sections will detail the findings from these simulations.

5.4.1 Workspace Coverage Verification

Figure (5.7) illustrates the extent of the workspace covered by the optimally designed assistive robot. This representation clearly indicates the portions of the seven designated workspaces the robot can efficiently access, while also demonstrating the comprehensive coverage the robot provides.

The comparison between the accessible workspace and the total designated workspaces is also displayed in the figure. Importantly, the robot is capable of reaching more than 90%

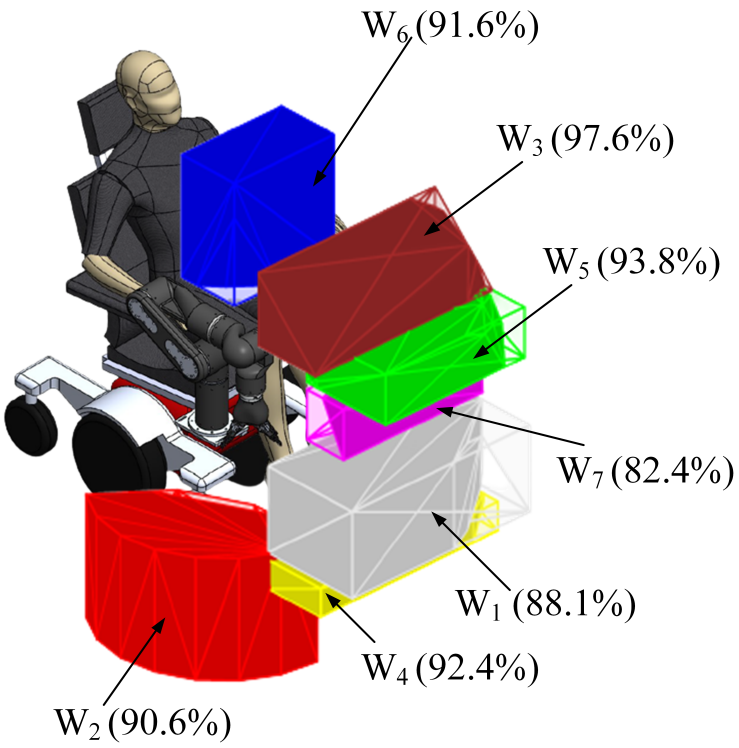


Figure 5.7: Comparison of the Accessible Workspace Covered by the Optimal Assistive Robot with the Total Designated Workspaces

of the total workspace, demonstrating the effectiveness of the optimization process in ensuring substantial workspace coverage. This expansive reach plays a crucial role in ensuring the robot's efficacy in assisting with the ADLs across varied locations within the home environment.

5.4.2 Foldability Verification

Figure (5.8) showcases the CAD model of the optimally designed assistive robot when in its fully folded state. This visual evidence further validates the optimization process, as it confirms

Figure 5.8: CAD Model Illustrating the Folded State of the Optimal Assistive Robot

the robot's capability to compactly fold for efficient storage or transportation purposes. This foldability feature is a crucial requirement for ensuring the robot's practical use in varied and sometimes constrained domestic environments.

5.5 Dynamic Validation

This section serves to validate the dynamic capabilities of the optimized assistive robot, particularly assessing its ability to hold a weight of 3kg at various positions within its workspace. This validation involves the Simscape Multibody model of the robot (Figure 5.9), utilized to

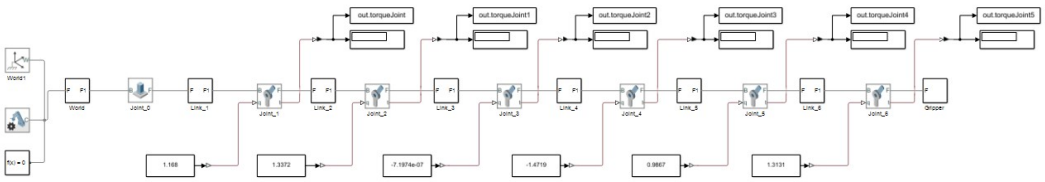


Figure 5.9: Simscape Multibody model of the optimal assistive robot

Table 5.6: Motor References for Each Joint of the Robot

Joint	Motor Reference	Rated Torque (Nm)
1	Qrob80	31
2	Qrob90	52
3	Qrob80	31
4	Qrob70	10
5	Qrob70	10
6	Qrob70h	5.4

simulate the torque requirements at each position within the workspace. The model takes into account the inertia and masses of each robot component as detailed in the unified robotics description format (URDF) file presented in Appendix 7.5.

Subsequently, the computed torque at each joint was evaluated against the rated torque of the corresponding motor, as detailed in Table 5.6. If the required torque exceeded the motor's rated torque, the corresponding workspace position was deemed unacceptable and discarded.

The remaining points, for which the required torque was within the motor's rated capacity, form the 'torque workspace' of the robot. Figure 5.10 illustrates this torque workspace, which encompasses 75% of the total workspace under consideration, validating the robot's capability

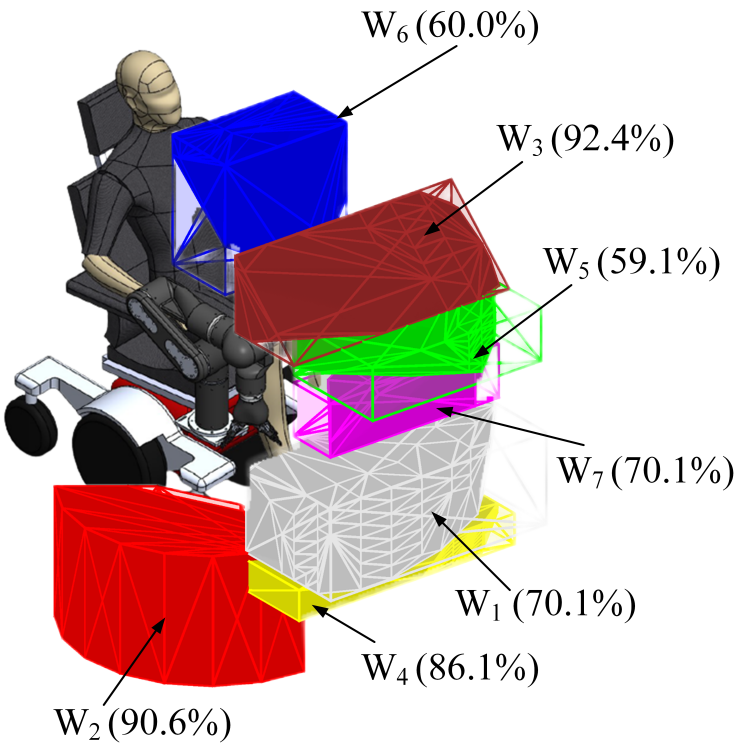


Figure 5.10: Torque workspace coverage of the optimal assistive robot

to hold a 3kg load.

Given that the 75% torque workspace coverage is concentrated around the mid-range of the workspace, it implies that the robot is well optimized for tasks that are most needed. The mid-range of the workspace is often the most utilized, aligning with typical human ergonomics and common task locations. For assistive robotics, this can cover many ADLs like eating, dressing, and personal hygiene, which typically occur within arm's reach.

Therefore, despite the apparent limitation, this result is promising as it suggests the robot is capable of handling the specified load where it's most frequently needed. However, considering

specific requirements and constraints, there may be room for further optimization. If some crucial tasks fall into the 25% of the workspace where the robot's performance is limited, design modifications might be necessary. In summary, while the 75% torque workspace coverage is a positive result, the specific needs and constraints of the application will ultimately determine its adequacy.

5.6 conclusions

This chapter is devoted to the practical application of the integrated methodology outlined in the previous section, employing it for the design of an optimal assistive robot. The methodology 4, which incorporates both manipulability and energy consumption indices, provides a structured framework for the design process.

The chapter begins with an overview of the parameters and specifications informing the robot design. Details on the Denavit-Hartenberg representation, kinematics, folding criteria, dynamic model, and collision avoidance algorithm set a solid foundation for a robust and informed design process.

Following this groundwork, the design phase ensues, which includes the formulation of an objective function subjected to optimization. The optimization phase leverages the Scatter Search algorithm, a well-established metaheuristic, guiding towards an optimal assistive robot design.

A thorough validation process ensures the efficacy of the proposed design. Through extensive simulation tests, it's confirmed that the robot can cover the necessary workspace, fold into a compact form, and hold a 3kg weight midrange within the workspace. This comprehen-

sive validation process underscores that the robot design effectively meets the efficiency and reliability benchmarks crucial for an assistive device.

In summary, this chapter presents a comprehensive walkthrough of an integrated methodology application for designing an optimal assistive robot. The Scatter Search-based optimization, coupled with an exhaustive validation process, results in a design that satisfies stringent efficiency and reliability criteria. The processes and techniques detailed here serve as valuable references for future works aimed at designing assistive robots or similar devices.

CHAPTER 6

DEVELOPMENT OF THE SOEM-BASED LIBRARY FOR ASSISTIVE ROBOT MANIPULATION

Assistive robotics has emerged as a pivotal field, aspiring to enrich the quality of life for individuals, especially those grappling with physical impairments. The orchestration of seamless, safe, and efficacious interaction between the robot and its user is a complex challenge that calls for sophisticated control strategies and robust communication interfaces. In particular, the real-time control and communication capabilities play a vital role in assistive robots to ensure prompt response, high precision, and user safety.

One of the reasons for the importance of real-time capabilities in assistive robotics lies in the delicate and dynamic nature of human-robot interaction. Any delays or inaccuracies in response can lead to inefficiencies, failures, or even safety hazards. The ability to respond in real time allows the assistive robot to react promptly to user commands or changes in the environment, resulting in smooth and reliable operation [115].

In order to accommodate the demands of real-time control and communication, the Robot Operating System 2 (ROS2) is utilized to control the proposed assistive robot [116]. ROS2, an advanced iteration of the original ROS, integrates the Data Distribution Service (DDS) as its transport system, leading to significant improvements in the real-time performance of autonomous robotic systems [117]. The capabilities of ROS2 for real-time applications have been shown to attain a latency of up to 500 microseconds [118], thereby aligning with the requirements of assistive robotic systems.

The ROS2 Library, a suite of software packages and tools, offers a structured framework for building robotic applications [119]. It houses an extensive range of capabilities, including libraries for communication, visualization, perception, and control. Given its comprehensive offerings and wide-spread usage in the robotics community, ROS2 emerges as a natural choice for our assistive robot control system [120]. ROS2 has been effectively implemented in a wide range of robotic systems, from the WALKING robot and ARI, to the Kinova Robotics JACO and MICO robotic manipulator arms [121]. Further, its implementation spans across diverse applications, such as mobile robots for Simultaneous Localization And Mapping (SLAM), odometry, navigation, and assistive robotic manipulators aiding in ADL [122, 123]. This proven versatility and effectiveness of ROS2 make it an ideal platform for controlling our assistive robot.

The capabilities of ROS2 can be significantly enhanced through the implementation of EtherCAT, forming the core of our control system for assistive robots. This strategic amalgamation greatly amplifies the real-time control and communication capacity. EtherCAT, a respected industrial field-bus protocol known for its high-speed data transfer rates reaching up to 100 Mbps, stands as a vital component in this system. It ensures rapid response times and minimizes cycle durations, essential for the real-time communication necessary in both position

and force-based robot control [124].

The EtherCAT Technology Group upholds open governance, thus ensuring that EtherCAT technology will consistently remain an open standard and be freely accessible [125]. Existing within the EtherCAT ecosystem is a substantial network of electronics, software, and solution providers. This facilitates the availability of a wide assortment of affordable hardware and software, keeping costs manageable while encouraging collaboration between organizations.

Given the synergistic potential of EtherCAT and ROS2, it became evident that an integration solution was necessary to harness their combined strengths effectively. Thus, this chapter delves into the development of a Simple Open EtherCAT Master (SOEM) based library that forms a seamless bridge between EtherCAT and ROS2. This library will empower us to control the proposed assistive robot in real-time, signifying a crucial step towards our assistive robotics goal.

The chapter is structured as follows: the first section provides a brief review of existing SOEM-based libraries integrated with ROS2, illuminating the current landscape of tools available. The second section describes the development process for our novel library, detailing the steps taken in its creation. Subsequently, the third section articulates the design constraints guiding the library's development, ensuring its alignment with crucial requirements. Finally, the fourth section presents the architecture of the library, offering an overview of its structure and functionality.

6.1 Review of Existing SOEM-based Libraries

The field of robotics acknowledges the profound influence of real-time and reliable communication on the efficacy and functionality of robotic systems. The Simple Open EtherCAT Master (SOEM) is one such powerful tool that facilitates these capabilities and has been extensively adopted in the industry [126, 127]. The SOEM's allure stems from its capacity to govern and oversee EtherCAT-enabled devices in real-time - an indispensable requirement in robotics where prompt, accurate control is paramount [128]. Additionally, the open-source nature of SOEM has spurred numerous researchers to adapt and extend it into SOEM-based libraries for tailored applications [129].

Reflecting the significance of EtherCAT communication in ROS, multiple SOEM-based libraries have been curated for diverse robotic applications, finding their niche in real-time robotic control and industrial automation. For instance, the design of a novel servo motor controller (SMC) founded on EtherCAT is documented in [130], striving to realize versatile networked multi-axis motion control with cost-effectiveness and superior performance. Similarly, other works outline the design of control systems using EtherCAT communication for different applications, such as the real-time control system for a 2-degree-of-freedom fluid-actuated manipulator [131], the servo control system for the automatic tuning system of a short-wave transmitter [132], and the reliable and high-performance control system for humanoid robots [133].

However, many of these libraries, while specialized for certain applications, often lack comprehensive safety features, which are crucial for assistive robots involved in close human interactions. The challenge amplifies when integrating these libraries with robust and new-age robotic frameworks, such as the ROS2. ROS2, with its enhanced communication infrastructure and support for safety-critical applications, has seen limited attention in integration with

SOEM-based libraries. This gap curtails the broader applicability and future scalability of these libraries in the fast-paced arena of robotics applications.

6.2 Overview of the SOEM-based library development

The pursuit of developing a SOEM-based library targets the construction of a comprehensive and streamlined library aimed at facilitating assistive robot manipulation, with a keen focus on controlling EtherCAT motors and seamlessly integrating with ROS2. The primary ambition of the library is to offer an effective and harmonious modus operandi to govern assistive robots designed to assist with ADL. The developmental stages of the library are as follows:

1. **Requirement Identification:** Defining the crucial requirements and specifications for the library, such as real-time performance, a broad array of safety features, and compatibility with both ROS2 and EtherCAT, was the initial step.

2. **Library Architecture Design:** Subsequently, the architecture of the library is designed, taking care to include base libraries and their interactions for modularity and scalability.

3. **Inverse Kinematics Algorithm Integration:** The library also incorporates the innovative HUNTER inverse kinematics algorithm to bolster the robot's ability to conduct ADL. **Safety Feature Incorporation:** Safety measures such as collision avoidance and other protective mechanisms are then embedded in the design to ensure user safety.

4. **Performance Testing and Validation:** The final stage encompasses testing and validating the library's performance, efficiency, and reliability across a multitude of scenarios and use cases.

The development of this SOEM-based library forms a pivotal component in achieving the overarching dissertation objectives, as it equips the assistive robot with the ability to execute the required tasks effectively, safely, and in real-time. The forthcoming sections delve into the integration with ROS2 for EtherCAT motor control and outline the objectives and scope of the library in further detail.

6.3 Requirement identification

The proposed library seeks to prioritize the safety of the user while maintaining the usability of the assistive robot. Thus, the proposed library is guided by the following design requirements:

- **Real-Time Performance:** In the field of assistive robotics, the capability to perform tasks in real time is a critical requirement. Therefore, the proposed library needs to provide real-time performance for the robot's control system with a latency of 1kHz (1ms). This is to ensure the accurate computation of HUNTER.

- **Safety Features:** As assistive robots operate in close proximity to human users, the library must incorporate an array of safety features to prevent any possible harm to the user. This includes a collision avoidance algorithm that takes into account both the user and the wheelchair, aiming to prevent collisions between the robot and the user.

- **Compatibility with ROS2 and EtherCAT:** The library should be compatible with both ROS2 and EtherCAT to ensure smooth and efficient operation of the assistive robot.

- **Ease of Use:** The library should be user-friendly and intuitive for both developers and users of the assistive robot.

- **Scalability:** As future advancements in assistive robotics may necessitate the addition of more complex functionalities, the library should be designed with scalability in mind.
- **Reliability:** The library must be reliable and robust, ensuring that the assistive robot can operate consistently over extended periods.
- **Flexibility:** The library should be flexible to adapt to different assistive robot models and their specific requirements.
- **Efficiency:** The library should efficiently manage resources to avoid unnecessary computation and maintain optimal performance.
- **Documentation:** Comprehensive and clear documentation should be available for developers to understand and effectively use the library.

6.4 Library Architecture

Figure 6.1 showcases the architecture of the proposed library and elucidates its integration with the ROS2 framework for efficient implementation. This schematic gives an insight into the library's structure and its synergistic interactions with ROS2 components, underscoring its harmonious operation within the larger system.

The library primarily breaks down into four core components:

- **Control Interface GUI:** The Control Interface GUI provides a graphical user interface (GUI) that is built in Python. This GUI allows for intuitive control of the robot's motions, offering a user-friendly way to interact with the robot's functionalities.

- **Robot Control Libraries:** The Robot Control Libraries are a set of libraries that manage the communication between the main control system and the EtherCAT slaves (the motors). This library is developed in C++, following object-oriented programming principles, to ensure high-speed and reliable communication.

- **Kinematics:** The Kinematics component is a set of libraries that implement the robot's kinematics, specifically the HUNTER model. To boost performance speed, this component is implemented in C++ using object-oriented programming (OOP).

- **Safety Library:** The Safety Library is tasked with detecting collisions between the robot and the user to ensure safe human-robot interactions. To maintain optimal performance and reliability, this library is also implemented in C++.

In the subsequent subsections, each of these key components is further discussed, elaborating on their individual functions, implementation details, and how they contribute to the overall operation of the proposed library.

6.4.1 Control interface GUI

Figure (6.2) showcases the Graphical User Interface (GUI) employed by the library for maneuvering the assistive robot. The GUI was built in Python due to its powerful yet straightforward syntax, the availability of multiple supporting libraries, and its wide acceptance within the robotics community. Tkinter, a standard Python interface to the Tk GUI toolkit, was chosen to construct the GUI because of its simplicity, versatility, and ease of use.

The library encapsulates all the elementary functions of the robot such as Cartesian mode, Joint mode, Homing, and Folding. This holistic approach ensures that the users have full

control over the robot's operations and can conveniently switch between various modes as per their requirements. The GUI acts as a user-friendly layer between the operator and the complex robotic control systems, simplifying interactions and enhancing usability.

The GUI communicates with the EtherCAT libraries through the "DoMyBidding" topic. This topic transmits a message to the robot control library containing the desired cartesian position or joint position and the operation mode. This communication protocol ensures precise and timely execution of commands, thereby enabling efficient real-time control of the assistive robot.

6.4.2 Robot control Libraries

The robot control library is essentially an amalgamation of three distinct libraries: the Object Dictionary library, the Motor library, and the Robot library. Each of these libraries plays a unique role in ensuring smooth control and operation of the robot.

- **Object Dictionary Library:** This library manages the Service Data Object (SDO) configuration. Utilizing the address of the SDO and the type of variable, it creates an object dictionary object. This object facilitates easy reading and writing with the SDO variable of the slave.

- **Motor Library:** This library deals with the configuration of the Process Data Objects (PDO), as well as the reading and writing of PDOs. It also manages the implementation of the control modes available in the slave. To achieve this, the library instantiates the required SDOs for the PDO configuration. Consequently, the library can implement various control modes such as Profile Position (PP), Continuous Synchronous Torque (CST),

and Continuous Synchronous Position (CSP). Additionally, this library also manages the different states of the motor and EtherCAT communication, ensuring reliable communication. It achieves these functionalities through a motor object instance that is capable of performing all the aforementioned operations.

- **Robot Library:** The primary role of this library is to facilitate control over the robot. This is achieved by creating six instances from the Motor library, thereby gaining control over the six distinct motors. The library is capable of managing various operational modes, including Cartesian control and joint-based control. Furthermore, it communicates with the Kinematics library via the "isViable" topic, transmitting both the current position and the intended position.

  Additionally, this library receives commands from the graphical user interface (GUI) via the "doMybidding" topic, as previously described.

6.4.3 Kinematics library

The Kinematics library is responsible for the implementation of the HUNTER model, ensuring the precision of the robot's movements and overall system stability. This library operates within a real-time ROS2 node to guarantee the system's stability, primarily because the computation of the HUNTER model is the most computationally intensive step.

To enhance the efficiency of the method, all matrices are treated in one column format, thereby eliminating the need for dealing with groups of concatenated arrays. Utilizing the properties of the Hessian matrix—its symmetry and positive definiteness—suitable methods were applied to compute the eigenvalues and the inverse. Specifically, the Cholesky method

was used for inverting the matrix, and the Jacobi method was employed for calculating the eigenvalues, both of which significantly increased the computational speed.

In addition to these optimizations, the library also checks if the solution provided by the HUNTER model is within the joint limits. If not, the algorithm adjusts the joint limits and reruns the HUNTER model until the solution falls within the acceptable joint limits. This iterative adjustment procedure ensures that the final solution aligns with the physical constraints of the robot, maintaining safety while maximizing maneuverability.

The desired position and orientation for the robot's movements are obtained from the Robot Library through the "isViable" topic. Once the Kinematics library processes this data and computes the desired joint values using HUNTER, it checks for potential collisions before proceeding. This is achieved by passing the joint values to the Collision Detection library via the 'isSecure' topic.

If there's no collision risk, the 'isSecure' topic returns a boolean one, and the Kinematics library sends back the computed desired position to the Robot Library through the "isViable" topic. This ensures the robot to move to the new position safely.

However, if there's a collision risk, the 'isSecure' topic returns a boolean zero. In this case, the Kinematics library sends back the same initial position to the Robot Library as the desired joint position via the "isViable" topic. This ensures that the robot does not move and maintains its initial position, effectively avoiding any potential collision.

6.4.4 Collision avoidance library

The Collision Avoidance library utilizes the algorithm discussed in Chapter (5.2). It is designed to work effectively with complex and dynamic environments by continuously updating and

processing data.

One key implementation detail of this library is the use of Oriented Bounding Boxes (OBBs). OBBs are used to represent each link of the robot. They are three-dimensional boxes that unlike Axis-Aligned Bounding Boxes (AABBs), are not restricted to be aligned with the axes of the coordinate system. This means they can be rotated to more tightly fit around the object, which in this case is the robot link, providing a more accurate representation of the robot's geometry.

The library creates an instance of an OBB for each link of the robot, taking into account the current position and orientation of the link. Then, for each proposed movement of the robot, the library updates the OBBs and checks for overlaps between them and any obstacles in the environment, such as the user or wheelchair. If an overlap is detected, this indicates a potential collision, and the library communicates this information back to the Kinematics library, as described earlier.

The Collision Avoidance library communicates with the Kinematics library through the 'isSafe' topic, as previously mentioned. This dedicated topic serves as a communication channel for passing potential collision alerts and safety confirmations.

By combining the collision avoidance algorithm with the use of OBBs, the library provides a robust and efficient system for predicting and preventing potential collisions. Moreover, by implementing these functionalities in C++, the library ensures high-speed operation, which is crucial for real-time performance.

6.5 Conclusion

In conclusion, the architecture of the proposed library plays a crucial role in managing the operations of the mR2A Assistive Robot, providing a comprehensive structure that balances user interaction, robot control, motion planning, and safety protocols.

The **Control Interface GUI** serves as an essential conduit for user interaction, simplifying the control of the robot and rendering its complex functions accessible through an intuitive interface. This Python-based GUI bridges the gap between the user and the advanced technical operations of the robot, providing a user-friendly platform to interact with and operate the robot.

The **Robot Control Libraries**, which include the Object Dictionary Library, the Motor Library, and the Robot Library, are integral to maintaining reliable communication between the control system and the EtherCAT slaves (motors), enabling various modes of control and managing the state of the motors.

The **Kinematics Library**, on the other hand, focuses on the precise execution of the robot's movements. It implements the HUNTER model, using optimized computational methods to ensure system stability and accuracy. Moreover, it checks the feasibility of proposed movements against the physical constraints of the robot, maintaining safety and ensuring maximum maneuverability.

Lastly, the **Collision Avoidance Library** adds an extra layer of safety, ensuring the robot's operations remain safe in dynamic environments. By representing the robot's geometry through Oriented Bounding Boxes (OBBs) and checking for overlaps with any obstacles, the library provides a robust mechanism for predicting and avoiding potential collisions.

This proposed library architecture, thus, presents a harmonious integration of diverse com-

ponents, each serving a distinct purpose but collectively contributing to the efficient and safe operation of the Assistive Robot. The clear demarcation of responsibilities among these components ensures a highly modular design, aiding in maintenance and potential future enhancements. Ultimately, it paves the way for seamless human-robot interaction in assistive robotics, underscoring the potential of such systems in improving the quality of life for users.

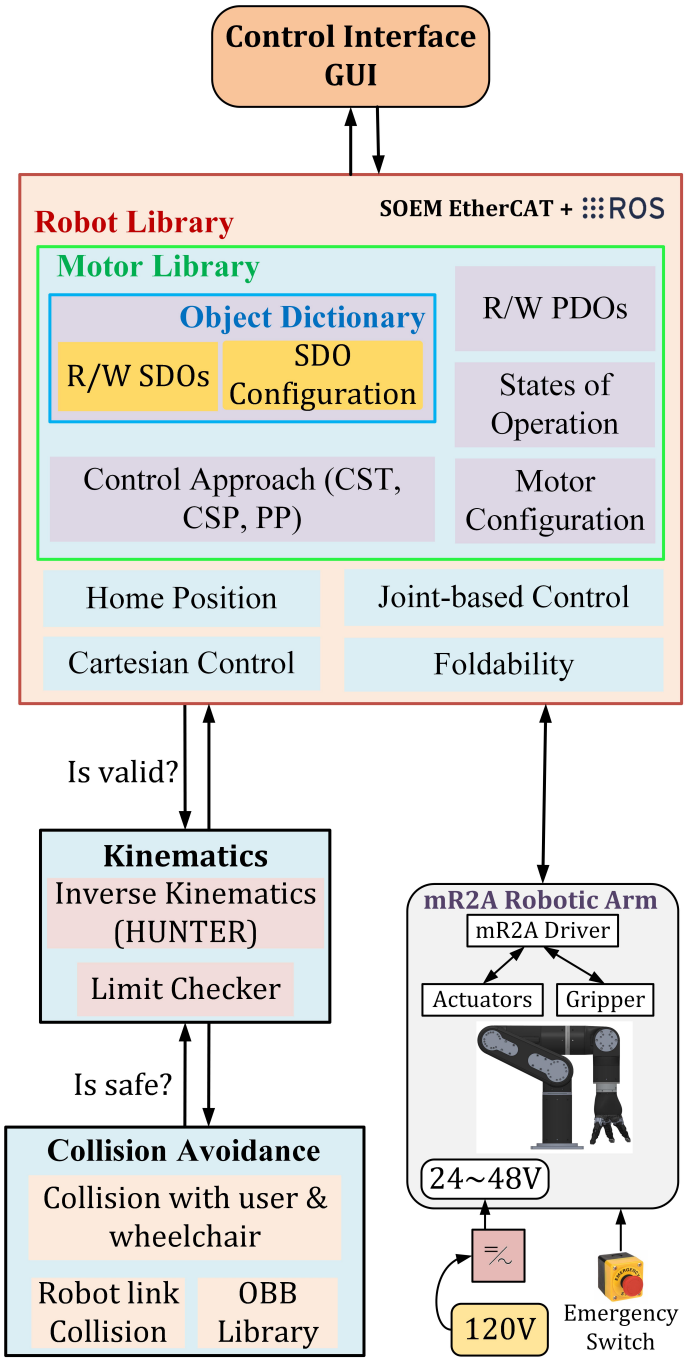


Figure 6.1: Library Architecture for the Assistive Robot.

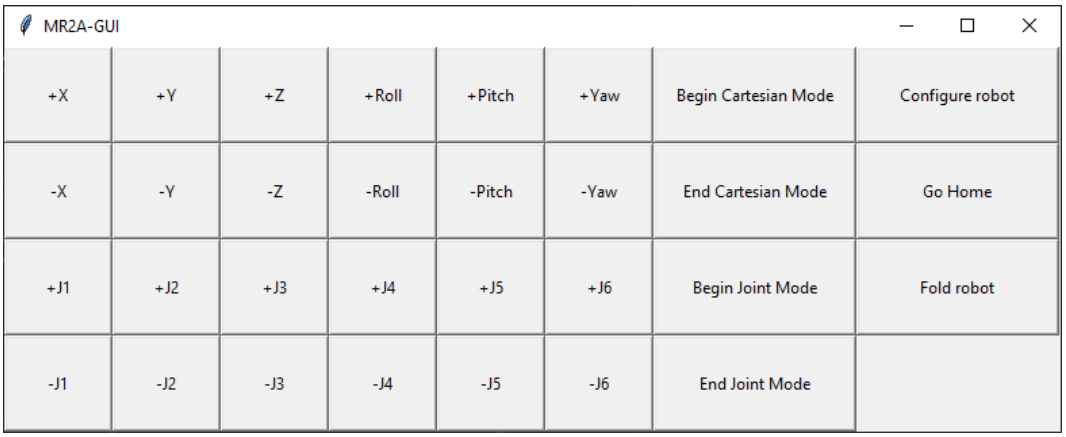


Figure 6.2: The Graphical User Interface (GUI) developed for controlling the assistive robot.

CHAPTER 7

FROM THEORY TO PRACTICE: TESTING THE ASSISTIVE ROBOT

The journey from conceptualization to realization of an assistive robot has unfolded throughout the preceding chapters, traversing design and development phases, optimization process, formulation of an SOEM-based library for robot control, and integration with ROS2. Approaching the culmination of this journey, the chapter turns towards testing and validating the physical robot against established objectives and specifications.

In this chapter, the performance and reliability of the assistive robot in its physical form is evaluated using the developed SOEM-based library implemented in ROS2. The evaluation process consists of four key components that provide a holistic view of the robot's performance and operational capabilities.

Firstly, the workspace of the physical robot is validated in Section 7.1. This process ensures alignment with the workspace as defined in the optimization process and confirms that the robot can cover the required range for carrying out the Activities of Daily Living (ADLs).

Secondly, the robot's folding mechanism is assessed in Section 7.2. This part of the validation

process ensures that the robot can fold into a compact form without issues when not in use, further emphasizing its practicality in daily life.

The third step, presented in Section 7.3, involves testing the behavior of the novel inverse kinematics algorithm, HUNTER, in various ADL scenarios. This includes examining its performance under singularities and positions out of reach. This assessment helps understand the efficiency and reliability of HUNTER in a real-world setting.

Lastly, Section 7.4 verifies the load-bearing capacity of the robot. This crucial aspect of the validation process directly affects the robot's ability to assist in daily activities. The objective is to confirm that the robot can comfortably hold and manipulate a 3kg weight, as per the design specifications.

All these tests are conducted using the developed SOEM-based libraryimplemented in ROS2, which further demonstrates the library's functionality and efficiency in controlling and managing the assistive robot. Through this rigorous validation process, the chapter aims to demonstrate the real-world applicability and robustness of the designed assistive robot and the SOEM-based control library. As a result, this chapter stands as a testament to the successful realization of the objectives and the commencement of a new era of assistive robotics in Section 7.5.

7.1 Workspace Validation

The validation of the workspace is conducted across each of the seven workspaces proposed throughout this manuscript. The key objective of this validation process is to confirm that the assistive robot can effectively maneuver within these workspaces.

The validation results provide insights into various aspects of the robot's operation within

these workspaces. These aspects include the joint positions, the trajectory of the end-effector, the torque required by the motors, and the velocities of the trajectory.

Each workspace is independently analyzed and studied in detail in the following subsections. This meticulous examination provides a comprehensive understanding of the robot's performance and operational capabilities within each workspace, laying the groundwork for further evaluations.

7.1.1 Workspace 1

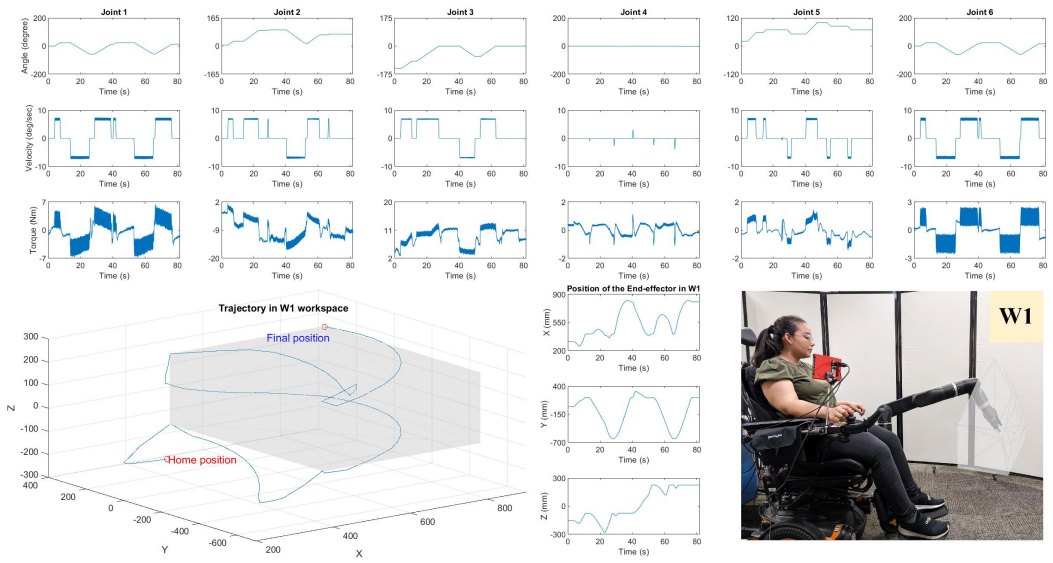


Figure 7.1: Evaluation Results for Workspace 1

Figure 7.1 presents the evaluation results for Workspace 1. This workspace is primarily associated with tasks involving handling objects on a table. The trajectory depicted showcases the robot maneuvering within the defined boundaries of the workspace. It is evident from the

plot that the robot is capable of achieving the desired end-effector positions, demonstrating its ability to operate within the constraints of Workspace 1.

From the accompanying image of the user interacting with the robot, it can be observed that the designed workspace aligns well with the real-world scenarios. The user is able to perform the Activities of Daily Living (ADLs) comfortably, indicating that the robot's structure and range of motion do not obstruct or impede these tasks.

Furthermore, the torque and velocity plots reveal that the robot operates within the defined limits, as specified in Table (5.6). This confirms the robot's capability to carry out the tasks associated with this workspace without overstressing the motors.

In conclusion, the evaluation results affirm the robot's ability to efficiently perform tasks within Workspace 1, providing a practical and versatile solution for the ADLs. The subsequent sections will further delve into the robot's performance within the other predefined workspaces.

7.1.2 Workspace 2

The evaluation of Workspace 2 is presented in Figure 7.2. Specifically, Workspace 2 corresponds to the act of picking and placing objects from the ground. The depicted trajectory and supporting data once again illustrate that the robot can successfully navigate within the given workspace.

As with Workspace 1, the torque values required for the tasks fall within the torque limits outlined in Table (5.6). This highlights the robot's capacity to effectively perform tasks that require ground-level interaction, reinforcing the robot's versatility and the robustness of the chosen motor configuration.

The trajectory plot, coupled with the torque, velocity, and position graphs, reiterate the

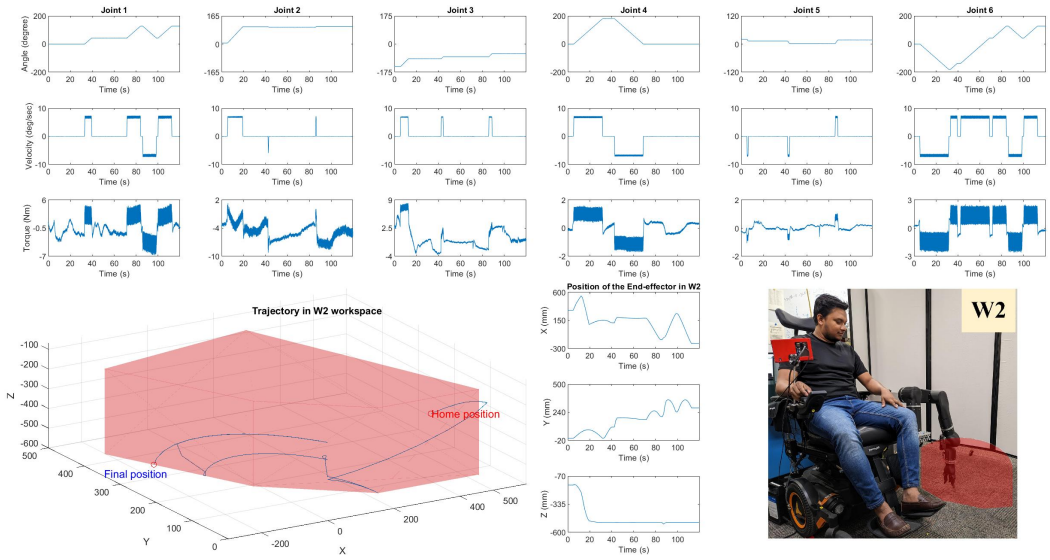


Figure 7.2: Evaluation Results for Workspace 2

robot's competence in maneuvering within the defined workspace, thereby ensuring the efficient performance of the tasks linked to Workspace 2.

7.1.3 Workspace 3

The results of Workspace 3 evaluation are shown in Figure 7.3. This workspace specifically pertains to tasks associated with handling objects on an upper shelf.

Similar to the previous workspaces, the trajectory and data plots demonstrate that the robot is able to operate efficiently within the constraints of Workspace 3. The end-effector follows the desired path successfully, indicating the effectiveness of the HUNTER algorithm and the robustness of the control system.

The torque values required for the tasks are once again within the specified torque limits,

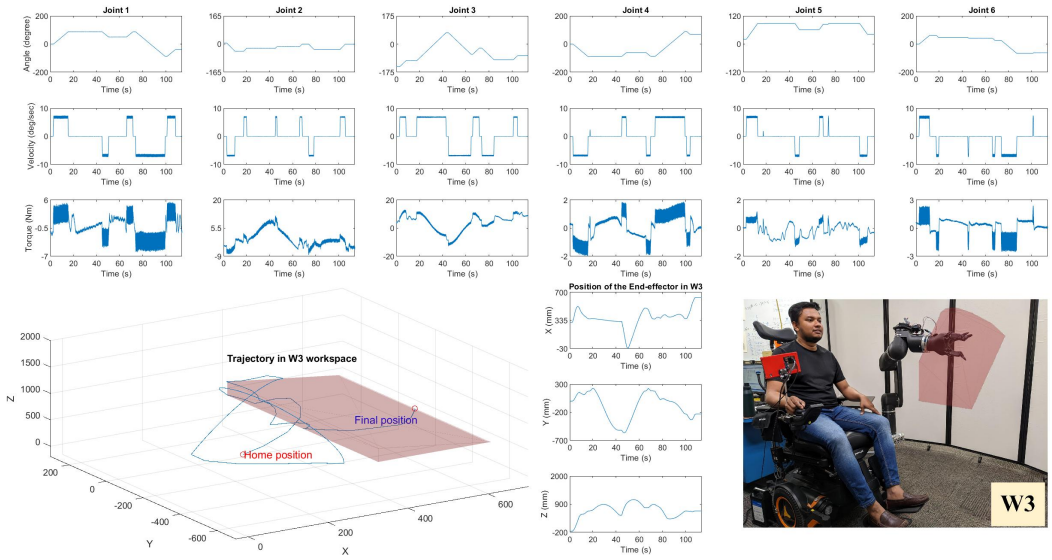


Figure 7.3: Evaluation Results for Workspace 3

as per Table (5.6). This highlights the robot's adaptability and capacity to handle tasks that involve reaching higher levels, further strengthening the case for the robot's versatility and the effectiveness of its motor configuration.

7.1.4 Workspace 4

Figure 7.4 illustrates the evaluation results of Workspace 4, which is designated for tasks involving the opening and closing of drawers. The plotted trajectory demonstrates the robot's movement within and outside the workspace boundaries. The extra-workspace trajectories are indicative of the robot finding the best configuration to accomplish the task at hand.

The graphical depiction of the robot in conjunction with the trajectory plot offers a comprehensive perspective on the robot's movement. It exhibits that the robot, while searching for

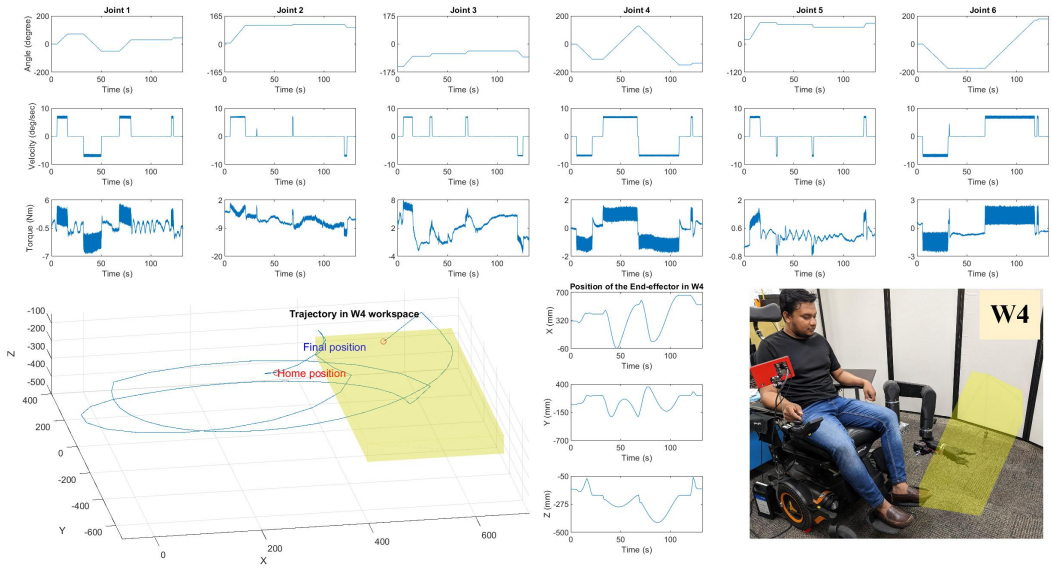


Figure 7.4: Evaluation Results for Workspace 4

an optimal configuration, is able to cover a large range of motion, thus suggesting its suitability for tasks related to manipulating drawers.

Further, the torque plots reiterate the robot's adherence to the defined performance parameters. All the values lie within the predefined limits as stated in Table (5.6). This affirms the robot's operational capacity and safety during task execution within Workspace 4.

Conclusively, the assistive robot demonstrates its capability to handle drawer-related tasks efficiently within Workspace 4.

7.1.5 Workspace 5

Figure 7.5 provides the evaluation results for Workspace 5, which is associated with opening and closing doors. The trajectory plot manifests the robot's movements within the specified

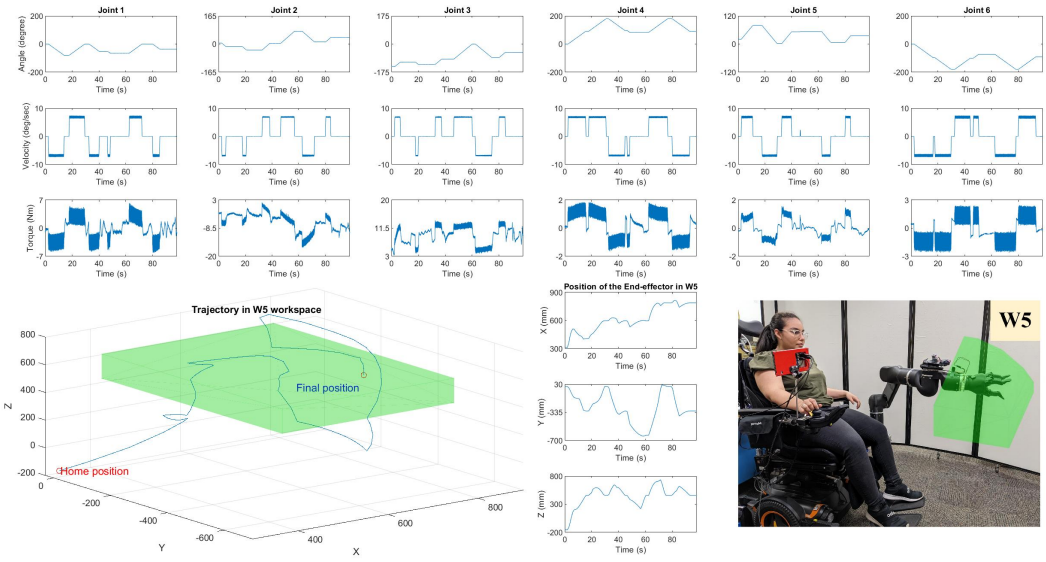


Figure 7.5: Evaluation Results for Workspace 5

workspace, effectively illustrating its proficiency in handling tasks associated with door manipulation.

7.1.6 Workspace 6

Figure 7.6 exhibits the evaluation results for Workspace 6, which pertains to tasks involving handing over objects to the user. The trajectory depicted in the plot reveals a distinct movement pattern: the robot first transitions to a suitable orientation before entering the defined workspace.

This preliminary motion, although seemingly outside the workspace, is a critical step towards achieving an efficient configuration for the task at hand. This maneuver demonstrates the robot's capability to adapt and reorient itself for various tasks, a feature that significantly

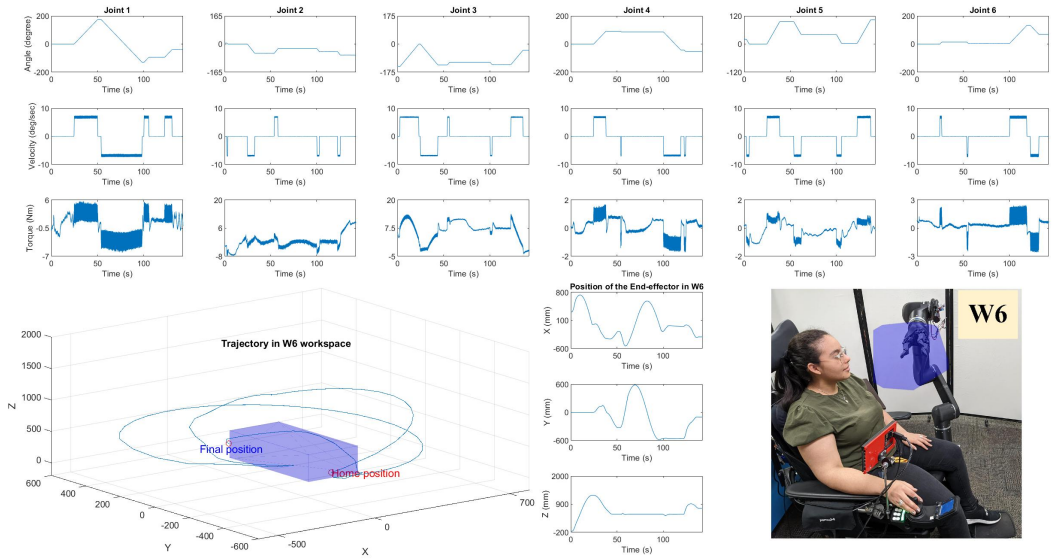


Figure 7.6: Evaluation Results for Workspace 6

enhances its operational versatility.

The corresponding plots of joint positions, velocity, and torque provide a comprehensive understanding of the robot's operational dynamics during this task. Consistent with previous results, the torque requirements and velocities remain within the predefined limits, reflecting the robot's ability to perform tasks efficiently and safely.

The user interaction picture accompanying the plots validates the practicality of the designed workspace. The assistive robot can be seen handing over an object to the user comfortably, indicating the robot's adeptness in facilitating user-centric tasks.

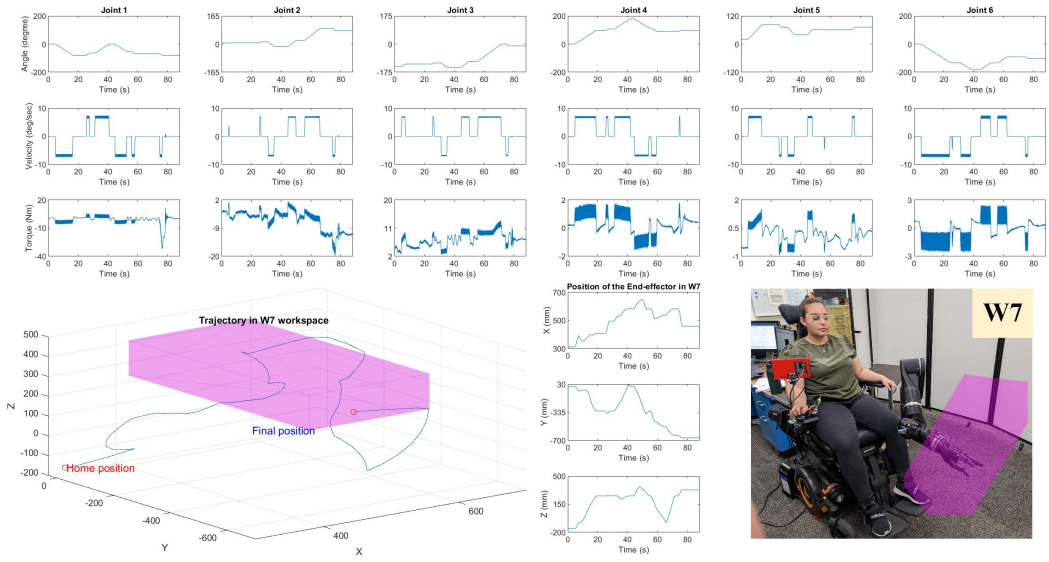


Figure 7.7: Evaluation Results for Workspace 7

7.1.7 Workspace 7

The final evaluation, as depicted in Figure 7.7, focuses on Workspace 7, which involves precise tasks such as manipulating a pen over a table. The trajectory plot showcases the robot's movements within this defined workspace, providing a graphical representation of its operational capabilities for tasks that require a higher degree of precision and control.

The corresponding plots of joint positions, velocity, and torque display the robot's dynamic performance during this task. Once again, the results show that the torques required and the velocities achieved remain within the defined limits, indicating the robot's capacity to efficiently handle tasks with precision and safety.

From the user interaction image, it can be observed that the assistive robot handles the pen effortlessly, displaying a commendable level of control and stability. This finding further

affirms the robot's proficiency in performing fine motor tasks.

In conclusion, the successful operation of the assistive robot within the diverse workspaces confirms its potential as a robust and versatile aid in performing ADLs. The adaptability, precision, and ergonomic design evident in the robot's performance validate the efficacy of the design objectives outlined in the development process.

The successful maneuverability within these workspaces can be seen as a testament to the effectiveness of the optimization process. The robot's ability to function efficiently within the bounds of the pre-defined workspaces affirms that the established objectives and constraints in the optimization phase were well-founded and practically relevant.

7.2 Folding Mechanism Assessment

Figure 7.8 depicts the evaluation results of the folding mechanism of the assistive robot. This mechanism is integral to the robot's design, featuring the robot's position, velocity, torque, and trajectory during the folding process, along with a real-life image of the robot in its compact, folded form.

The trajectory showcases a non-linear path from the home position to the final folded state, demonstrating the complexity of the folding process. The y-value remains constant, indicating that the folding process is primarily planar, with no vertical motion involved.

When not in active use, the robot can fold into a compact shape, thereby minimizing its spatial footprint and ensuring its compatibility within everyday household environments. In its folded state, the robot allows for the user to comfortably rest their hand on the wheelchair's armrest, further emphasizing the user-centered design approach. This safety feature signifi-

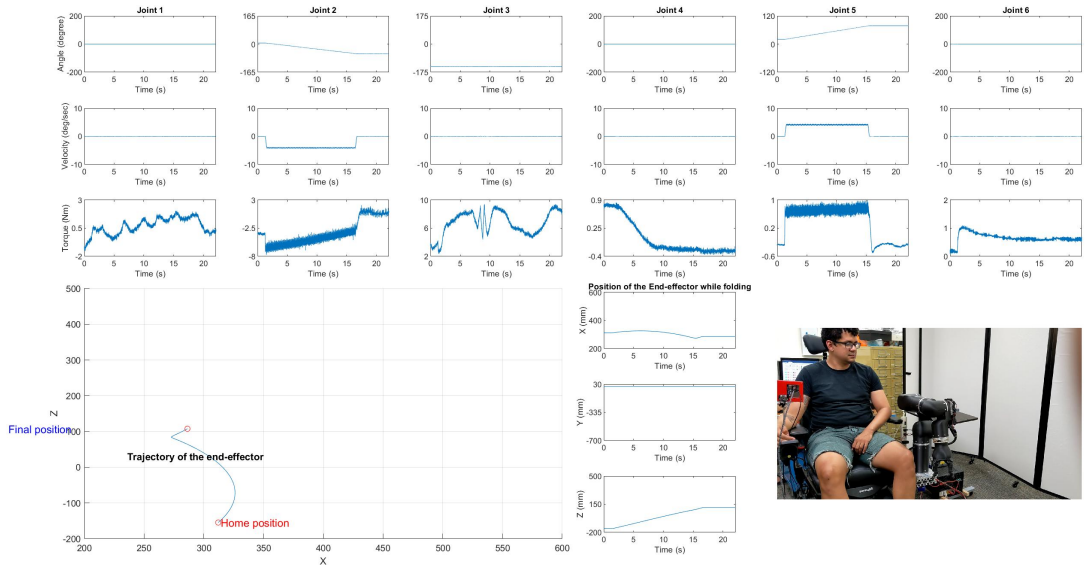


Figure 7.8: Evaluation Results of Folding Mechanism

cantly reduces the risk of accidental collisions or obstructions, ensuring that the robot provides assistance without impeding the user's personal space or comfort.

These evaluation results demonstrate the successful realization of the design objective of creating a foldable robot, reinforcing the robot's usability and adaptability in daily life scenarios. The successful implementation of the folding mechanism not only testifies to the practicality and user-friendliness of the assistive robot but also validates the effectiveness of the constraint implementation in the objective function of the optimization process.

7.3 HUNTER Performance Evaluation

This section is dedicated to the evaluation of the performance of HUNTER, the proposed inverse kinematics algorithm. To this end, three critical experiments are conducted, each focusing on a unique set of Activities of Daily Living (ADLs) that present the robot with varied and distinct challenges.

1. **Picking an Object from a Shelf:** This scenario potentially introduces singularities within the workspace as the robot has to reach for objects at different heights while avoiding obstacles.

2. **Picking an Object from the Floor:** This task demands the robot to stretch to its lower limit, thereby testing its range of motion and adaptability to extreme positions.

3. **Picking an Object from a Table:** This routine task presents an opportunity to assess the robot's precision and dexterity in handling objects at a comfortable height.

These experiments were carefully selected due to the singularities they introduce within the workspace. Moreover, some positions in these tasks may be out of the robot's immediate reach. However, thanks to HUNTER's extended reach feature, the robot can still align the end effector to effectively pick up these objects. This section's evaluation will present valuable insights into the efficiency and reliability of HUNTER in various practical scenarios.

7.3.1 Picking an object from a shelf

Figure 7.9 presents the trajectory followed by the robot for picking an object from a shelf. The trajectory highlights the ability of HUNTER to navigate the robot effectively through singular

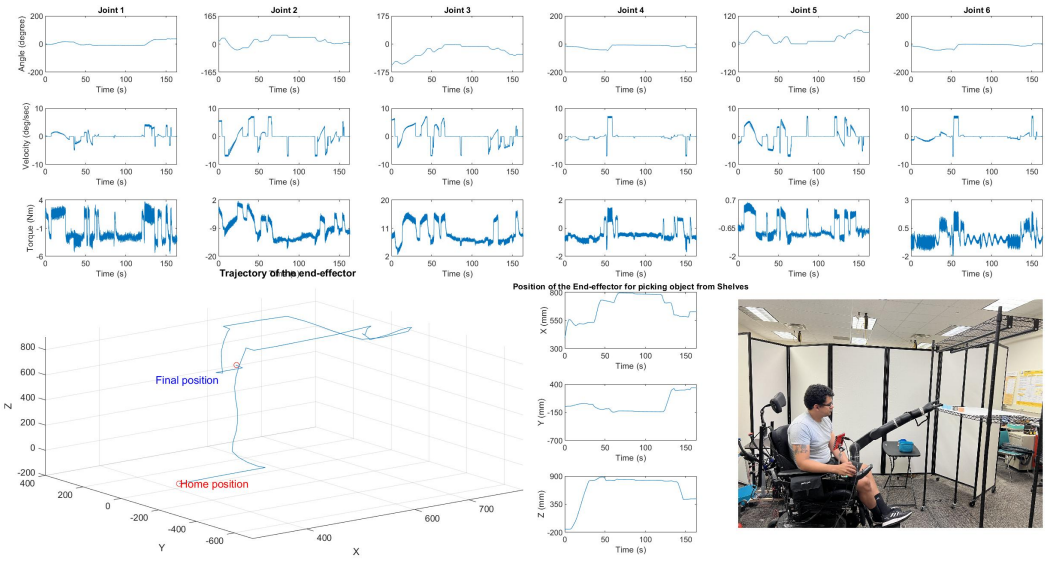


Figure 7.9: Evaluation Results for Inverse Kinematics in Shelf Scenario

positions, further validating its utility in solving inverse kinematics.

It is worth noting that the object was placed in a position beyond the robot's immediate workspace. This scenario thus presented an opportunity to test the extended reach feature of HUNTER. In response, HUNTER efficiently aligned the gripper towards the target object. Consequently, the wheelchair was maneuvered closer to the object, thereby facilitating a successful pick-up operation.

This successful execution under challenging conditions not only underscores the effectiveness of HUNTER but also illustrates the practical benefits of having an extended reach feature. The results from this evaluation demonstrate HUNTER's reliable performance and adaptability in diverse real-world scenarios.

7.3.2 Picking object from the floor

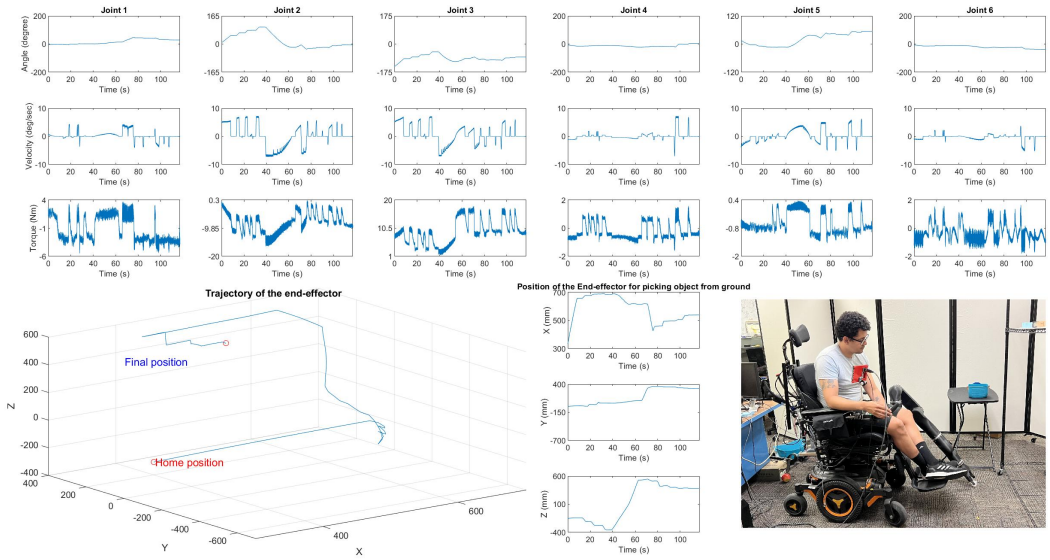


Figure 7.10: Evaluation Results for Inverse Kinematics in Floor Scenario

Figure 7.10 depicts the trajectory of the robot while picking an object from the floor. This complex task is inherently challenging due to the need for precise positioning and potential singularity issues near the ground. However, HUNTER successfully navigates the robot through these complexities, further substantiating its effectiveness in handling singularities, even in the most strenuous of scenarios.

The smooth and continuous nature of the trajectory, as observed in the joint plot, testifies to HUNTER's ability to compute solutions in real-time. This real-time computation capability is crucial for ensuring seamless and efficient robot operations in dynamic environments.

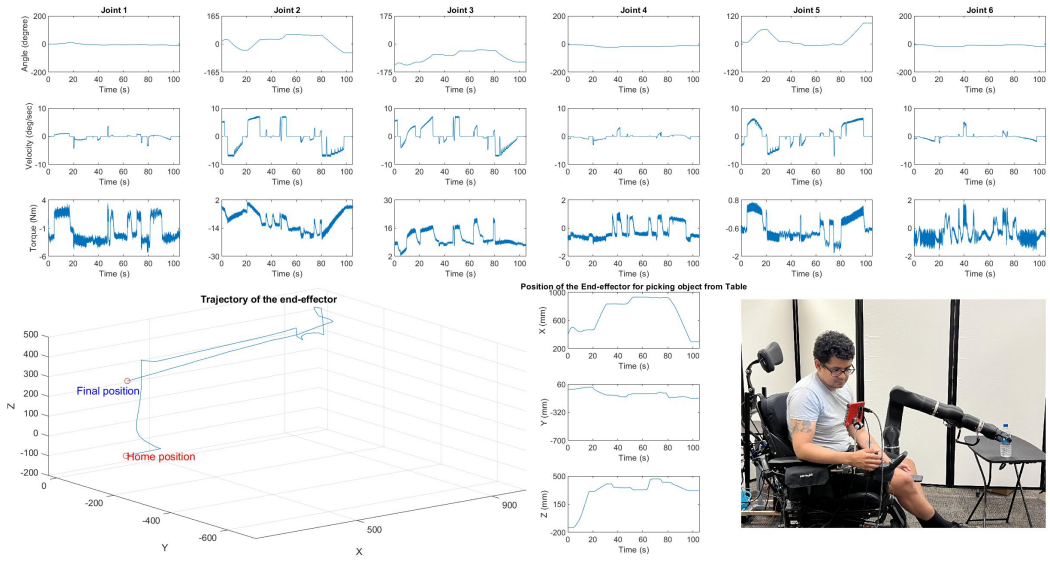


Figure 7.11: Evaluation Results for Inverse Kinematics in Table Scenario

7.3.3 Picking object from the table

Figure 7.11 demonstrates the results of implementing HUNTER for picking objects from a table. This task, although seemingly simple, is a fundamental operation in most ADLs and serves as a crucial test case for the assistive robot's capabilities within its designated workspace.

The successful completion of this task affirms the robot's efficiency in executing essential daily activities, reinforcing the robustness of HUNTER as a real-time inverse kinematics solution. The trajectory plot shows that the robot was able to smoothly navigate to the desired position to pick up the object from the table. This operation was achieved through a real-time implementation of HUNTER with a thread frequency of 1 millisecond, showcasing the solution's capability for instantaneous adaptation and response.

In conclusion, these results demonstrate HUNTER's ability to handle both simple and com-

plex tasks within the robot's workspace, consolidating its effectiveness in real-world application scenarios. The successful real-time implementation of HUNTER validates its precision and potential in various task contexts, reinforcing its applicability in the realm of assistive robotics.

7.4 Load-Bearing Capacity Verification

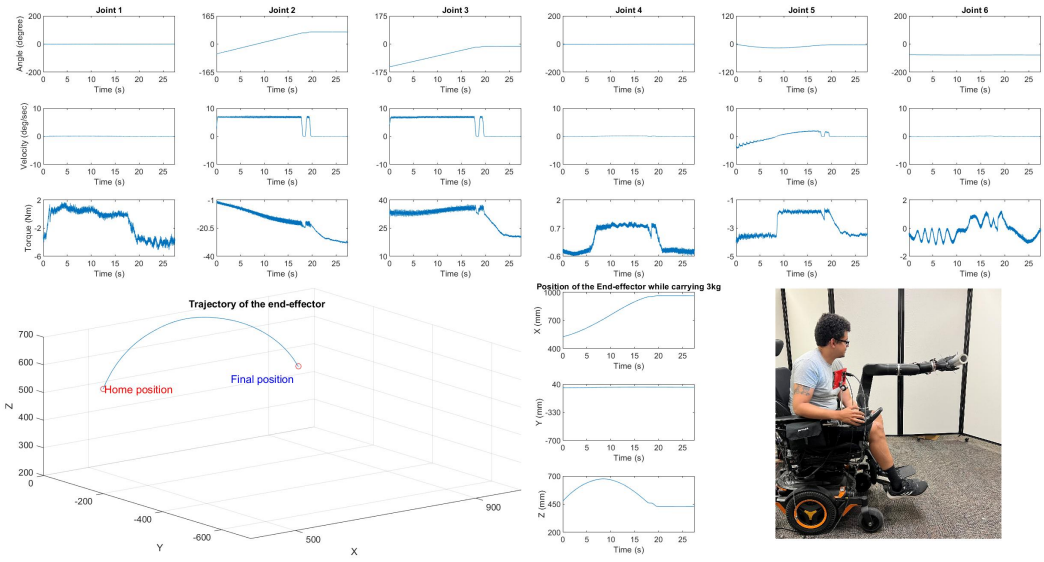


Figure 7.12: Load Handling Evaluation

Figure 7.12 showcases the results of a critical load handling test, where the robot was tasked with moving a 3 kg weight horizontally. The motive behind this examination was to ascertain the robot's load handling capabilities at maximum motor load, specifically in a fully extended horizontal configuration.

From the data obtained, it is evident that all the robot's motors managed to sustain the 3

kg load under their respective rated torques, with the exception of joint 3. This joint reached its rated torque of 32 Nm at a distance of 1.08 m from the origin, corresponding to approximately 80% of the robot's maximum reach. Therefore, one could conclude that the robot can effectively handle a 3 kg load within 80% of its maximum extension.

However, this extension is in relation to the robot's maximum reach and not the designated workspace used for design. Thus, while the robot has successfully demonstrated its load handling capabilities, there is still room for optimization and improvement.

Potential future research could focus on topological optimization to reduce the robot's weight or enhance the gripper's design to improve load retention. Although the gripper design falls outside the scope of this current research, its enhancement could further bolster the robot's load handling efficiency.

In conclusion, these results demonstrate the robot's capability to handle a 3 kg weight within its workspace, solidifying the relevancy and effectiveness of the developed methodology.

7.5 Conclusions

This chapter presented a comprehensive evaluation of the assistive robot's capabilities and performance. The assessments were carried out in a multi-faceted approach to comprehensively cover the diverse aspects of the robot's operations.

The assessments began with a workspace validation, evaluating seven distinct workspaces corresponding to different Activities of Daily Living (ADLs). The results from these evaluations underscored the robot's versatility and adaptability, as it successfully maneuvered within the constraints of each workspace. The satisfactory results affirm the robustness of the optimization

process undertaken in designing the robot.

Next, the focus shifted to validating HUNTER, the proposed inverse kinematics solution. Three distinct experiments were carried out, involving the robot picking objects from different locations: a shelf, the floor, and a table. These experiments showcased HUNTER's proficiency in handling singularities and achieving end-effector alignment even in challenging conditions, underscoring its real-time effectiveness.

The final assessment evaluated the robot's load handling capabilities. The robot was tasked with moving a 3 kg weight horizontally, and it demonstrated the ability to sustain this load within 80% of its maximum extension. While this test revealed room for future improvements, it also showcased the robot's substantial load handling capabilities within the workspace.

In conclusion, the successful evaluations across various aspects reaffirm the robustness and versatility of the assistive robot. The results provide a testament to the relevancy and effectiveness of the proposed methodology and design. They offer a solid foundation for future research directions, focusing on enhancements in specific areas such as topological optimization and gripper design. This ultimately aims to create an assistive robot capable of efficiently aiding users in a wide range of ADLs.

Bibliography

[1] Z. Bien, D.-J. Kim, M.-J. Chung, D.-S. Kwon, and P.-H. Chang, "Development of a wheelchair-based rehabilitation robotic system (kares ii) with various human-robot interaction interfaces for the disabled," in *Proceedings 2003 IEEE/ASME international conference on advanced intelligent mechatronics (AIM 2003)*, vol. 2. IEEE, 2003, pp. 902–907.

[2] M. Hillman, K. Hagan, S. Hagan, J. Jepson, and R. Orpwood, "The weston wheelchair mounted assistive robot-the design story," *Robotica*, vol. 20, no. 2, pp. 125–132, 2002.

[3] B. Driessen, H. Evers, and J. v Woerden, "Manus—a wheelchair-mounted rehabilitation robot," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of engineering in medicine*, vol. 215, no. 3, pp. 285–290, 2001.

[4] A. Campeau-Lecours, V. Maheu, S. Lepage, H. Lamontagne, S. Latour, L. Paquet, and N. Hardie, "Jaco assistive robotic device: Empowering people with disabilities through innovative algorithms," *Rehabilitation Engineering and Assistive Technology Society of North America (RESNA)*, 2016.

[5] I. Volosyak, O. Ivlev, and A. Graser, "Rehabilitation robot friend ii-the general concept

and current implementation," in *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.* IEEE, 2005, pp. 540–544.

[6] K. Edwards, R. Alqasemi, and R. Dubey, "Wheelchair-mounted robotic arms: design and development," in *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006.* IEEE, 2006, pp. 613–618.

[7] J. Vogel, A. Hagengruber, M. Iskandar, G. Quere, U. Leipscher, S. Bustamante, A. Dietrich, H. Höppner, D. Leidner, and A. Albu-Schäffer, "Edan: An emg-controlled daily assistant to help people with physical disabilities," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2020, pp. 4183–4190.

[8] Reeve Foundation, "Stats about paralysis," https://www.christopherreeve.org/living-with-paralysis/stats-about-paralysis, Accessed: 01/08/2020.

[9] B. S. Armour, E. A. Courtney-Long, M. H. Fox, H. Fredine, and A. Cahill, "Prevalence and causes of paralysis—united states, 2013," *American journal of public health*, vol. 106, no. 10, pp. 1855–1857, 2016.

[10] A.-L. Salminen, Å. Brandt, K. Samuelsson, O. Töytäri, and A. Malmivaara, "Mobility devices to promote activity and participation: a systematic review," pp. 697–706, 2009.

[11] E. Munoz, M. S. H. Sunny, I. Rulik, J. D. S. De Caro, and M. H. Rahman, "Kinematics and workspace analysis of xarm6 robot for activities of daily living," in *Proceedings. International Conference on Industrial & Mechanical Engineering and Operations Management*, vol. 4. NIH Public Access, 2021, p. 885.

[12] Robot Care Systems, "Care robot lea," https://www.sparkdesign.nl/nl/447projecten/veilig-vitaal-zorgrobot-lea, Accessed: 11/25/2022.

[13] H. A. Yanco, "Wheelesley: A robotic wheelchair system: Indoor navigation and user interface," *Assistive technology and artificial intelligence*, pp. 256–268, 1998.

[14] T. Mukai, S. Hirano, H. Nakashima, Y. Kato, Y. Sakaida, S. Guo, and S. Hosoe, "Development of a nursing-care assistant robot riba that can lift a human in its arms," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 5996–6001.

[15] M. S. H. Sunny, M. I. I. Zarif, I. Rulik, J. Sanjuan, M. H. Rahman, S. I. Ahamed, I. Wang, K. Schultz, and B. Brahmi, "Eye-gaze control of a wheelchair mounted 6dof assistive robot for activities of daily living," *Journal of NeuroEngineering and Rehabilitation*, vol. 18, no. 1, pp. 1–12, 2021.

[16] L. Petrich, J. Jin, M. Dehghan, and M. Jagersand, "A quantitative analysis of activities of daily living: Insights into improving functional independence with assistive robotics," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6999–7006.

[17] M. Adinehvand, E. Asadi, C. Y. Lai, H. Khayyam, and R. Hoseinnezhad, "Design and adaptive control of a kinematically redundant robot with enhanced trajectory tracking for climbing in tight spaces," *Mechanism and Machine Theory*, vol. 177, p. 104994, 2022.

[18] A. Graser, T. Heyer, L. Fotoohi, U. Lange, H. Kampe, B. Enjarini, S. Heyer, C. Fragkopoulos, and D. Ristic-Durrant, "A supportive friend at work: Robotic work-

place assistance for the disabled," *IEEE Robotics & Automation Magazine*, vol. 20, no. 4, pp. 148–159, 2013.

[19] J. Sanjuan, D. Serje, and J. Pacheco, "Closed form solution for direct and inverse kinematics of a us-rs-rps 2-dof parallel robot," *Scientia Iranica*, vol. 25, no. 4, pp. 2144–2154, 2018.

[20] C. Yang, W. Ye, and Q. Li, "Review of the performance optimization of parallel manipulators," *Mechanism and Machine Theory*, vol. 170, p. 104725, 2022.

[21] A. Rosyid, B. El-Khasawneh, and A. Alazzam, "Performance measures of parallel kinematics manipulators," *Mechanical Sciences*, vol. 11, no. 1, pp. 49–73, 2020.

[22] S. B. Liu and M. Althoff, "Optimizing performance in automation through modular robots," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4044–4050.

[23] B. Bounab, "Multi-objective optimal design based kineto-elastostatic performance for the delta parallel mechanism," *Robotica*, vol. 34, no. 2, pp. 258–273, 2016.

[24] M. Russo, S. Herrero, O. Altuzarra, and M. Ceccarelli, "Kinematic analysis and multi-objective optimization of a 3-upr parallel mechanism for a robotic leg," *Mechanism and Machine Theory*, vol. 120, pp. 192–202, 2018.

[25] J. Jiao, W. Tian, W. Liao, L. Zhang, and Y. Bu, "Processing configuration off-line optimization for functionally redundant robotic drilling tasks," *Robotics and Autonomous Systems*, vol. 110, pp. 112–123, 2018.

[26] C. A. Nelson, M. A. Laribi, and S. Zeghloul, "Multi-robot system optimization based on redundant serial spherical mechanism for robotic minimally invasive surgery," *Robotica*, vol. 37, no. 7, pp. 1202–1213, 2019.

[27] J. Brinker, B. Corves, and Y. Takeda, "Kinematic performance evaluation of high-speed delta parallel robots based on motion/force transmission indices," *Mechanism and Machine Theory*, vol. 125, pp. 111–125, 2018.

[28] W. Ye, X. Chai, and K. Zhang, "Kinematic modeling and optimization of a new reconfigurable parallel mechanism," *Mechanism and Machine Theory*, vol. 149, p. 103850, 2020.

[29] J. Peng, Y. Ding, G. Zhang, and H. Ding, "Smoothness-oriented path optimization for robotic milling processes," *Science China Technological Sciences*, vol. 63, no. 9, pp. 1751–1763, 2020.

[30] A. Zeiaee, R. Soltani-Zarrin, R. Langari, and R. Tafreshi, "Kinematic design optimization of an eight degree-of-freedom upper-limb exoskeleton," *Robotica*, vol. 37, no. 12, pp. 2073–2086, 2019.

[31] B. Rout and R. Mittal, "Simultaneous selection of optimal parameters and tolerance of manipulator using evolutionary optimization technique," *Structural and Multidisciplinary Optimization*, vol. 40, no. 1, pp. 513–528, 2010.

[32] N. Sihag and K. S. Sangwan, "A systematic literature review on machine tool energy consumption," *Journal of Cleaner Production*, vol. 275, p. 123125, 2020.

[33] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, "Path planning and trajectory planning algorithms: A general overview," *Motion and operation planning of robotic systems*, pp. 3–27, 2015.

[34] V. Arakelian, "Inertia forces and moments balancing in robot manipulators: A review," *Advanced Robotics*, vol. 31, no. 14, pp. 717–726, 2017.

[35] J. T. Allison, "Plant-limited co-design of an energy-efficient counterbalanced robotic manipulator," *Journal of Mechanical Design*, vol. 135, no. 10, p. 101003, 2013.

[36] F. Wang, Z. Qian, Y. Lin, and W. Zhang, "Design and rapid construction of a cost-effective virtual haptic device," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 1, pp. 66–77, 2020.

[37] A. Molaei, N. A. Foomany, M. Parsapour, and J. Dargahi, "A portable low-cost 3d-printed wrist rehabilitation robot: Design and development," *Mechanism and Machine Theory*, vol. 171, p. 104719, 2022.

[38] Y. Gu, S. Feng, Y. Guo, F. Wan, J. S. Dai, J. Pan, and C. Song, "Overconstrained coaxial design of robotic legs with omni-directional locomotion," *Mechanism and Machine Theory*, vol. 176, p. 105018, 2022.

[39] J. J. Craig, *Introduction to robotics*. Pearson Educacion, 2006.

[40] R. Diankov, *Automated construction of robotic manipulation programs*. Carnegie Mellon University, 2010.

[41] J. Villalobos, I. Y. Sanchez, and F. Martell, "Singularity analysis and complete methods to compute the inverse kinematics for a 6-dof ur/tm-type robot," *Robotics*, vol. 11, no. 6, p. 137, 2022.

[42] M. H. Rahman, "I-corps: Wheelchair mounted robotic assistive arm," may 2018, nSF Award # 1848912.

[43] L. Petrich, J. Jin, M. Dehghan, and M. Jagersand, "Assistive arm and hand manipulation: How does current research intersect with actual healthcare needs?" *arXiv preprint arXiv:2101.02750*, 2021.

[44] C.-S. Chung, H. Wang, and R. A. Cooper, "Functional assessment and performance evaluation for assistive robotic manipulators: Literature review," *The journal of spinal cord medicine*, vol. 36, no. 4, pp. 273–289, 2013.

[45] R. Klebbe, S. Scherzinger, C. Eicher *et al.*, "Assistive robots for patients with amyotrophic lateral sclerosis: Exploratory task-based evaluation study with an early-stage demonstrator," *JMIR Rehabilitation and Assistive Technologies*, vol. 9, no. 3, p. e35304, 2022.

[46] W.-K. Song, H. Lee, and Z. Bien, "Kares: Intelligent wheelchair-mounted robotic arm system using vision and force sensor," *Robotics and Autonomous Systems*, vol. 28, no. 1, pp. 83–94, 1999.

[47] D. Ramirez and R. Foulds, "Adaptive real-time interfaces for wheelchair-mounted manipulators in unstructured environments," in *Proceedings of the 2010 IEEE 36th Annual Northeast Bioengineering Conference (NEBEC)*. IEEE, 2010, pp. 1–2.

[48] O. Ivlev, C. Martens, and A. Graeser, "Rehabilitation robots friend-i and friend-ii with the dexterous lightweight manipulator," *Technology and Disability*, vol. 17, no. 2, pp. 111–123, 2005.

[49] M. Palankar, K. J. De Laurentis, R. Alqasemi, E. Veras, R. Dubey, Y. Arbel, and E. Donchin, "Control of a 9-dof wheelchair-mounted robotic arm system using a p300 brain computer interface: Initial experiments," in *2008 IEEE International Conference on Robotics and Biomimetics*. IEEE, 2009, pp. 348–353.

[50] P. Schrock, F. Farelo, R. Alqasemi, and R. Dubey, "Design, simulation and testing of a new modular wheelchair mounted robotic arm to perform activities of daily living," in *2009 IEEE International Conference on Rehabilitation Robotics*. IEEE, 2009, pp. 518–523.

[51] M. H. Rahman, "I-corps: Wheelchair mounted robotic assistive arm," May 2018, nSF Award # 1848912.

[52] E. I. Isik, S. Yilmaz, I. Uysal, and S. Basar, "Adaptation of the lawton instrumental activities of daily living scale to turkish: validity and reliability study," *Annals of geriatric medicine and research*, vol. 24, no. 1, p. 35, 2020.

[53] W. A. Alkeridy, T. A. Muayqil, R. A. Al Khalifah, A. S. Mohammedin, R. A. Khallaf, and R. S. Bucks, "Arabic translation and cross-cultural adaptation of the bristol activities of daily living scale (badls)," *Alzheimer's & Dementia*, vol. 17, p. e050856, 2021.

[54] A. M. Dollar, "Classifying human hand use and the activities of daily living," in *The*

Human Hand as an Inspiration for Robot Hand Development. Springer, 2014, pp. 201–216.

[55] J. Zhang, N. Zhao, and F. Qu, "Bio-inspired flapping wing robots with foldable or deformable wings: a review," *Bioinspiration & Biomimetics*, 2022.

[56] S.-H. Kim, B. Yoon, H. J. Lee, S. Kim, K.-S. Kim, J. C. Kim, and T. Y. Noh, "Portable serial robot manipulator with distributed actuation mechanism," in *2014 14th International Conference on Control, Automation and Systems (ICCAS 2014).* IEEE, 2014, pp. 1590–1593.

[57] J. W. Kang, B. S. Kim, and M. J. Chung, "Development of assistive mobile robots helping the disabled work in a factory environment," in *2008 IEEE/ASME International Conference on Mechtronic and Embedded Systems and Applications.* IEEE, 2008, pp. 426–431.

[58] T. Wang, S. Luan, L. Hu, Z. Liu, W. Li, and L. Jiang, "Force-based control of a compact spinal milling robot," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 6, no. 2, pp. 178–185, 2010.

[59] T. Hassan, M. Manti, G. Passetti, N. d'Elia, M. Cianchetti, and C. Laschi, "Design and development of a bio-inspired, under-actuated soft gripper," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC).* IEEE, 2015, pp. 3619–3622.

[60] S. Seok, A. Wang, M. Y. Chuah, D. Otten, J. Lang, and S. Kim, "Design principles for

highly efficient quadrupeds and implementation on the mit cheetah robot," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3307–3312.

[61] I. Rulik, M. S. H. Sunny, J. D. S. De Caro, M. I. I. Zarif, B. Brahmi, S. I. Ahamed, K. Schultz, I. Wang, T. Leheng, J. P. Longxiang *et al.*, "Control of a wheelchair-mounted 6dof assistive robot with chin and finger joysticks," *Frontiers in Robotics and AI*, vol. 9, 2022.

[62] I. Rulik, A. A. Z. Swapnil, J. S. De Caro, M. S. H. Sunny, M. Rahman, I. Wang, S. Ahamed, and K. Schultz, "Finger/chin joystick control for robotic arms and power wheelchair systems," *Archives of Physical Medicine and Rehabilitation*, vol. 102, no. 10, p. e84, 2021.

[63] M. S. H. Sunny, M. I. I. Zarif, I. Rulik, J. Sanjuan, M. H. Rahman, S. I. Ahamed, I. Wang, K. Schultz, and B. Brahmi, "Eye-gaze control of a wheelchair mounted 6dof assistive robot for activities of daily living," *Journal of NeuroEngineering and Rehabilitation*, vol. 18, no. 1, pp. 1–12, 2021.

[64] S. R. Buss, "Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods," *IEEE Journal of Robotics and Automation*, vol. 17, no. 1-19, p. 16, 2004.

[65] L. Unzueta, M. Peinado, R. Boulic, and Á. Suescun, "Full-body performance animation with sequential inverse kinematics," *Graphical models*, vol. 70, no. 5, pp. 87–104, 2008.

[66] J.-T. Lapresté, F. Jurie, M. Dhome, and F. Chaumette, "An efficient method to compute the inverse jacobian matrix in visual servoing," in *IEEE International Conference on*

Robotics and Automation, 2004. Proceedings. ICRA'04. 2004, vol. 1. IEEE, 2004, pp. 727–732.

[67] S. R. Buss and J.-S. Kim, "Selectively damped least squares for inverse kinematics," *Journal of Graphics tools*, vol. 10, no. 3, pp. 37–49, 2005.

[68] A. Colomé and C. Torras, "Redundant inverse kinematics: Experimental comparative review and two enhancements," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5333–5340.

[69] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU press, 2013.

[70] Y. Nakamura, *Advanced robotics: redundancy and optimization*. Addison-Wesley Longman Publishing Co., Inc., 1990.

[71] P. C. Hansen, *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*. SIAM, 1998.

[72] S. A. Teukolsky, B. P. Flannery, W. Press, and W. Vetterling, "Numerical recipes in c," *SMR*, vol. 693, no. 1, pp. 59–70, 1992.

[73] S. Wright, J. Nocedal *et al.*, "Numerical optimization," *Springer Science*, vol. 35, no. 67-68, p. 7, 1999.

[74] L. T. Biegler, *Nonlinear programming: concepts, algorithms, and applications to chemical processes*. SIAM, 2010.

[75] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[76] N. Jorge and J. W. Stephen, "Numerical optimization," 2006.

[77] A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir, "Inverse kinematics techniques in computer graphics: A survey," in *Computer graphics forum*, vol. 37, no. 6. Wiley Online Library, 2018, pp. 35–58.

[78] K. Dautenhahn, "Design issues on interactive environments for children with autism," in *In: Procs of ICDVRAT 2000, the 3rd Int Conf on Disability, Virtual Reality and Associated Technologies*. University of Reading, 2000.

[79] D. Di Vito, C. Natale, and G. Antonelli, "A comparison of damped least squares algorithms for inverse kinematics of robot manipulators," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 6869–6874, 2017.

[80] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[81] C. Gosselin, J. Cote, and D. Laurendeau, "Inverse kinematic functions for approach and catching operations," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 783–791, 1993.

[82] F. Cavallo, R. Esposito, R. Limosani, A. Manzi, R. Bevilacqua, E. Felici, A. Di Nuovo, A. Cangelosi, F. Lattanzio, P. Dario *et al.*, "Robotic services acceptance in smart environments with older adults: user satisfaction and acceptability study," *Journal of medical Internet research*, vol. 20, no. 9, p. e9460, 2018.

[83] M. A. Goodrich, A. C. Schultz *et al.*, "Human–robot interaction: a survey," *Foundations and Trends® in Human–Computer Interaction*, vol. 1, no. 3, pp. 203–275, 2008.

[84] G. Cui, D. Zhang, H. Zhou, and Y. Zhang, "Operating dexterity optimization and analysis of a 3-dof parallel manipulator for a tunnel segment assembly system," *International Journal of Mechanics and Materials in Design*, vol. 11, no. 3, pp. 277–285, 2015.

[85] W. A. Khan and J. Angeles, "The Kinetostatic Optimization of Robotic Manipulators: The Inverse and the Direct Problems," *Journal of Mechanical Design*, vol. 128, no. 1, pp. 168–178, 08 2005. [Online]. Available: https://doi.org/10.1115/1.2120808

[86] V. Kumar, "Characterization of Workspaces of Parallel Manipulators," *Journal of Mechanical Design*, vol. 114, no. 3, pp. 368–375, 09 1992. [Online]. Available: https://doi.org/10.1115/1.2926562

[87] E. Mirshekari, A. Ghanbarzadeh, and K. H. Shirazia, "Structure comparison and optimal design of 6-rus parallel manipulator based on kinematic and dynamic performances," *Latin American Journal of Solids and Structures*, vol. 13, pp. 2414–2438, 2016.

[88] J. P. Merlet, "Jacobian, Manipulability, Condition Number, and Accuracy of Parallel Robots," *Journal of Mechanical Design*, vol. 128, no. 1, pp. 199–206, 06 2005. [Online]. Available: https://doi.org/10.1115/1.2121740

[89] A. Fattah and A. Hasan Ghasemi, "Isotropic design of spatial parallel manipulators," *The International Journal of Robotics Research*, vol. 21, no. 9, pp. 811–824, 2002.

[90] J. Angeles, "Is there a characteristic length of a rigid-body displacement?" *Mechanism and Machine Theory*, vol. 41, no. 8, pp. 884–896, 2006.

[91] G. Pond and J. A. Carretero, "Formulating jacobian matrices for the dexterity analysis of

parallel manipulators," *Mechanism and Machine Theory*, vol. 41, no. 12, pp. 1505–1519, 2006.

[92] T. Yoshikawa, "Manipulability of robotic mechanisms," *The international journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[93] J. Wu, J. Wang, L. Wang, and Z. You, "Performance comparison of three planar 3-dof parallel manipulators with 4-rrr, 3-rrr and 2-rrr structures," *Mechatronics*, vol. 20, no. 4, pp. 510–517, 2010.

[94] C. Barus, "A treatise on the theory of screws. by sir robert stawell ball, ll. d., frs, lowndean professor of astronomy and geometry in the university of cambridge. cambridge, the university press; new york, the macmillan company. 1900. pp. xix+ 544, quarto." *Science*, vol. 12, no. 313, pp. 1001–1003, 1900.

[95] M. S. C. Yuan, F. Freudenstein, and L. S. Woo, "Kinematic Analysis of Spatial Mechanisms by Means of Screw Coordinates. Part 2â€"Analysis of Spatial Mechanisms," *Journal of Engineering for Industry*, vol. 93, no. 1, pp. 67–73, 02 1971. [Online]. Available: https://doi.org/10.1115/1.3427919

[96] G. Sutherland and B. Roth, "A Transmission Index for Spatial Mechanisms," *Journal of Engineering for Industry*, vol. 95, no. 2, pp. 589–597, 05 1973. [Online]. Available: https://doi.org/10.1115/1.3438195

[97] *The Transmissivity and Manipulability of Spatial Mechanisms*, ser. International Design Engineering Technical Conferences and Computers and Information in Engineering

Conference, vol. 22nd Biennial Mechanisms Conference: Flexible Mechanisms, Dynamics, and Analysis, 09 1992. [Online]. Available: https://doi.org/10.1115/DETC1992-0393

[98] Y. Takeda and H. Funabashi, "Motion transmissibility of in-parallel actuated manipulators," *JSME international journal. Ser. C, Dynamics, control, robotics, design and manufacturing*, vol. 38, no. 4, pp. 749–755, 1995.

[99] Y. Takeda, H. Funabashi, and H. Ichimaru, "Development of spatial in-parallel actuated manipulators with six degrees of freedom with high motion transmissibility," *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing*, vol. 40, no. 2, pp. 299–308, 1997.

[100] J. Brinker, B. Corves, and Y. Takeda, "On the motion/force transmissibility and constrainability of delta parallel robots," in *Computational kinematics*. Springer, 2018, pp. 340–348.

[101] M. Carricato and C. Gosselin, "On the modeling of leg constraints in the dynamic analysis of gough/stewart-type platforms," *Journal of computational and nonlinear dynamics*, vol. 4, no. 1, p. 011008, 2009.

[102] J. Sanjuan, E. Muñoz, M. Padilla, and M. Rahman, "The kinematic effects of simplifications in the analysis of linear translational parallel robots," *International Journal of Dynamics and Control*, pp. 1–18, 2022.

[103] Life Performance Research, "9-axis bluetooth imu lpms-b2 series - lp-research," https://lp-research.com/9-axis-bluetooth-imu-lpmsb2-series/, accessed: 2021-11-14.

[104] H. Al-Dois, A. Jha, and R. Mishra, "Task-based design optimization of serial robot manipulators," *Engineering Optimization*, vol. 45, no. 6, pp. 647–658, 2013.

[105] V. Gupta, S. K. Saha, and H. Chaudhary, "Optimum Design of Serial Robots," *Journal of Mechanical Design*, vol. 141, no. 8, 04 2019.

[106] P. Boscariol, R. Caracciolo, D. Richiedei, and A. Trevisani, "Energy optimization of functionally redundant robots through motion design," *Applied Sciences*, vol. 10, no. 9, p. 3022, 2020.

[107] J. D. Sanjuan De Caro, M. S. H. Sunny, E. Muñoz, J. Hernandez, A. Torres, B. Brahmi, I. Wang, J. Ghommam, and M. H. Rahman, "Evaluation of objective functions for the optimal design of an assistive robot," *Micromachines*, vol. 13, no. 12, p. 2206, 2022.

[108] S. Pedrammehr, S. Nahavandi, and H. Abdi, "Closed-form dynamics of a hexarot parallel manipulator by means of the principle of virtual work," pp. 883–895, 2018.

[109] J. Coopersmith, "The principle of virtual work," pp. 59–87, 2017.

[110] C. Ericson, *Real-time collision detection*. Crc Press, 2004.

[111] R. Martí, "Principles of scatter search," *Eur. J. Oper. Res.*, 2006.

[112] F. Berthaut, "A path relinking-based scatter search for the resource-constrained project scheduling problem," *International Journal of Project Organisation and Management*, 2018.

[113] A. Duarte, "Hybrid scatter tabu search for unconstrained global optimization," *Annals of Operations Research*, 2011.

[114] M. Gallego, "Hybrid heuristics for the maximum diversity problem," *Computational Optimization and Applications*, 2009.

[115] A. H. Fathaliyan, "Exploiting three-dimensional gaze tracking for action recognition during bimanual manipulation to enhance human–robot collaboration," *Frontiers in Robotics and AI*, 2018.

[116] M. Reke, "A self-driving car architecture in ros2," *2020 International SAUPEC/RobMech/PRASA Conference*, 2020.

[117] Y. Yang and T. Azumi, "Exploring real-time executor on ros 2," in *2020 IEEE International Conference on Embedded Software and Systems (ICESS)*. IEEE, 2020, pp. 1–8.

[118] L. Puck, P. Keller, T. Schnell, C. Plasberg, A. Tanev, G. Heppner, A. Roennau, and R. Dillmann, "Performance evaluation of real-time ros2 robotic control in a time-synchronized distributed network," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2021, pp. 1670–1676.

[119] "JACO," Mar. 2023, [Online; accessed 4. Jun. 2023]. [Online]. Available: https://robots.ros.org/jaco

[120] J. Kim, "Security and performance considerations in ros 2: A balancing act," *ArXiv*, 2018.

[121] Kinovarobotics, "kinova-ros," Jun. 2023, [Online; accessed 4. Jun. 2023]. [Online]. Available: https://github.com/Kinovarobotics/kinova-ros

[122] S. P. Thale, M. M. Prabhu, P. V. Thakur, and P. Kadam, "Ros based slam implementation for autonomous navigation using turtlebot," in *ITM Web of conferences*, vol. 32. EDP Sciences, 2020, p. 01011.

[123] M. Chapman, "SMA Patients' Lives Getting Better with Kinova's Jaco Robot," *SMA News Today*, Oct. 2017. [Online]. Available: https://smanewstoday.com/news/kinovas-jaco-robot-is-improving-lives-of-spinal-muscular-atrophy-patients

[124] I.-K. Jung and S. Lim, "An ethercat based control system for human-robot cooperation," in *2011 16th International Conference on Methods and Models in Automation and Robotics*, 2011, pp. 341–344.

[125] "E. t. group," [Online; accessed 4. Jun. 2023]. [Online]. Available: https://www.ethercat.org

[126] D. Orfanus, R. Indergaard, G. Prytz, and T. Wien, "Ethercat-based platform for distributed control in high-performance industrial applications," in *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*. IEEE, 2013, pp. 1–8.

[127] "soem," [, Online; accessed 4. Jun. 2023]. [Online]. Available: https://github.com/orocos/soem#Installation

[128] S. J. Kang, "A study on implementation of real-time ethercat master," *Journal of the Semiconductor & Display Technology*, vol. 20, no. 2, pp. 131–136, 2021.

[129] M. Cereia, I. C. Bertolotti, and S. Scanzio, "Performance of a real-time ethercat master

under linux," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 679–687, 2011.

[130] J. Liu, S. Chen, G. Zhang, and L. Shi, "The development of a novel servo motor controller based on ethercat and fpga," in *2016 Chinese Control and Decision Conference (CCDC)*. IEEE, 2016, pp. 3174–3179.

[131] Z. Yu, "An ethercat-based real-time control system design for a remote-direct-drive 2-dof manipulator," Ph.D. dissertation, Northeastern University, 2019.

[132] L. Shanshan, Z. Jian, and Y. Xiaoling, "Research on the automatic tuning servo system of short wave transmitter based on ethercat," in *2021 IEEE International Conference on Data Science and Computer Application (ICDSCA)*. IEEE, 2021, pp. 148–151.

[133] F. Sygulla, R. Wittmann, P. Seiwald, T. Berninger, A.-C. Hildebrandt, D. Wahrmann, and D. Rixen, "An ethercat-based real-time control system architecture for humanoid robots," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2018, pp. 483–490.

Appendix

Coefficients equation (5.20)

The following equations expresses the coefficients k_1 to k_4.

$$\begin{aligned}\kappa_1 = &-g\,\frac{2\,L_2\,m_1\,s\left(\theta_2\right)+4\,L_2\,m_2\,s\left(\theta_2\right)+4\,L_2\,m_3\,s\left(\theta_2\right)+2\,L_3\,m_2\,s\left(\theta_2+\theta_3\right)}{4}\\ &-g\,\frac{4\,L_3\,m_3\,s\left(\theta_2+\theta_3\right)+2\,L_4\,m_2\,s\left(\theta_2+\theta_3\right)+4\,L_4\,m_3\,s\left(\theta_2+\theta_3\right)}{4}\\ &-g\,\frac{L_5\,m_3\,c\left(\theta_2+\theta_3\right)\,s\left(\theta_4+\theta_5\right)+L_6\,m_3\,c\left(\theta_2+\theta_3\right)\,s\left(\theta_4+\theta_5\right)}{4}\\ &-g\,\frac{2\,L_5\,m_3\,s\left(\theta_2+\theta_3\right)\,c\left(\theta_5\right)+2\,L_6\,m_3\,s\left(\theta_2+\theta_3\right)\,c\left(\theta_5\right)}{4}\\ &+g\,\frac{L_5\,m_3\,s\left(\theta_4-\theta_5\right)\,c\left(\theta_2+\theta_3\right)-L_6\,m_3\,s\left(\theta_4-\theta_5\right)\,c\left(\theta_2+\theta_3\right)}{4}\end{aligned}$$

$$\begin{aligned}\kappa_2 = &-\frac{L_3\,g\,m_2\,s\left(\theta_2+\theta_3\right)}{2}-L_3\,g\,m_3\,s\left(\theta_2+\theta_3\right)\\ &-\frac{L_4\,g\,m_2\,s\left(\theta_2+\theta_3\right)}{2}-L_4\,g\,m_3\,s\left(\theta_2+\theta_3\right)\\ &-\frac{L_5\,g\,m_3\,s\left(\theta_2+\theta_3\right)\,c\left(\theta_5\right)}{2}-\frac{L_6\,g\,m_3\,s\left(\theta_2+\theta_3\right)\,c\left(\theta_5\right)}{2}\\ &-\frac{L_5\,g\,m_3\,c\left(\theta_2+\theta_3\right)\,c\left(\theta_4\right)\,s\left(\theta_5\right)}{2}-\frac{L_6\,g\,m_3\,c\left(\theta_2+\theta_3\right)\,c\left(\theta_4\right)\,s\left(\theta_5\right)}{2}\end{aligned}$$

$$\kappa_3 = \frac{g\,m_3\,s\left(\theta_2+\theta_3\right)\,s\left(\theta_4\right)\,s\left(\theta_5\right)\,\left(L_5+L_6\right)}{2}$$

$$\kappa_4 = -g\, m_3\, (L_5 + L_6)\, \frac{c\,(\theta_2)\; c\,(\theta_3)\; s\,(\theta_5) - s\,(\theta_2)\; s\,(\theta_3)\; s\,(\theta_5)}{2} - g\, m_3\, \frac{c\,(\theta_2)\; c\,(\theta_4)\; c\,(\theta_5)\; s\,(\theta_3) + c\,(\theta_3)\; c\,(\theta_4)\; c\,(\theta_5)\; s\,(\theta_2)}{2}$$

URDF of the Optimal Robot

This section presents the Unified Robot Description Format (URDF) of the optimal assistive robot. The file enumerates configurations of each link, along with their inertias and masses. These parameters were calculated to closely represent the properties of the actual materials, thereby accurately simulating the physical behavior of the real robot.

```
<?xml version="1.0" encoding="utf-8"?>
<robot
  name="MR2a">
  <link name="World"/>
  <link
    name="Link_1">
    <inertial>
      <origin
        xyz="0␣0␣0.0983"
        rpy="0␣0␣0" />
      <mass
        value="2.9232" />
      <inertia
        ixx="0.0182"
        ixy="0.0"
        ixz="0.0"
```

```
      iyy="0.0176"
      iyz="0.0"
      izz="0.0059" />
  </inertial>
  <visual>
    <origin
      xyz="0␣0␣0"
      rpy="0␣0␣0" />
    <geometry>
      <mesh
        filename="package://MR2a/meshes/Link_1.STL" />
    </geometry>
    <material
      name="">
      <color
        rgba="0.75294␣0.75294␣0.75294␣1" />
    </material>
  </visual>
</link>
<joint
    name="Joint_0"
    type="fixed">
    <origin
        xyz="0␣0␣0"
```

```
        rpy="0␣0␣0" />
    <parent
      link="World" />
    <child
      link="Link_1" />
</joint>
<link
  name="Link_2">
  <inertial>
    <origin
      xyz="0␣0.0256␣-0.0004"
      rpy="0␣0␣0" />
    <mass
      value="2.3234" />
    <inertia
      ixx="0.0032"
      ixy="0"
      ixz="0"
      iyy="0.0029"
      iyz="0.0"
      izz="0.0032" />
  </inertial>
  <visual>
    <origin
```

```
      xyz="0␣0␣0"
      rpy="0␣0␣0" />
    <geometry>
      <mesh
        filename="package://MR2a/meshes/Link_2.STL" />
    </geometry>
    <material
      name="">
      <color
        rgba="0.75294␣0.75294␣0.75294␣1" />
    </material>
  </visual>
</link>
<joint
  name="Joint_1"
  type="revolute">
  <origin
      xyz="0␣0␣0.267"
      rpy="0␣0␣-1.5708" />
  <parent
    link="Link_1" />
  <child
    link="Link_2" />
  <axis
```

```
    xyz="0␣0␣1" />
  <limit
    lower="-3.1416"
    upper="3.1416"
    effort="0"
    velocity="0" />
</joint>
<link
  name="Link_3">
  <inertial>
    <origin
      xyz="0.1117␣0.00007␣0.0877"
      rpy="0␣0␣0" />
    <mass
      value="0.8037" />
    <inertia
      ixx="0.0006"
      ixy="0"
      ixz="0"
      iyy="0.0092"
      iyz="0"
      izz="0.0097" />
  </inertial>
  <visual>
```

```
    <origin
      xyz="0␣0␣0"
      rpy="0␣0␣0" />
    <geometry>
      <mesh
        filename="package://MR2a/meshes/Link_3.STL" />
    </geometry>
    <material
      name="">
      <color
        rgba="0.75294␣0.75294␣0.75294␣1" />
    </material>
  </visual>
</link>
<joint
  name="Joint_2"
  type="revolute">
  <origin
    xyz="0␣0␣0"
    rpy="-1.5708␣-1.5708␣0" />
  <parent
    link="Link_2" />
  <child
    link="Link_3" />
```

```
  <axis
    xyz="0␣0␣1" />
  <limit
    lower="-2.5307"
    upper="2.5307"
    effort="0"
    velocity="0" />
</joint>
<link
  name="Link_4">
  <inertial>
    <origin
      xyz="0.0␣0.0948␣-0.0161"
      rpy="0␣0␣0" />
    <mass
      value="3.0775" />
    <inertia
      ixx="0.0389"
      ixy="0.0"
      ixz="0.0"
      iyy="0.0043"
      iyz="0.0047"
      izz="0.0377" />
  </inertial>
```

```
  <visual>
    <origin
      xyz="0␣0␣0"
      rpy="0␣0␣0" />
    <geometry>
      <mesh
        filename="package://MR2a/meshes/Link_4.STL" />
    </geometry>
    <material
      name="">
      <color
        rgba="0.75294␣0.75294␣0.75294␣1" />
    </material>
  </visual>
</link>
<joint
  name="Joint_3"
  type="revolute">
  <origin
    xyz="0.245␣0␣0"
    rpy="3.1416␣0␣1.5708" />
  <parent
    link="Link_3" />
  <child
```

```
    link="Link_4" />
  <axis
    xyz="0␣0␣1" />
  <limit
    lower="-2.7053"
    upper="2.7053"
    effort="0"
    velocity="0" />
</joint>
<link
  name="Link_5">
  <inertial>
    <origin
      xyz="0␣0.0053␣-0.0227"
      rpy="0␣0␣0" />
    <mass
      value="1.5108" />
    <inertia
      ixx="0.0046"
      ixy="0.0"
      ixz="0.0"
      iyy="0.0044"
      iyz="0.0001"
      izz="0.0018" />
```

```
  </inertial>
  <visual>
    <origin
      xyz="0␣0␣0"
      rpy="0␣0␣0" />
    <geometry>
      <mesh
        filename="package://MR2a/meshes/Link_5.STL" />
    </geometry>
    <material
      name="">
      <color
        rgba="0.75294␣0.75294␣0.75294␣1" />
    </material>
  </visual>
</link>
<joint
  name="Joint_4"
  type="revolute">
  <origin
    xyz="0␣0.3898␣0"
    rpy="-1.5708␣0␣0" />
  <parent
    link="Link_4" />
```

```
  <child
    link="Link_5" />
  <axis
    xyz="0␣0␣1" />
  <limit
    lower="-3.1416"
    upper="3.1416"
    effort="0"
    velocity="0" />
</joint>
<link
  name="Link_6">
  <inertial>
    <origin
      xyz="0.0001␣-0.1003␣-0.0064"
      rpy="0␣0␣0" />
    <mass
      value="1.1835" />
    <inertia
      ixx="0.0032"
      ixy="0.0"
      ixz="0.0"
      iyy="0.0013"
      iyz="0.0"
```

```
      izz="0.003" />
  </inertial>
  <visual>
    <origin
      xyz="0␣0␣0"
      rpy="0␣0␣0" />
    <geometry>
      <mesh
        filename="package://MR2a/meshes/Link_6.STL" />
    </geometry>
    <material
      name="">
      <color
        rgba="0.75294␣0.75294␣0.75294␣1" />
    </material>
  </visual>
</link>
<joint
  name="Joint_5"
  type="revolute">
  <origin
    xyz="0␣0␣0"
    rpy="-1.5708␣0␣0" />
  <parent
```

```
    link="Link_5" />
  <child
    link="Link_6" />
  <axis
    xyz="0␣0␣1" />
  <limit
    lower="-1.7453"
    upper="1.7453"
    effort="0"
    velocity="0" />
</joint>
<link
  name="Gripper">
  <inertial>
    <origin
      xyz="0.1150␣-0.0597␣-0.22"
      rpy="0␣0␣0" />
    <mass
      value="4" />
    <inertia
      ixx="0.0039"
      ixy="0.0"
      ixz="0.0"
      iyy="0.0032"
```

```
      iyz="0.0"
      izz="0.0018" />
  </inertial>
  <visual>
    <origin
      xyz="0 0 0"
      rpy="0 0 0" />
    <geometry>
      <mesh
        filename="package://MR2a/meshes/Gripper.STL" />
    </geometry>
    <material
      name="">
      <color
        rgba="0.79216 0.81961 0.93333 1" />
    </material>
  </visual>
</link>
<joint
  name="Joint_6"
  type="revolute">
  <origin
    xyz="0 -0.34813 0"
    rpy="1.5708 0 0" />
```

```
    <parent
      link="Link_6" />
    <child
      link="Gripper" />
    <axis
      xyz="0␣0␣1" />
    <limit
      lower="-3.1416"
      upper="3.1416"
      effort="0"
      velocity="0" />
  </joint>
</robot>
```