PolyPublie
Polytechnique Montréal

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Titre: Title:	Personalized Upper Limb Robotic Rehabilitation Using Impedance-Based Control Strategies
Auteur: Author:	Ayoub Behidj
Date:	2023
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Behidj, A. (2023). Personalized Upper Limb Robotic Rehabilitation Using Impedance-Based Control Strategies [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/57001/

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/57001/
Directeurs de recherche: Advisors:	Abolfazl Mohebbi, & Sofiane Achiche
Programme: Program:	Génie mécanique

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal

https://publications.polymtl.ca

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Personalized Upper Limb Robotic Rehabilitation using Impedance-based Control Strategies

AYOUB BEHIDJ
Département de génie mécanique

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie mécanique

Novembre 2023

© Ayoub Behidj, 2023.

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Personalized Upper Limb Robotic Rehabilitation using Impedance-based Control Strategies

présenté par **Ayoub BEHIDJ**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Luc BARON, président
Abolfazl MOHEBBI, membre et directeur de recherche
Sofiane ACHICHE, membre et codirecteur de recherche
Samuel KADOURY, membre

DEDICATION

To my beloved grandmother,
whose unwavering support and unconditional love have been the guiding light in my life.
Your faith in me has left a mark on my heart.
الله يرحمها

À ma chère grand-mère,
dont le soutien indéfectible et l'amour inconditionnel ont été le phare de ma vie.
Ta foi en moi a laissé une empreinte indélébile dans mon cœur.
الله يرحمها

ACKNOWLEDGEMENTS

First and foremost, my heartfelt appreciation goes to my research advisor, Dr. Abolfazl Mohebbi. During the two years under his guidance, he was much more than an academic mentor. His deep involvement, both academically and socially, was truly invaluable. His emphasis on fostering a healthy team dynamic and organizing out-of-lab events created an environment conducive to my growth and fulfillment. I am deeply grateful for his shared wisdom and the discipline he instilled in me.

I would also like to extend my deepest gratitude to my co-advisor, Dr. Sofiane Achiche. Beyond his academic prowess, his unique approach to problem-solving, paired with his patience and genuine commitment to my success, made my research journey both enlightening and rewarding. His guidance, advice, and kindness have not only been unwavering pillars throughout this academic pursuit but have also shaped my approach to research and collaboration.

A big thank you to my team, POLAR. Your energy and good spirits have been a continuous source of inspiration. Working alongside you has been an absolute privilege.

My gratitude also extends to the Arbour Foundation and the Natural Sciences and Engineering Research Council of Canada (NSERC) for their essential financial support.

To my parents, my sister, and all my loved ones, your unwavering support has been my anchor. Your love and belief in me have powered me through every step.

Lastly, I would like to thank the jury members, Dr. Luc Baraon and Dr. Samuel Kadoury, for their expertise and insightful feedback, which significantly enhanced the quality of this thesis.

RÉSUMÉ

Les membres supérieurs, essentiels pour de nombreuses activités quotidiennes, est souvent confrontée à des déficiences fonctionnelles, notamment chez les patients victimes d'AVC, en raison de troubles neurologiques. À mesure que les dispositifs robotiques deviennent plus courants dans le secteur de la santé, il est crucial d'améliorer leur contrôle, leur adaptabilité et leur synergie avec les systèmes de réalité virtuelle. Ce mémoire de maîtrise se concentre sur le développement de stratégies de contrôle adaptatives pour optimiser l'interaction entre ces manipulateurs robotiques et le membre affecté pendant les exercices de rééducation.

La méthodologie de recherche a impliqué l'intégration de manipulateurs robotiques, d'un système de support de bras et d'une interface de réalité virtuelle. Des stratégies de contrôle telles que le contrôle par admittance, par impédance, par impédance adaptative et des méthodes pour éviter les singularités ont été mises en œuvre et analysées. Plus précisément, le contrôleur par admittance a été développé pour générer des trajectoires de référence spécifiques au patient pour les exercices de rééducation, tandis que le contrôleur par impédance facilitait les exercices en permettant une certaine erreur de position, permettant aux patients d'exercer un effort lorsque cela était possible. Le contrôle par impédance adaptative, en modulant les paramètres d'impédance en fonction des performances du patient, permet une utilisation polyvalente du dispositif robotique sur un éventail d'exercices de rééducation, allant des exercices passifs aux activités de la vie quotidienne. Sur la base de la métrique de manipulabilité, un champ de force virtuel et une stratégie de contrôle par espace nul ont été mis en œuvre pour éviter les singularités. Cette étude a évalué des aspects tels que la génération de trajectoire, le suivi de trajectoire, et l'influence des paramètres d'impédance sur l'efficacité.

Les résultats suggèrent que la stratégie de contrôle par admittance est efficace pour générer des trajectoires de référence. De plus, le contrôle par impédance a révélé un compromis entre le suivi de trajectoire et la réactivité aux forces externes, influencé par les paramètres de raideur et d'amortissement. Le contrôle par impédance adaptative ajustait les paramètres de contrôle en fonction des performances de suivi de trajectoire de l'utilisateur. Ces résultats étaient comparables à ceux des contrôleurs non adaptatifs, mais avec trois avantages distincts : assurer la sécurité de l'utilisateur et de l'équipement en atténuant les réactions extrêmes, maintenir une performance constante malgré les changements ou perturbations imprévus, et moduler dynamiquement le niveau de support en fonction des performances du patient. L'utilisation de la force virtuelle et de la commande par espace nul a démontré leur efficacité

pour éviter les points de singularité.

En conclusion, cette thèse met en évidence le développement d'une plateforme robotique de rééducation entièrement personnalisée pour la rééducation du membre supérieur. Grâce à l'utilisation de techniques de contrôle adaptatif, elle propose des conceptions d'exercices sur mesure via le contrôle par admittance et assure un support adapté en temps réel durant l'exécution de l'exercice avec le dispositif robotique. Ce travail enrichit la compréhension fondamentale du contrôle robotique dans le domaine de la rééducation et présente une approche prometteuse pour une rééducation adaptée au patient.

ABSTRACT

The human upper extremity, essential for many daily activities, often faces functional impairments, especially among stroke patients, due to neurological disorders. As robotic devices become more prevalent in healthcare, enhancing their control, adaptability, and synergy with virtual reality systems becomes crucial. This master's thesis focuses on the development of adaptive control strategies that optimize the interaction between these robotic manipulators and the affected limb during rehabilitation exercises.

The research methodology involved the integration of robotic manipulators, an arm support system, and a virtual reality interface. Control strategies such as admittance, impedance, adaptive impedance, and methods to avoid singularities, were implemented and analyzed. Specifically, the admittance controller was developed to generate patient-specific reference trajectories for rehabilitation exercises, while the impedance controller facilitated exercises by permitting a certain positional error, allowing patients to exert effort when possible. The adaptive impedance control, by modulating the impedance parameters in response to patient performance, enables the robotic device's versatile use across a spectrum of rehabilitation exercises, from passive exercises to activities of daily living. Based on the manipulability metric, a virtual force field and a null-space control strategy were implemented to avoid singularities. This investigation evaluated aspects such as trajectory generation, trajectory tracking, and the impacts of impedance parameters on efficiency.

The findings suggest that the admittance control strategy is adept at producing reference trajectories. Moreover, impedance control highlighted a trade-off between trajectory tracking and compliance, influenced by the stiffness and damping model parameters. Adaptive impedance control adjusted control parameters based on the user's trajectory-following performance. These results were on par with non-adaptive controllers, but with three distinct advantages: ensuring user and equipment safety by mitigating extreme reactions, maintaining consistent performance despite unforeseen changes or disruptions, and dynamically modulating the level of support in relation to patient performance. The employment of virtual force and null-space command demonstrated efficacy in avoiding singularity configurations.

In conclusion, this thesis highlights the development of a fully personalized robotic rehabilitation platform for upper limb rehabilitation. Through the use of adaptive control techniques, it offers customized exercise designs via admittance control and ensures real-time adapted support during actual exercise performance with the robotic device. This work enriches the foundational understanding of robotic control within the rehabilitation domain and presents

a promising approach to patient-specific rehabilitation.

TABLE OF CONTENTS

DEDICATION iii

ACKNOWLEDGEMENTS iv

RÉSUMÉ v

ABSTRACT viii

TABLE OF CONTENTS ix

LIST OF TABLES xii

LIST OF FIGURES xiii

LIST OF SYMBOLS AND ACRONYMS xvii

LIST OF APPENDICES xviii

CHAPTER 1 INTRODUCTION 1

CHAPTER 2 LITERATURE REVIEW AND THEORETICAL BACKGROUND . 3

2.1 Upper limb impairment 3

2.2 Stroke 4

2.3 Post-stroke rehabilitation 5

2.3.1 Recovery stages 5

2.3.2 Post-Stroke Rehabilitation Methods 6

2.3.3 Classification of Therapeutic Methods 7

2.3.4 Continuity of Care and Remote Therapies 8

2.4 Virtual Reality in Rehabilitation 8

2.5 Robotic Rehabilitation Devices 11

2.5.1 Mechanical Design 11

2.5.2 Examples of Robotic Systems for Rehabilitation. 11

2.6 Control Strategies for Robotic Rehabilitation Devices 14

2.7 Adaptive Control 16

2.8 Robotics 18

2.8.1 Forward Kinematics of Serial Robotic Manipulator 18

2.9 Differential Kinematics and the Geometric Jacobian Matrix 21
2.9.1 Inverse Kinematics 23
2.9.2 Singularities in Robotic Manipulation 24
2.9.3 Manipulability Gradient 24
2.9.4 Handling Orientations in Robot Kinematics 25
2.9.5 Dynamic Model 26

CHAPTER 3 RESEARCH OBJECTIVES 29
3.1 Research Question 29
3.2 Hypotheses 29
3.3 Main Objective and Specific Objectives 29

CHAPTER 4 METHODOLOGY 30
4.1 Methodology Overview 30
4.2 Experimental Setup 30
4.2.1 Virtual Reality 30
4.2.2 Robotic Manipulator I: Kinova MICO^2 30
4.2.3 Robotic Manipulator II: UFactory xArm7 31
4.2.4 Arm Support 34
4.2.5 System Integration 35
4.3 Control Strategies 37
4.3.1 Dynamic Models 38
4.3.2 External Force Estimation 41
4.3.3 Admittance Control 42
4.3.4 Impedance Control 45
4.3.5 Adaptive Impedance Control 47
4.4 Cartesian Admittance and Impedance Control 51
4.4.1 Cartesian Admittance Control 53
4.4.2 Cartesian Impedance Control 53
4.4.3 Inverse Kinematics 54
4.5 Singularity Avoidance 56
4.5.1 Manipulability Gradient 57
4.5.2 Null-Space Control for Singularity Avoidance 59
4.5.3 Virtual Force/Torque Computation for Singularity Avoidance 60
4.6 Design of Experiments 61
4.6.1 Experiment 1 : Trajectory Generation using Cartesian Admittance Control 62

4.6.2 Experiment 2 : Effect of Impedance Parameters 63
4.6.3 Experiment 3 : Response to External Perturbations 63
4.6.4 Experiment 4 : Impedance Control Evaluation with and without Perturbations 64
4.6.5 Experiment 5 : Adaptive Control for Rehabilitation Purposes 65

CHAPTER 5 RESULTS AND DISCUSSION 67
5.1 Results 67
5.1.1 Experiment 1 : Trajectory Generation 67
5.1.2 Experiment 2: Impact of Impedance Parameters on Performance . . . 74
5.1.3 Experiment 3: Analyzing Response to External Perturbations 77
5.1.4 Experiment 4: Trajectory Tracking Using Cartesian Impedance Control 79
5.1.5 Experiment 5: Adaptive Impedance Control Rehabilitation Exercises 84
5.2 Discussion 90
5.2.1 Cartesian Admittance Control 90
5.2.2 Impedance Control: The Effect of Parameters 94
5.2.3 Cartesian Impedance Control: Trajectory Tracking 94
5.2.4 Adaptive Cartesian Impedance Control 95
5.2.5 Adaptive Cartesian Impedance Control: Advantages over Non-Adaptive Controller 97

CHAPTER 6 CONCLUSION 98
6.1 Conclusion 98

REFERENCES 100

APPENDICES 108

LIST OF TABLES

Table 2.1	Comparison of Exoskeleton-based and End-effector based Robotic Devices for Rehabilitation [1]	13
Table 4.1	Specifications of the UFactory xArm7 robotic arm	32
Table 4.2	UFACTORY xArm7-Model Dynamics [2]	32
Table 4.3	xArm7 DH parameters	33
Table 4.4	Axis Force Torque Sensor specifications	34
Table 4.5	Types of exercises and corresponding robot support	49
Table 4.6	Combinations of impedance parameters used during testing	64
Table 4.7	Impedance settings for response testing with external perturbations	64
Table A.1	VIVE Pro 2 specifications	109
Table B.1	Specifications of the Kinova MICO^2 6 DOF robotic arm	111
Table B.2	MICO 6 DOF curved wrist DH parameters	112
Table B.3	Transformation from DH algorithm to robotic arm physical angles	112

LIST OF FIGURES

Figure 2.1 Visual representation of strokes [3] 4
Figure 2.2 Brunnstrom Stages of Stroke Recovery. Left-hand side describes the motor control of the impaired limb and the right-hand side describes the type of recommended exercise 6
Figure 2.3 The different types of VR 10
Figure 2.4 Exoskeleton and End-Effector-Based Robotic Devices [4] 12
Figure 2.5 Illustrative examples of innovative robotic systems designed for rehabilitation purposes. Each subfigure represents a different system used in physical therapy and recovery 15
Figure 2.6 Impedance and Admittance Control Paradigms 16
Figure 2.7 Representation of the DH parameters between 3 joints 21
Figure 4.1 xArm7 robotic arm 31
Figure 4.2 Frames for xArm7 DH parameters. [2] 32
Figure 4.3 Modbus-TCP communication protocol for xArm7. [5] 33
Figure 4.4 Arm Support developed for the MICO 6 DOF 35
Figure 4.5 Adapted Arm Support for the xArm7 36
Figure 4.6 Experimental setup showing the VR, robotic device, and arm support 37
Figure 4.7 Use case representation of the robotic system for rehabilitation 39
Figure 4.8 Forward and Inverse Dynamics models inputs and outputs in joint and Cartesian space.Here, τ represents the joint torques, F denotes the wrench in Cartesian space, $\ddot{q}$ is the joint's acceleration, and $\ddot{X}$ is the end-effector's acceleration 39
Figure 4.9 Admittance and Impedance models inputs and outputs.Here, τ represents the joint torques, F denotes the wrench in Cartesian space, $\dot{q}$ is the joint's velocities, and $\dot{X}$ is the end-effector's velocity. The subscript d denotes a desired parameter. 43
Figure 4.10 Joint space admittance control using the gravity compensated interaction torques 46
Figure 4.11 Joint-space impedance control strategy 48
Figure 4.12 Graphical representation of position error in impedance control . . . 50
Figure 4.13 (a) adaptive law parameter for the translation DOF, (b) adaptive law parameter for the rotation DOF 52
Figure 4.14 Joint space Adaptive Impedance control 53

Figure 4.15 Cartesian admittance control using the force sensor 54
Figure 4.16 Cartesian Impedance control using the force sensor 55
Figure 4.17 Cartesian admittance control loop with singularity avoidance mechanisms framed by the red dashed boxes. Here : w is the manipulability, ∇w is the manipulability gradient, and F_v is the virtual force 57
Figure 4.18 Cartesian Impedance control with singularity avoidance mechanisms . 58
Figure 4.19 Visual representation of the MLP 59
Figure 4.20 Graphical representation of the function $k(w)$, showcasing the scaling behavior as the manipulability measure approaches critical singularity points. The regions corresponding to w_{cr} and w_{th} are distinctly highlighted 61
Figure 4.21 Shoulder joint passive exercises [6] 65
Figure 5.1 Generated trajectory using the Cartesian Admittance Controller . . . 67
Figure 5.2 Interaction forces during trajectory generation for different degrees of freedom (DOFs) 68
Figure 5.3 Force-Velocity Relationship in Cartesian Admittance Control: This figure illustrates the relation between external forces and velocities command during trajectory generation using the Cartesian admittance controller. The calculated slope, a fit through the data points, represents the compliance level of the admittance model 70
Figure 5.4 Analysis of manipulability and virtual force during robotic trajectory generation, highlighting the active mechanism of the robotic system to avoid singularity configurations 71
Figure 5.5 Velocity commands for each joint, comparing the null space velocity and the admittance joint velocity commands 73
Figure 5.6 Analysis of 4^{th} Joint Step Response: The figure displays the step response of the robot's 4^{th} joint, grouped into four categories based on shared response characteristics. These responses help understand the impact of different impedance control settings on the joint's movement behavior 75
Figure 5.7 Performance Metrics Overview: This figure summarizes the performance metrics, including Overshoot(%), Response Time(s), and Position Error(%), across different experimental conditions. The metrics, broken down by joint, draw attention to the relationship between impedance settings, particularly stiffness (K_d), and overall system behavior in task execution 76

Figure 5.8 Comparative Analysis of External Perturbation : This graph depicts the response of the position error under Cartesian impedance control when subjected to external perturbations, contrasting the responses between $Z_{d,low}$ and $Z_{d,high}$ settings 78
Figure 5.9 Tracking Performance in X and Pitch 80
Figure 5.10 Analysis of the control force and position error relationship during trajectory tracking with the Cartesian impedance controller under the $Z_{d,high}$ setting. **(a-d)**: Comparative insights under no perturbation and under external perturbations for both translational and orientational control. **(e-f)**: Control force with respect to the position error for the X direction and the Roll orientation 81
Figure 5.11 Tracking Performance in X and Pitch 82
Figure 5.12 Analysis of the command force and position error relationship during trajectory tracking with the Cartesian impedance controller under the $Z_{d,low}$ setting. **(a-d)**: Forces and error under no perturbation and under external perturbations for both translational and orientational control. **(e-f)**: Control force with respect to the position error for X translation and Roll orientation 83
Figure 5.13 Tracking Performance in X and Pitch 85
Figure 5.14 Analysis of the control force and position error relationship during trajectory tracking with the adaptive Cartesian impedance controller for passive rehabilitation exercises. **(a-d)**: Comparative insights between the adaptive parameters and passive parameters control for both translational and orientational control. **(e-f)**: Control force with respect to the position error relationship for adaptive parameters in passive exercises 86
Figure 5.15 Tracking Performance in X and Pitch 87
Figure 5.16 Analysis of the control force and position error relationship during trajectory tracking with the adaptive Cartesian impedance controller for ADL exercises. **(a-d)**: Comparison between the adaptive parameters and active parameters control for both translational and orientational control. **(e-f)**: Control force with respect to the position error relationship for adaptive parameters in ADL exercises 88
Figure 5.17 Analysis of the impact relationship between the command torque and the orientation error for different experimental setup. This figure presents the command torque from the Cartesian impedance controller using constant parameters-passive and ADL- and variable parameters . . . 89

Figure 5.18 Variable Impedance Control Parameters : this figure shows the variation of the impedance parameters throughout the shoulder passive exercise 91
Figure 5.19 Variable Impedance Control Parameters : this figure shows the variation of the impedance parameters throughout the *Can Off Shelf* exercise 92
Figure A.1 The VIVE Pro 2 Virtual Reality headset 109
Figure A.2 Robotic Rehabilitation Virtual Reality Environment 110
Figure B.1 Kinova's MICO^2 (left) and JACO^2 (right) 6 DOF robotic manipulator 111
Figure C.1 Tracking Performance of the Cartesian Impedance Controller 116
Figure C.2 Tracking Performance of the Cartesian Impedance Controller 117
Figure C.3 Relationship between position error and control force from the impedance controller. **(a)-(f)** Display the results with high impedance values (Z_{high}). **(g)-(l)** Show the results with low impedance values (Z_{low}) . . 118
Figure D.1 Tracking Performance of the Cartesian Impedance Controller 120
Figure D.2 Tracking Performance of the Cartesian Impedance Controller 121
Figure D.3 Relationship between position error and control force from the impedance controller. **(a)-(f)** Display the results for the passive shoulder flexion exercise. **(g)-(l)** Show the results for the Can Off Shelf exercise . . . 122
Figure D.4 Variable Impedance Control Parameters : this figure shows the variation of the impedance parameters throughout the shoulder passive exercise 123
Figure D.5 Variable Impedance Control Parameters : this figure shows the variation of the impedance parameters throughout the *Can Off Shelf* exercise124

LIST OF SYMBOLS AND ACRONYMS

ADL	Activites of Daily
UL	Upper Limb
VR	Virtual Reality
CNS	Central Nervous System
CVA	Cerebrovascular Accident

LIST OF APPENDICES

Appendix A Virtual Reality 108
Appendix B Kinova MICO^2 111
Appendix C Experiment 4: Trajectory Tracking Cartesian Impedance Control . . 115
Appendix D Experiment 5 : Adaptive Cartesian Impedance Control 119

CHAPTER 1 INTRODUCTION

The human upper extremity, including the arm, forearm, and hand, plays a crucial role in many daily activities, from basic tasks like eating and getting dressed to more complex actions like writing and using tools. However, many people face difficulties with upper-limb function due to neurological conditions, cardiovascular events, or injuries. These challenges have driven the fields of rehabilitation science and engineering to seek new methods to improve upper-limb functionality.

Recent years have seen the rise of robotic interventions, merging insights from biomechanics, neurology, and robotics [7]. The success of robot-assisted upper-limb rehabilitation is largely based on neuroplasticity, the brain's ability to form new neural connections in response to repetitive and systematic stimulation. Despite the potential advantages of robotic approaches, several obstacles remain. One major issue is the adaptation of robotic devices to patients' specific needs and progress, which requires personalized therapy plans [8,9]. The mechanical complexity of the human upper limb, with its multiple movement joints, makes this customization a challenging task [10].

Additionally, these systems need to provide realistic feedback to the user, ensuring the rehabilitation is not only effective mechanically but also supportive cognitively [11]. Safety is also a key concern. Robotic systems must interact safely with users, avoiding movements or forces that could cause harm [12, 13]. Furthermore, they should not be overly complex to understand or operate, allowing the user to focus on therapy rather than the technology itself [14].

In this broader scenario, stroke patients are a specific group who often experience severe forms of these general challenges due to damage in their central nervous system (CNS). Their recovery is particularly strenuous, with traditional methods requiring significant time and physical effort from both patients and therapists [15,16]. However, robotic rehabilitation devices currently available are not without limitations. They often have design restrictions that limit movement to specific joints, are complicated to use, and are expensive, making them less accessible [17, 18].

This master thesis introduces a new robotic platform designed for upper limb rehabilitation in patients with various neuromuscular deficiencies. The system aims for both versatility and cost-effectiveness to accommodate a wide range of needs. It features an end-effector type robotic device combined with a Virtual Reality (VR) system. The use of VR is critical due to its proven ability to establish realistic, engaging environments for patients, enhancing the

rehabilitation process [19].A distinguishing feature of this platform is its adaptive impedance control mechanism, seamlessly interfaced with the VR environment to facilitate targeted rehabilitative protocols [20, 21]. The control mechanism responds to the user's movements and provides appropriate resistance, promoting effective and responsive therapy sessions.

This design serves a broad spectrum of patients with neuromuscular challenges, needing upper limb therapy. Its adaptability and user-focused technology make it a valuable development in the field of remote and accessible rehabilitation solutions.

The rest of master this thesis is organized as follows: Chapter 2 begins with a systematic literature survey, providing an overview of the existing work in the field of robotic rehabilitation. This review helps identify the current gaps in research and areas needing further exploration.

In Chapter 3, the core objectives of this study are laid out, explaining the technical considerations that guide the development of the new rehabilitation platform. This explanation includes a detailed look at the specific challenges in robotic-assisted therapy and how this work aims to address them.

Chapter 4 covers the methodology employed in this research. It discusses the hardware selection process, the design of the algorithms used, and the configuration of the experiments conducted. This chapter ensures a clear understanding of the practical steps taken during the development phase of the platform.

Chapters 5 and 6 present the results of the experiments and a discussion of these outcomes, respectively. These sections analyze the data collected, evaluate the performance of the system, and discuss its implications, particularly concerning its effectiveness and potential improvements.

Finally, Chapter 7 concludes the thesis, summarizing the main findings and highlighting the contributions this research has made to the field of robotic-assisted rehabilitation. It also suggests potential avenues for future research, considering the broader impacts of this work on both the scientific community and everyday rehabilitation practices.

The project carries significant practical value beyond its scholarly contributions. The intended robotic system is designed to improve the efficiency of resource use in healthcare environments, increase the reliability of therapy, and broaden access to advanced rehabilitation methods. This research represents a key intersection of fundamental mechanical engineering principles with immediate healthcare needs, underscoring the importance of interdisciplinary approaches in engineering.

CHAPTER 2 LITERATURE REVIEW AND THEORETICAL BACKGROUND

2.1 Upper limb impairment

To design efficient rehabilitation strategies, it is crucial to understand the specifics of different impairments. Impairments fall into two primary categories. The first includes abnormalities or losses in bodily functions, particularly those related to the neuromuscular and movement systems. These issues translate into reduced joint mobility, diminished muscle strength, abnormal muscle tone, or the occurrence of involuntary movements. The second category involves structural impairments, identified by noticeable anomalies within the nervous system or in structures related to movement, such as the arms or the hands. [22, 23]

Neuromotor or neuromuscular disorders includes a range of conditions that impact muscles and/or the nerves controlling them. These disorders can lead to impairments in the upper limbs, compromising the movement of arms, hands, fingers, and shoulders. Examples of such diseases include:

- **Amyotrophic Lateral Sclerosis (ALS)**, a condition impacting the motor neurons controlling voluntary muscle movement. This disease leads to muscle weakness and atrophy, affecting various parts of the body, including the limbs.
- **Muscular Dystrophies**, a group of genetic disorders characterized by progressive muscle weakness and loss of muscle mass. Specific types, like Duchenne and Becker muscular dystrophies, can impair upper limb function over time.
- **Multiple Sclerosis (MS)**, which may cause muscle weakness, spasms, and coordination problems, potentially affecting the functionality of the upper limbs.
- **Spinal Muscular Atrophy (SMA)**, a genetic condition causing muscle weakness and atrophy, particularly in the skeletal muscles, that can affect the arms and hands.
- **Cerebral Palsy**, a group of neurological disorders that affect movement, muscle tone, or posture. It can lead to upper limb impairments, among other motor challenges [24].
- **Stroke**, potentially resulting in hemiplegia (paralysis of one side of the body) or hemiparesis (weakness on one side of the body), causing impairments in the upper limb on the affected side.

Each of these conditions manifests distinct types and degrees of upper limb impairments. Consequently, management and rehabilitation approaches must be adapted to the specific challenges and characteristics of the individual patient. While these disorders present a broad field of study, this work particularly concentrates on replicating physical therapies specific to stroke rehabilitation using robotic intervention programs. Subsequent sections will provide a detailed exploration of stroke from a rehabilitation engineering perspective.

2.2 Stroke

A cerebrovascular accident (CVA), or stroke, is a primary cause of movement disorders in the upper limb (UL). Every year, over 700,000 people experience a stroke in the United States alone, with 75% of these incidents representing new cases [25]. Additionally, it is the leading cause of long-term disability [26]. Studies show that 30-60% of stroke survivors have difficulties using their affected arm [27]. Many stroke survivors encounter challenges performing basic daily activities due to post-stroke outcomes. Research demonstrates that approximately 20% of stroke patients regain full motor function within 9 weeks, depending on the impairment's severity [28].

A cerebrovascular accident occurs when the brain's blood supply, or a part of it, is disrupted. In the absence of vital nutrients and oxygen, brain cells rapidly begin to die. The specific type of stroke is determined by the cause of this reduced blood supply; ischemic strokes are due to blocked or narrowed arteries supplying the brain tissues, while hemorrhagic strokes occur when blood vessels are damaged, leading them to leak or burst. Ischemic strokes are more prevalent [29]. Figure 2.1 provides a visual representation of both types of stroke. A less common form of stroke, called a transient ischemic attack (TIA), is caused by temporary disruption of the brain's blood flow and results in temporary symptoms. [30]

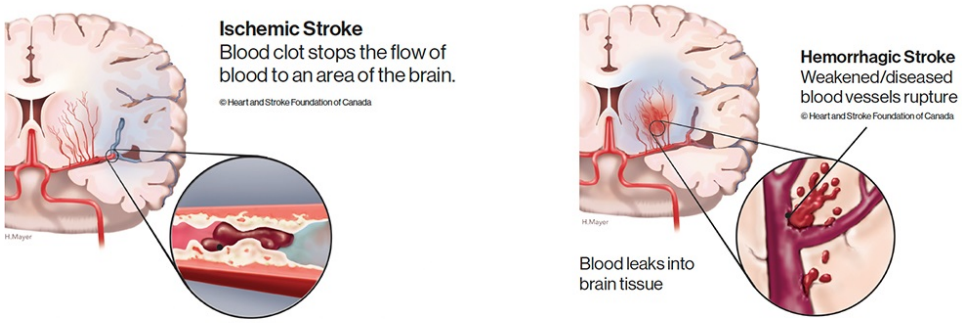


Figure 2.1 Visual representation of strokes [3]

Upper limb impairment occurs when dying brain cells damage the areas of the brain responsible for limb movement and sensation, specifically the motor cortex in the frontal lobe. Stroke can lead to various types of impairment. Hemiparesis, also known as weakness or paralysis, results in the inability to move the arm. It's important to note that if the stroke occurs on one side of the brain, the opposite limb is affected. Spasticity is another common post-stroke disability, which leads to involuntary muscle contractions, causing rigidity during certain movements. This condition limits the stroke survivor's control over limb movements by inducing unwanted movements. It is important to note that stroke can cause both hemiparesis and spasticity.

2.3 Post-stroke rehabilitation

Rehabilitation treatments are rapidly given to stroke survivors to aid recovery and improve their quality of life. In fact, post-stroke patients see their ability to tend to basic self-care tasks greatly reduced and cannot be discharged from the hospital without going through therapy. The therapists work with the patients towards increasing independence for self-care by setting and updating realistic and attainable goals.

Two main principals justify the importance of post-stroke therapy and dictate the way the therapy is structured. The first one is the changing pattern of brain activation. The adult central nervous system is plastic, meaning that it can restore deteriorated cognitive and motor functions by re-organizing it's structure. By changing behavior due to the loss neural control, by practicing and learning, regional plasticity of the neural structure associated to actions and thoughts can be observed. The second principal states that for continued motor and functional gains, at any time following the stroke, progressive motor practice is necessary. The effectiveness of the treatment is proportional to the brain's engaged attention, motivation and learning networks associated to the desired learning task. [31–34]

2.3.1 Recovery stages

The recovery process following a stroke is divided into multiple phases, each requiring a different approach in terms of rehabilitation. Early intervention is crucial as it can significantly improve the outcomes of the rehabilitation process [26].

Brunnstrom stages of recovery (BSSR)

Following a stroke, stroke patients with one or both arms impaired will follow certain stages of recovery. The Brunnstrom Stages of Stroke Recovery (BSSR) is a reference for post stroke

recovery. Figure 2.2 illustrates the BSSR, where, for each given stage of post stroke recovery, the left hand-side describes the motor control of the impaired limb and the right-hand side describes the type of recommended exercise. This concept highlights the importance of the progressive increase and adaptation of the exercises throughout the rehabilitation process. [35]

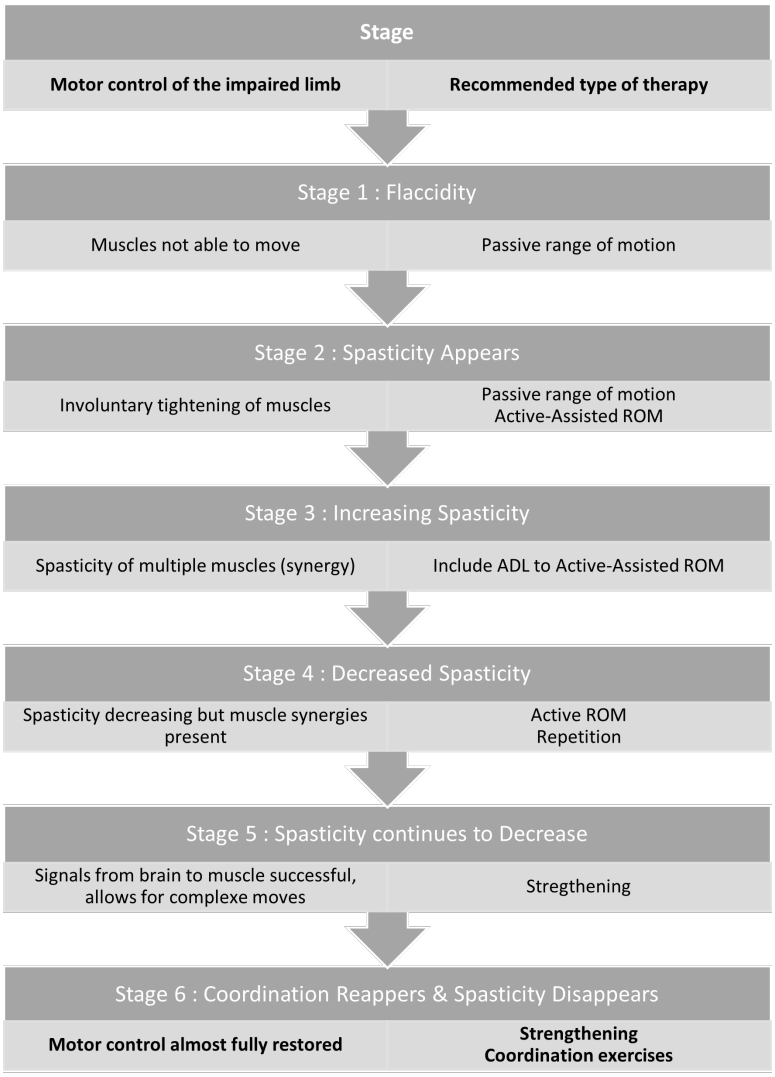


Figure 2.2 Brunnstrom Stages of Stroke Recovery. Left-hand side describes the motor control of the impaired limb and the right-hand side describes the type of recommended exercise

2.3.2 Post-Stroke Rehabilitation Methods

After a stroke, the necessity for consistent medical care for patients is highlighted. Conventional rehabilitation techniques, while effective, require substantial manual intervention from therapists [36]. The current trend is pivoting towards an analytical assessment of residual

motor capabilities, intending to provide an objective evaluation of treatment effectiveness, be it rehabilitative or pharmacological. This systematic approach may lead to enhanced motor recovery and consequently, augment the probability of patients regaining independence. The ultimate objective is to expedite the reintegration of patients into their societal and occupational roles, potentially leading to a reduction in long-term healthcare expenditures [37].

Customized therapeutic strategies are crucial. For patients with pronounced deficits, therapies predominantly focus on activating proximal muscular groups. Conversely, those with minimal impairments are steered towards regaining fine motor skills. Various techniques, encompassing the facilitation method, intensified physical therapy, electrical stimulations, and passive manipulations, have been employed for post-stroke rehabilitation [38]. Notably, the integration of robotic and mechatronic devices in sensory-motor rehabilitation is gaining prominence. When complemented with conventional therapeutic methods, this combination showcases enhanced motor outcomes, decreased rehabilitation periods, and provides objective evaluation metrics. [39, 40]

2.3.3 Classification of Therapeutic Methods

Selecting an appropriate therapeutic intervention is contingent upon various parameters, including the extent of limb impairment and the current phase of the patient's recovery. Instruments designed for upper limb rehabilitation can offer distinct modalities of motion assistance, including active, passive, haptic, and coaching interventions [41].

Passive Intervention

Within passive interventions, the patient remains inactive, with the motion executed solely by the robot or the therapist. Such interventions are typically employed during initial rehabilitation phases, especially when the patient's capacity to exert meaningful effort is limited. The assisting apparatus ensures completion of the designated exercises without active patient involvement. A prevalent instance of this therapeutic approach is the Range of Motion (ROM) exercises, which concentrate on maneuvering the limb within predefined motion parameters [42, 43].

Active Intervention

Active interventions are recommended for patients who retain some degree of movement capability in their impaired limb. This can be further dived into the following:

- **Active-Assistive Intervention:** Here, an external force—exerted by either the therapist or a robotic apparatus—assists the patient in achieving the assigned task [44]. This modality seeks to augment the motion range and has proven beneficial for patients with compromised shoulder and elbow functions. The integrated robotic system provides corrective actions if the patient's performance deviates from the intended motion [45].

- **Active-Resistive Intervention:** This approach involves presenting a counteracting force to the affected limb, delivered either manually by the therapist or mechanically by a robot. This counterforce can be systematically escalated throughout the rehabilitation, adapting in tandem with the patient's recuperative trajectory. Consequently, the patient exerts augmented effort, intensifying muscle activation. Isometric exercises exemplify Active-Resistive interventions. Here, a resistive force is engaged against the patient's limb, but muscular elongation remains unchanged, resulting in a static limb posture without any discernible movement [46–48].

- **Bilateral Intervention:** Bilateral interventions leverage the principle of motion mirroring in rehabilitation, whereby the compromised limb replicates the actions of the functional limb. This methodology grants the patient complete agency over the impaired limb. Robotic exoskeletons, such as MIME, and other mechanical devices deploy this therapeutic approach. A research study comprising fourteen participants manifested noteworthy enhancements in both the impaired hemisphere and overarching motor functionalities [49].

This work will focus on passive and active-assistive exercises for upper limb rehabilitation.

2.3.4 Continuity of Care and Remote Therapies

Maintaining continuous rehabilitation is vital, especially when patients transition between medical facilities or following their discharge. This is imperative to ensure an uninterrupted recovery process. Nonetheless, traditional healthcare systems occasionally face challenges in sustaining this continuity, largely attributed to resource constraints and time limitations.

2.4 Virtual Reality in Rehabilitation

Virtual reality (VR) provides a computer-simulated environment, allowing users to immerse themselves in a three-dimensional setting that can span a full 360 degrees. Depending on

the system's sophistication, VR immersion varies and can be classified into three primary categories:

1. **Non-Immersive VR:** A basic level of VR primarily seen in daily applications, non-immersive VR primarily utilizes screens and basic interaction tools such as computers, consoles, and peripheral input devices. While it furnishes a computerized environment, it permits users to remain connected to their physical surroundings.

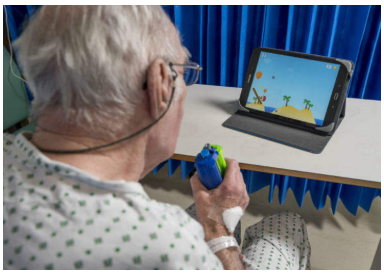
2. **Semi-Immersive VR:** Providing an intermediary level of immersion, semi-immersive VR uses devices like panoramic televisions or projection systems. It presents a balance, allowing users to engage with a digital setting, yet retain a link to their physical environment. These systems, often employed in educational or training contexts, are bolstered by high-resolution displays, powerful computational resources, and partial real-world mechanism replicas.

3. **Fully Immersive VR:** Representing the pinnacle of VR immersion, this type relies on Head Mounted Displays (HMDs) for interaction. Delivering the most authentic simulation experience, it integrates stereoscopic 3D effects, requiring specific VR glasses or HMDs combined with input tracking mechanisms. While primarily associated with entertainment, its applicability extends to other domains such as educational settings.

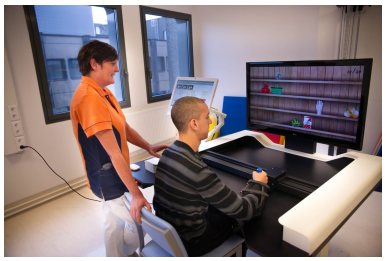
In summary, the scope of VR systems ranges from non-immersive to fully immersive setups, each offering distinct interaction experiences. Depending on their design and capability, these systems find applications in domains from entertainment to pedagogy [50–52].

For post-stroke upper-limb rehabilitation, Virtual Reality (VR) approaches can be divided into two categories: those custom-designed for rehabilitation purposes and commercially available systems that have been repurposed for therapy. The former, referred to as "custom VR," are developed with therapeutic goals in mind, while the latter, known as "commercial VR," are off-the-shelf solutions not initially intended for medical use [51]. Maier et al. conducted a detailed meta-analysis to evaluate the effectiveness of these VR categories in aiding motor recovery following a stroke. This comprehensive review included thirty studies involving 1473 patients, indicating that custom VR had a more pronounced impact on bodily functions, making it more effective than conventional therapy methods. However, commercial VR did not demonstrate significant benefits. Additionally, the analysis identified six neurorehabilitation principles that potentially contribute to the effectiveness of custom VR over both standard practices and commercial VR. Interestingly, more than half of the studies on custom VR incorporated strategies like offering both indirect and direct feedback, adjusting

difficulty to match patient progress, employing varied training routines, encouraging the use of the affected limb, and focusing on task-oriented activities [53].



(a) Non-immersive VR [54]



(b) Semi-immersive VR [55]



(c) Fully-immersive VR [56]

Figure 2.3 The different types of VR

Feedback is a vital component of VR systems in rehabilitation applications. This feedback can be provided through visual, auditory, or haptic mechanisms and serves to give real-time data on performance and progression [51]. Emphasizing the relevance of VR-based feedback, Kiper et al. conducted a randomized controlled trial involving 136 post-stroke patients, which included both ischemic (n=78) and hemorrhagic strokes (n=58). The study compared the effectiveness of reinforced feedback in a virtual environment (RFVE) combined with conventional rehabilitation (CR) against traditional rehabilitation treatments. The results indicated that patients subjected to the RFVE and CR combination showed improved results in upper limb recovery metrics compared to the control group [57].

VR systems, with their enhanced virtual capabilities, offer a more engaging experience for patients, potentially increasing motivation during the rehabilitation process. Supporting this notion, a review by Kang et al. (2020) on the role of VR in stroke rehabilitation suggested that VR-based rehabilitation can provide moderate improvements in functional outcomes compared to traditional therapy methods. When VR was added as an adjunct treatment to standard care, it demonstrated positive effects on the function of hemiparetic upper limbs and daily activities. These findings indicate that the interactive nature of VR can make the rehabilitation process more engaging and may lead to better outcomes. However, the integration of VR in clinical settings is not without challenges, including technical issues, patient-related concerns such as motion sickness, and the ongoing need for more research to determine the appropriate dosage and intensity of VR therapies [58].

2.5 Robotic Rehabilitation Devices

The landscape of post-stroke care is evolving with the introduction of robotic rehabilitation devices. Emulating exercises typically performed in traditional physical therapy, these devices add value by capturing metrics like movement speed, direction, and strength. By blending facilitation and motor learning techniques, they interactively gauge and guide patient movements. The trajectory for each motor task is precise, ensuring that every therapy session is uniform yet adaptable, simultaneously gathering essential data to fine-tune the recovery process.

The consensus around intensive motion training post-stroke is solidifying [59]. Robotic systems promise to deliver this intensity but in a more streamlined manner [60]. Some of these robotic interventions have showcased results on par with age-old methods [61], highlighting their efficiency in providing high-intensity therapy without draining resources [62]. Passive movements, though beneficial, don't guarantee comprehensive motor recovery [63]. A patient's active involvement is the key [64]. Modern robotic systems cater to this by tuning themselves to patients' specific requirements and monitoring their motion intentions [65–67] The final aim is clear: elevate therapy efficiency, boost patient recovery rates, and utilize resources judiciously.

2.5.1 Mechanical Design

In the field of robotic rehabilitation, 2 main type of devices are used : the end-effector-based and the exoskeleton-based devices, shown on Figure 2.4 [41,68]. These two mechanical designs differ in the way the movement is transferred from the device to the user. Exoskeletons are defined as a wearable mechanical structure that aims to reproduce the limb's morphology and provides assistance by applying a torque on the human joints. This is done by having rigid links covering the limb and motors, attached to the rigid links, that provides the assistance force. On the other hand, the end-effector based robots are attached to the impaired limb with the use of an interface and provides the assistive effort by applying the appropriate force where the limb is attached. 2.1 shows the pros and cons of the both devices. [1]

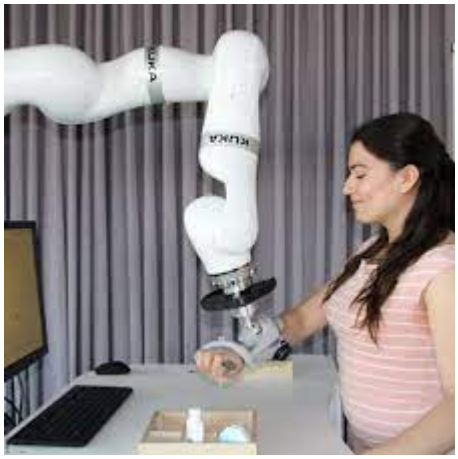
2.5.2 Examples of Robotic Systems for Rehabilitation.

REHAROB Therapeutic System

The REHAROB Therapeutic System, engineered for rehabilitating hemiparetic extremities, addresses conditions such as spastic hemiparesis resulting from neurological incidents. It



(a) Exoskeleton Robot [69]



(b) End-Effector Rehabilitation Robot

Figure 2.4 Exoskeleton and End-Effector-Based Robotic Devices [4]

employs two standard industrial robots, the IRB 140 and IRB 1400H, for coordinated upper and lower arm movements, providing seven degrees of freedom to facilitate complex shoulder and elbow exercises.

Notably, REHAROB's utilization of standard industrial robots not only contributes to cost-effectiveness but also expedites the medical device certification process. The system includes an adjustable powered couch for patient comfort and optimal positioning. Therapeutically, it implements passive, controlled movements as per a physical therapist's plan, aiming for outcomes like enhanced range of motion and reduced muscle hypertonia. Its design and therapeutic protocols have undergone stringent clinical validation per the European Medical Device Directive. [70]

ARAMIS Rehabilitation System

The ARAMIS Rehabilitation System is an innovative platform for recuperating upper limb function post-neurological events, utilizing a unique dual exoskeleton structure. It combines biomechanical movement with cognitive stimuli, featuring high-performance components for efficient torque delivery and precision in motion tracking.

ARAMIS is built around a sophisticated motion control architecture, including custom management software integrated into the Aramis Framework. Its therapeutic modalities encompass asynchronous, synchronous, and virtual exercises, each designed for specific rehabilitation phases and goals.To evaluate the impact of the ARAMIS rehabilitation protocol, a set of standardized scales was utilized: the Fugl-Meyer scale for upper limbs, the FIM (Functional

Table 2.1 Comparison of Exoskeleton-based and End-effector based Robotic Devices for Rehabilitation [1]

	Exoskeleton-based Robotic Devices	**End-effector based Robotic Devices**
Pros	• Anatomically aligned movement • Accurate feedback • Assist-as-needed support	• Simplicity and cost-effectiveness • Versatility • Ease of use
Cons	• Complexity and cost • Limited degrees of freedom • One-size-fits-all problem	• Less anatomically aligned • Less accurate feedback • Less stability

Independence Measure), and the Motricity Index for Upper Extremity. Comparing baseline to end-of-treatment scores, the Fugl-Meyer total score showed an impressive enhancement, rising from 48±18 to 75±27, indicating a mean improvement of 56%. Similarly, the pain score ameliorated from 4.5±2 to 7±1.2, while the overall motor function progressed from 11.7±10 to 27.5±17.4. Furthermore, the FIM total score witnessed a significant boost, ascending from 65±21 at baseline to 94±14 by the treatment's conclusion. These empirical metrics underline the ARAMIS system's effectiveness in advancing motor functionality, reducing pain, and bolstering overall motor skills. [71]

NJIT-RAVR System for Pediatric Rehabilitation

The NJIT-RAVR system adopts a holistic approach in pediatric rehabilitation, targeting upper extremity impairments in children with cerebral palsy through a fusion of adaptive robotics and virtual reality. Central to the system is the Haptic Master robot, facilitating interaction with virtual scenarios while providing haptic feedback.

The system boasts a range of therapeutic simulations, encouraging patient engagement and motivation. With adaptive algorithms, it ensures exercises maintain an appropriate level of challenge based on the child's capabilities. Pilot studies indicate the NJIT-RAVR's potential in enhancing motor function, as evidenced by objective kinematic data and clinical assessments. [72]

Intelligent Physiotherapy Assistant for Movement (iPAM) in Rehabilitation

The iPAM system serves as an advanced solution for upper limb rehabilitation in patients with movement impairments. It features robot orthoses and sophisticated sensors for precise movement analysis, controlled via a LabVIEW Real-time interface. [73]

iPAM's pneumatic actuators and servo-valves facilitate smooth, low-resistance exercises, while its computational model negates gravitational effects, ensuring user comfort. It operates under therapist supervision, recording detailed movement data for analysis and insight, thereby refining passive arm movement therapies. Overall, these robotic systems illustrate the evolution of rehabilitative therapies, emphasizing patient-specific, data-driven, and interactive approaches to recovery. [73]

2.6 Control Strategies for Robotic Rehabilitation Devices

Robotic devices in physical rehabilitation play a pivotal role. However, it is the underlying control strategies that dictate the behavior of these devices, making them an integral aspect of the design process.

For robotic-assisted rehabilitation, the expectation is that these devices emulate, if not entirely, then at least in part, the functions of a physical therapist. Critical specifications of these control strategies include the robot being in direct or indirect contact with the patient's limb, its capability to exert specific forces, and its adaptability to the user's motions. These specifications underpin the control strategy and determine the robot's applicability in rehabilitation scenarios [74].

Traditional position control prioritizes achieving a designated position, often neglecting the forces arising from interactions. This oversight can be detrimental, especially when the robot is in a dynamic environment, or when force considerations are paramount. Safety is paramount when the robot interacts with users; hence, the interaction forces between the rehabilitation device and the impaired limb must be meticulously regulated to ensure a secure yet effective experience. This scrutiny renders simplistic position control strategies ineffective.

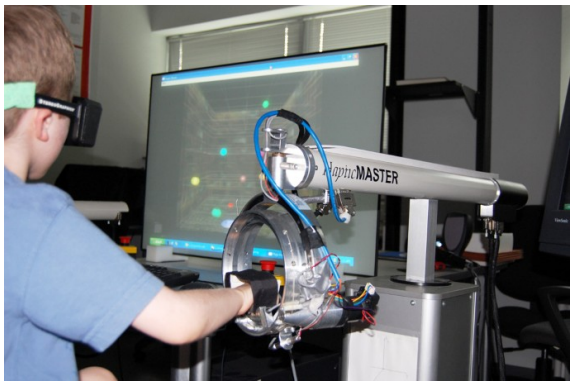
Force control strategies offer a framework where the robot's exerted forces on its environment can be meticulously defined—a requisite when the robot actively interacts with its surroundings. Impedance control, a subset of force control models, delineates the interaction forces between the robot and its environment based on the robot's current states, including position, velocity, and acceleration. Given its suitability in fragile, unpredictable, or evolving environments, impedance control finds extensive application in upper limb rehabilitation,



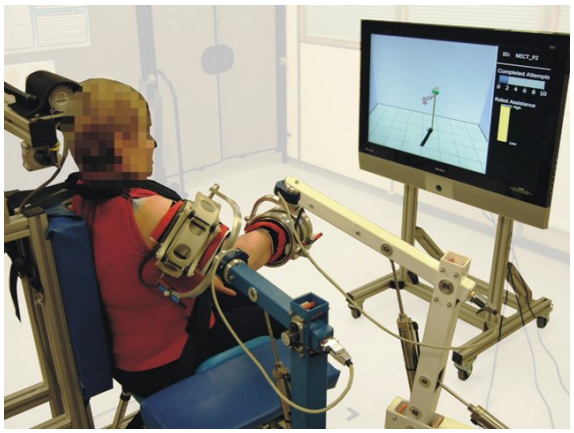
(a) REHAROB Therapeutic System. [70]



(b) ARAMIS Rehabilitation System. [71]



(c) NJIT-RAVR System. [72]



(d) iPAM System. [73]

Figure 2.5 Illustrative examples of innovative robotic systems designed for rehabilitation purposes. Each subfigure represents a different system used in physical therapy and recovery

ensuring safe interactions with patients.

As its nomenclature suggests, impedance control relates the robot's state to the applied force through a mechanical impedance model. Mechanical impedance quantifies the resistance encountered when mechanical energy traverses a system, and can be envisaged as a force or moment per unit of velocity or acceleration. Characterized by three parameters—inertia (M), damping (D), and stiffness (K)—the system's behavior varies based on these factors. A high system impedance renders the system "hard" upon contact, while a low impedance results in a "soft" feel.

While impedance-based models utilize desired motion to determine resultant force, admittance control operates in reverse. It leverages external force as an input to ascertain the robot's subsequent motion, as depicted in figure 2.6.

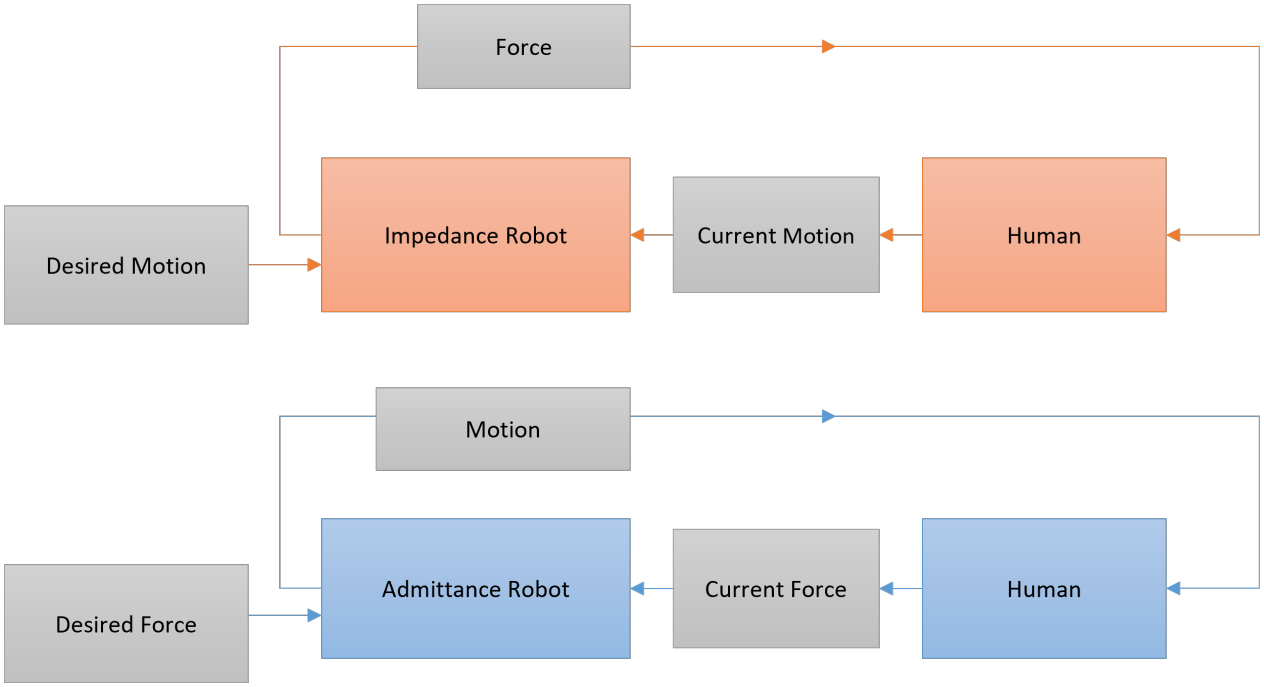


Figure 2.6 Impedance and Admittance Control Paradigms

2.7 Adaptive Control

Adaptive laws are instrumental in adjusting the parameters of impedance controllers. As previously delineated, the controller's performance is closely tied to parameter values, underscoring the significance of judicious parameter selection. In the realm of therapy, the demands placed on robotic devices are not static but shift based on exercise types and the

evolving recovery status of the patient. Consequently, the imperative lies in continually recalibrating the controller's parameters to suit the given scenario. Automating this recalibration becomes pivotal given the unpredictable nature of patient performance. Manual continual adjustments not only lack feasibility but also pose a substantial roadblock to the broader acceptance of rehabilitation robots.

Clinical metrics, such as the Fugl-Meyer Assessment and the Functional Independence Measure, coupled with sensory data like kinematics, dynamics, and task-specific metrics, offer insights into a patient's performance and recovery trajectory throughout rehabilitation. Given the real-time nature of adaptive laws during therapy sessions, such sensory data typically steer the algorithms.

In the rehabilitation context, the quintessential behavior expected of a robotic tool is "Assisted-As-Needed." This philosophy dictates that the rehabilitation device exerts only the requisite force on the patient's limb to navigate a specified trajectory.

1. **Active Training Based on Impedance Control by Liang Peng et al. [75]**

    - *Study Focus:* Implementation of active training in a 2-DOF upper-limb rehab robot primarily for post-stroke patients.
    - *Control Strategy:* Impedance control that allows compliant human-robot interactions, adjusting the robot's motion according to the deviation from a pre-set trajectory.
    - *Unique Features:* Integration with a virtual reality game to foster patient engagement and enhance motor recovery.
    - *Results:* Minimal error in robot's force feedback, indicating high accuracy in force adjustments during rehab sessions.

2. **Backstepping Adaptive Impedance Control by Jing Bai et al. [76]**

    - *Study Focus:* Introduction of backstepping adaptive impedance control for upper limb rehab robots tailored for home-based therapies.
    - *Control Strategy:* Combines backstepping control, adaptive control, and fuzzy logic to bolster robot stability and optimize trajectory tracking.
    - *Unique Features:* Implementation of a virtual rehabilitation training environment for increased patient enthusiasm and precise therapeutic outcomes.
    - *Results:* Compared to traditional methods, smoother trajectories were achieved, resulting in enhanced system robustness and adaptability.

3. **Adaptive Impedance Control Using Recurrent fuzzy Neural Network by Guozheng Xu [77]**

    - *Study Focus:* Development of an adaptive impedance controller for upper-limb rehab robots, catering to the physical recovery condition of the impaired limb.
    - *Control Strategy:* Uses an evolutionary dynamic recurrent fuzzy neural network optimized through hybrid learning algorithms, to modulate robot's desired impedance dynamically.
    - *Unique Features:* Incorporation of the slide average least squares identification algorithm to deduce stiffness and damping parameters of the impaired limb.
    - *Results:* Proficient adaptation to the physical condition of the impaired limb, offering a personalized and effective rehabilitation process.

4. **Safety Supervisory Strategy Based on Impedance Control by Pan et al. [78]**

    - *Study Focus:* Integration of impedance control in upper-limb rehab robots to ensure safe and effective human-machine interactions.
    - *Control Strategy:* Impedance control combined with a Safety Supervisory Fuzzy Controller for increased robustness and safety during rehab exercises.
    - *Unique Features:* The safety supervisory fuzzy controller monitors the real-time state of the limb, adjusting the interaction force between the robot and the impaired limb to ensure compliance and safety.
    - *Results:* The combination of impedance control and safety supervisory fuzzy controller provides a comprehensive control strategy, guaranteeing precision, safety, and resilience during therapeutic sessions.

2.8 Robotics

In this section, we will address the necessary theoretical background in robotics and control. Note that only serial robotic manipulators are discussed as the project revolves around this type of robot. This section is based on concepts presented in [79, 80].

2.8.1 Forward Kinematics of Serial Robotic Manipulator

Serial robotic manipulators are mechanical devices comprised of a series of segments, linked by joints. These joints allow one degree of freedom between consecutive segments. Thus,

the number of joints between the segments determines the degrees-of-freedom of the robotic manipulator. The following convention will be adhered to: the base of the robot is denoted as segment 0, while the last segment is denoted as segment n. An effector is attached to the terminal segment n, and the frame {e} is fixed to it, earning it the name of end-effector.

The configuration of the manipulator can be fully described by the values of the degrees of freedom at each joint. The variables determining the configuration of the robot are denoted by $q = (q_1, ..., q_n)$ and are referred to as the joint space configuration.

Each link of the manipulator has a relative position and orientation with respect to the robot's base. This concept is referred to as the reference frame in robotics, which can be related to the reference frames of the other links by applying homogeneous transformations.

In the context of these robotic manipulators, we can consider two distinct frames, $\{A\} : (x_a, y_a, z_a)$ and $\{B\} : (x_b, y_b, z_b)$, along with a specific point P. The position of P within $\{A\}$ is denoted as P^A. We can also consider the position of reference $\{A\}$ in relation to $\{B\}$, denoted as P_A^B, as well as the orientation between these two frames, which is defined by a 3×3 rotation matrix, R_A^B.

With these considerations in mind, we can define the position of point P within the reference frame b. Equation 2.1 provides a comprehensive understanding of the point's position within the robotic manipulator's structure.

$$P^B = P_A^B + R_A^B P^A \tag{2.1}$$

In the context of affine transformations, we can use homogeneous coordinates to represent points in an affine space. In this case, a point P in a three-dimensional affine space is represented by a 4-dimensional vector $\bar{P^A} = [p_x, p_y, p_z, 1]^T$, and a vector is represented by a 4-dimensional vector $\bar{u^A} = [u_x, u_y, u_z, 0]^T$. To simplify the use of affine transformations, a matrix representation is used to fully describe the homogeneous transformation as shown in equation 2.2.

$$T_A^B = \begin{bmatrix} R_A^B & P_A^B \\ 0_{1\times3} & 1 \end{bmatrix} \tag{2.2}$$

where R_a^b is the rotation matrix from frame $\{A\}$ to $\{B\}$, P_A^B is the translation vector from the origin of frame B to the origin of $\{A\}$ in the coordinates of $\{B\}$, and $0_{1\times3}$ is a row vector of zeros. This approach to defining positions within the manipulator's structure is crucial for understanding and controlling the movements and operations of the robotic manipulator.

Forward Kinematics

The forward kinematics of a robot manipulator allows us to compute the position and orientation, or pose, X of the end-effector based on the robot's joint parameters. This calculation is fundamental for predicting the end-effector's location in the robot's workspace. The concept of forward kinematics is expressed as:

$$T_0^e(q) = T_1^0(q_1)T_2^1(q_2)...T_n^{n-1}(q_n)T_e^n \tag{2.3}$$

Within a robotic system, the specific configuration of joints and links directly influences the end-effector's final pose. Forward kinematics facilitates this computation. The transformation matrix T_i^{i-1} denotes the spatial relationship between successive joint/link pairs. This relationship, defined by each joint's motion (rotational or translational), merges information about both rotation and translation into a single matrix, offering a complete representation of one frame's position relative to another.

In the context of serial manipulators, calculating the end-effector's pose requires combining all individual joint motions. This combination is performed through the multiplication of each joint's transformation matrices, starting from the base (fixed) frame up to the end-effector. The result, $T_0^e(q)$, is a comprehensive transformation mapping that reflects the end-effector's pose relative to the base, factoring in all joint positions and orientations along the way. Successfully solving the forward kinematics equates to determining the $T_0^e(q)$ consistent with known joint parameters. The following subsection presents the Denavit-Hartenberg parameters method to compute the transformation matrices T_i^{i-1}.

Denavit-Hartenberg Parameters

The Denavit-Hartenberg (DH) parameters are a method used to standardize the mathematical modeling of robotic arms. These are four parameters associated with a convention for attaching frames to the links of a spatial kinematic chain or robot arm. The parameters are defined as follows: Link length (a), Link twist (α), Link offset (d), and Joint angle (θ). Figure 2.7 provides a visual representation of those parameters.

These parameters can be used to compute the transformation matrices for each link of the robotic manipulator using the following equation:

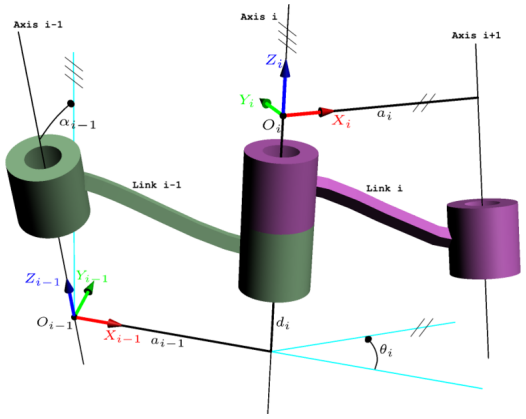


Figure 2.7 Representation of the DH parameters between 3 joints

$$T_{i-1}^{i} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_{i-1}) & \sin(\theta_i)\sin(\alpha_{i-1}) & a_{i-1}\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_{i-1}) & -\cos(\theta_i)\sin(\alpha_{i-1}) & a_{i-1}\sin(\theta_i) \\ 0 & \sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

These parameters are crucial for describing the geometry of a robot, and are typically specified by the manufacturer of a manipulator.

2.9 Differential Kinematics and the Geometric Jacobian Matrix

Differential kinematics describes the relationship between the derivative of joint velocities of a robot manipulator and the corresponding linear and angular velocities of the end-effector. This relationship is encapsulated by a Jacobian matrix, which plays a central role in inverse kinematics algorithms, manipulator dynamics modeling, and control.

Linear velocity is the rate of change of the position of the end-effector with respect to time, while angular velocity is the rate of change of the orientation of the end-effector with respect to time. The linear and angular velocities of the end-effector, $\dot{p}$ and ω respectively, can be unified and represented by $\dot{x} = [\dot{p}, \omega]^T$ and $J = [J_v, J_\omega]^T$ is the corresponding Jacobian matrix. The velocity of the end-effector, $v_e = \dot{x}$, can be calculated using the Jacobian matrix and the joint velocities as follows:

$$v_e = \dot{X} = J\dot{q} \tag{2.5}$$

The linear velocity portion contributed by the i^{th} joint, $\dot{p}_i$, and the angular velocity portion, ω_i, can be calculated using the respective parts of the Jacobian matrix and the joint velocity

as follows:

$$\dot{p_i} = J_{v_i}\dot{q_i} \tag{2.6}$$

$$\omega_i = J_{\omega_i}\dot{q_i} \tag{2.7}$$

Differential kinematics is used in the control of robotic manipulators. It allows the calculation of joint velocities required to achieve desired end-effector velocities, given the current configuration of the robot. This is particularly useful in trajectory planning and tracking.

The geometric Jacobian matrix is a crucial concept in the field of robotics, particularly in the context of manipulator kinematics. It establishes a linear relationship between the joint velocities and the end-effector velocities, thus providing a means to describe the attainable linear and angular velocities of a manipulator in any given configuration.

The geometric Jacobian matrix can be computed from the transformation matrices. For a manipulator with n joints, the Jacobian matrix J is a $6 \times n$ matrix, where each column J_i is given by:

$$J_i = \begin{bmatrix} J_{v_i} \\ J_{\omega_i} \end{bmatrix} \tag{2.8}$$

where J_{v_i} and J_{ω_i} are the linear and angular velocity components of the Jacobian, respectively. These components can be computed as follows:

- For a revolute joint:

$$J_{v_i} = z_{i-1} \times (o_n - o_{i-1}) \tag{2.9}$$

$$J_{\omega_i} = z_{i-1} \tag{2.10}$$

- For a prismatic joint:

$$J_{v_i} = z_{i-1} \tag{2.11}$$

$$J_{\omega_i} = 0 \tag{2.12}$$

In these equations, z_{i-1} is the axis of motion for the i^{th} joint, o_n is the origin of the end-effector frame, and o_{i-1} is the origin of the $(i-1)^{th}$ joint frame.

It's important to note that the Jacobian matrix is configuration-dependent, meaning it changes with the joint angles of the manipulator. This property makes it a powerful tool for

analyzing the manipulator's behavior in different configurations, including the detection of singularities where the manipulator loses its ability to move in certain directions.

2.9.1 Inverse Kinematics

The workspace of a robotic manipulator often serves as the defining factor for trajectory due to its direct relevance to the task at hand. Typically, these trajectories are expressed in the Cartesian space of the manipulator, as it allows for intuitive interpretation and definition. However, in order to effectively reproduce the desired motion, the trajectory must also be converted into the corresponding joint space. This conversion represents the problem of *Inverse Kinematics*. The inverse kinematics is represented by the following equation :

$$q_{k+1} = q_k + J^{-1}(q_k)(X_d - X_k) \tag{2.13}$$

Here, q_{k+1} and q_k denote the updated and current joint variables respectively. The term $J^{-1}(q_k)$ represents the inverse of the Jacobian matrix, computed using the present joint variables. This matrix serves to map joint velocities to the corresponding end-effector velocities. The Jacobian matrix being potentially non-square and/or non-invertible, the inverse is replaced by the Moore-Penrose pseudo-inverse. The pseudoinverse of a matrix J can be calculated using the following expression:

$$J^{\dagger} = (J^T J)^{-1} J^T \tag{2.14}$$

where J^T represents the transpose of J, and $(J^T J)^{-1}$ is the inverse of $J^T J$.

The Cartesian space desired pose of the end-effector is symbolized by X_d, while X_k signifies the current end-effector pose.

Thus, $J^{\dagger}(q_k)(X_d - X_k))$ calculates the requisite adjustment in the joint variables to minimize this difference. As a result, the joint variables are updated in every iteration with the objective of reducing this error effectively.

The iterative process continues until the difference between the current end-effector position and the desired position falls below a certain threshold or until a certain maximum number of iterations has been reached. The ultimate result is a set of joint variables that aligns the end-effector as closely as possible with the desired position in Cartesian space.

2.9.2 Singularities in Robotic Manipulation

In the domain of robotic kinematics, singularities represent configurations where a robot loses one or more degrees of freedom, potentially hindering its ability to achieve certain motions or exert forces in specific directions. Singularities occur in various forms, and understanding them is crucial for effective and safe robot operation.

Singularities arise primarily from the Jacobian matrix associated with the robot's kinematics. The Jacobian matrix relates joint velocities to end-effector velocities. A configuration is termed singular if the determinant of the Jacobian matrix becomes zero, rendering its inversion impossible or ill-conditioned.

There are several reasons for the occurrence of singularities:

- **Alignments**: When two or more joint axes align, a singularity can arise. For instance, in a typical 3-DOF planar robot, aligning all joints leads to a singularity.
- **Reach Extremes**: Extending the arm to its maximum reach or retracting it fully can induce singularities, as seen in certain industrial robots.
- **Wrist Configurations**: For robots with spherical wrists, certain wrist orientations can lead to singularities, especially when the wrist center aligns with the forearm.

The presence of singularities in robotic systems introduces several challenges:

- **Infinite Velocities**: At singular configurations, infinitesimally small changes in the end-effector's position or orientation might require infinite joint velocities.
- **Loss of Motion Control**: Robots might lose the capability to move in certain directions when in a singular configuration.
- **Control Instabilities**: Near singularities, control algorithms, especially those reliant on the Jacobian's inversion, can become unstable, leading to erratic robot behavior.
- **Safety Concerns**: Unbounded velocities and control instabilities can pose serious safety threats, especially in human-robot collaborative settings.

2.9.3 Manipulability Gradient

In robotics, the concept of manipulability quantifies a manipulator's ability to execute motion and exert forces in various directions within its workspace. It is crucial in assessing the

robot's performance based on its current joint configuration. The manipulability measure (w) is expressed as:

$$w = \sqrt{\det(JJ^T)} \tag{2.15}$$

where J is the Jacobian matrix that relates joint velocities to end-effector velocities. This measure reflects the isotropy and efficiency of the manipulator's movements.

Critical to manipulability is its gradient (∇w) concerning the joint positions, given by:

$$\nabla w = \begin{bmatrix} \frac{\partial w}{\partial q_1} \\ \vdots \\ \frac{\partial w}{\partial q_n} \end{bmatrix} \tag{2.16}$$

The gradient ∇w indicates how alterations in the joint variables influence manipulability, pointing toward the direction of its maximum rate of increase. This understanding is vital for optimizing the robot's configuration for specific tasks, enhancing its dexterity and precision. Thus, the manipulability gradient serves as a significant analytical tool in the nuanced control and improved functionality of robotic systems.

2.9.4 Handling Orientations in Robot Kinematics

In robotic kinematics, accurately representing and controlling the orientation of elements like the end-effector is critical. Various methodologies can facilitate this, each with its peculiarities and application nuances. Among these, the Roll-Pitch-Yaw (RPY) system and quaternions are prominently used due to their distinctive advantages and limitations in different scenarios.

Roll-Pitch-Yaw: An Intuitive Approach to Orientation

The Roll-Pitch-Yaw (RPY) system is an intuitive method for describing the orientation of an object in space. Orientations are described using three angles: roll (ϕ), pitch (θ), and yaw (ψ), which represent rotations around the X, Y, and Z axes, respectively. The RPY angles are used to construct a rotation matrix, often in the order of roll (X-axis rotation), pitch (Y-axis rotation), and then yaw (Z-axis rotation):

$$R = R_z(\psi)R_y(\theta)R_x(\phi) \tag{2.17}$$

However, the RPY representation is not without its complications, notably the issue of gimbal lock, which arises when the pitch angle approaches ± 90 degrees, causing a loss of one degree of freedom in 3D space.

Quaternions: Avoiding Singularities and Complexities

Quaternions provide a robust way to represent orientations and rotations in three dimensions, avoiding the pitfalls of gimbal lock. A quaternion is a four-element vector that extends complex numbers. It is usually represented as $q = [q_w, q_x, q_y, q_z]$, where q_w is the scalar part and q_x, q_y, and q_z form the vector part. The quaternion representing a rotation is given by:

$$q = [\cos(\frac{\theta}{2}), \vec{v}\sin(\frac{\theta}{2})] \tag{2.18}$$

where θ is the rotation angle and $\vec{v}$ is the unit vector along the rotation axis.

Conversion from RPY to Quaternion: Converting RPY angles to a quaternion involves a formula that considers all three rotations. The conversion can be expressed as:

$$q = \begin{bmatrix} \cos(\frac{\phi}{2})\cos(\frac{\theta}{2})\cos(\frac{\psi}{2}) + \sin(\frac{\phi}{2})\sin(\frac{\theta}{2})\sin(\frac{\psi}{2}) \\ \sin(\frac{\phi}{2})\cos(\frac{\theta}{2})\cos(\frac{\psi}{2}) - \cos(\frac{\phi}{2})\sin(\frac{\theta}{2})\sin(\frac{\psi}{2}) \\ \cos(\frac{\phi}{2})\sin(\frac{\theta}{2})\cos(\frac{\psi}{2}) + \sin(\frac{\phi}{2})\cos(\frac{\theta}{2})\sin(\frac{\psi}{2}) \\ \cos(\frac{\phi}{2})\cos(\frac{\theta}{2})\sin(\frac{\psi}{2}) - \sin(\frac{\phi}{2})\sin(\frac{\theta}{2})\cos(\frac{\psi}{2}) \end{bmatrix} \tag{2.19}$$

Conversion from Quaternion to RPY: Conversely, to obtain RPY angles from a quaternion, the following equations are used:

$$\phi = \text{atan2}(2(q_w q_x + q_y q_z), 1 - 2(q_x^2 + q_y^2)) \tag{2.20}$$

$$\theta = \text{asin}(2(q_w q_y - q_z q_x)) \tag{2.21}$$

$$\psi = \text{atan2}(2(q_w q_z + q_x q_y), 1 - 2(q_y^2 + q_z^2)) \tag{2.22}$$

These transformations between RPY and quaternions are fundamental in robotics, particularly when dealing with 3D orientations and rotations, as they offer a more stable and continuous way of handling rotations compared to traditional Euler angles or RPY, which suffer from singularities.

2.9.5 Dynamic Model

In this section, the theoretical framework of the robotic manipulator's dynamic model will be explored. While kinematics focuses on the motion and positions, dynamics provides insights

into the causes of movements in relation to the mass and forces in the system. The dynamic model of a manipulator is critical for control strategies, especially sophisticated ones like force control. Recognizing how the system responds to applied force is vital because the actuator at each joint exerts a force on a certain link, and this behaviour is dictated by the way the mass and forces are distributed. Therefore, understanding this behaviour by establishing the model can enhance the performance and efficiency of a robotic manipulator in executing complex tasks.

A common way to express the dynamic model is via the Euler-Lagrange model, defined by equation 2.23.

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \tag{2.23}$$

Here:

- $M(q)$ is the mass matrix, linking joint accelerations ($\ddot{q}$) to joint torques (τ).
- $C(q, \dot{q})$ represents the matrix of Coriolis and centrifugal forces, correlating joint velocities ($\dot{q}$) to joint torques.
- $G(q)$ refers to the gravity forces vector, encapsulating the effects of gravity on the robot's joints.
- τ stands for the joint torques or forces exerted on the robot's joints.

When the robotic manipulator operates at low accelerations, the mass matrix, also referred to as the inertia matrix, has little to no impact on the dynamic model. The same goes for the velocities, if the joint velocities are low, the effect of the Coriolis and centrifugal forces are negligible. However, the gravitational effects play a consistent role in the dynamics of a robot manipulator, without any regards to the joint's velocities and accelerations. This effect is translated in form of a gravitational wrench. This wrench is a 6D vector composed of gravitational force (F_x, F_y, F_z) and the associated torque (τ_x, τ_y, τ_z). Both force and torque are influenced by the mass of components attached to the joint and their positions.

The gravitational force on a body (F_g) is calculated as $F_g = m \cdot g$, where m is the body's mass and g is the acceleration due to gravity. The corresponding torque on the body from gravity is computed as $\tau_g = r \times F_g$, where r is the position vector from the joint to the body's center of mass (COM). To compute the total gravitational force and torque at each joint, we sum up the individual forces and torques from all bodies attached to that joint. This

process starts from the last joint and moves towards the base of the robot, resulting in the gravitational force and torque in the base frame.

The calculated forces and torques are then transformed from the base frame to the joint frame using the transformation matrix derived from the robot's current pose. This matrix provides the position and orientation of each joint relative to the base.

The final result is the gravitational wrench at each joint, expressed in the joint frame. This wrench directly represents the effects of gravity on the joint, helping us understand its behavior under the influence of gravity on the robot. In control, to consider the effect of gravity on the dynamic model, only the gravity wrench's component acting on the rotational axis of the given joint's actuator is considered, resulting in $G(q)$.

CHAPTER 3 RESEARCH OBJECTIVES

3.1 Research Question

How a Virtual Reality-integrated upper limb robotic rehabilitation system can enhance the safety, effectiveness, and personalization of rehabilitation exercises for individuals with upper limb impairments?

3.2 Hypotheses

Hypothesis #1: An admittance-based control approach is an effective and safe solution for recording and replicating therapist-induced movements of patient's upper limb.

Hypothesis #2: An impedance-based control approach to follow the exercise trajectory can ensure both safety and required precision in an upper limb rehabilitation scenario.

Hypothesis #3: An adaptive control strategy of the robotic platform will allow the patient to execute upper limb rehabilitation more precisely and efficiently.

3.3 Main Objective and Specific Objectives

The main objective of this project is to develop an upper limb robotic rehabilitation solution by using Virtual Reality, and an impedance-based control strategy to improve the quality of life for individuals with upper limb impairments.

To achieve the main objective, the following specific objectives are defined:

1. SO1: Identify the requirements related to the effective integration of a robotic solution into an upper limb rehabilitation framework.
2. SO2: Develop a human-robot interaction control strategy based on adaptive impedance to ensure safe, responsive and personalized interaction during upper limb rehabilitation exercises based on the patient's requirements.
3. SO3: Validate the developed solution in a controlled setting.

CHAPTER 4 METHODOLOGY

4.1 Methodology Overview

This section outlines the methodology used in this research, starting with a thorough description of the experimental setup. It includes an explanation of the virtual reality system and the specific virtual environment created for this study. Next, it introduces the two robotic manipulators utilized: Kinova's MICO^2 and UFactory's xArm7, and discusses how these elements were integrated to form a cohesive research apparatus.

Following this, the focus shifts to the development of three distinct control strategies essential for the interaction between the user and the robot during rehabilitation exercises:

- The first strategy, **admittance control**, is used to make the robot compliant to the movements of the therapist user to learn a desired trajectory to be replicated by the robotic device in a passive exercise.
- The second strategy, **impedance control**, enables the robot to follow a specific trajectory while ensuring the safety of the patient user by allowing deviations to the trajectory to be followed.
- The third strategy is based on an **adaptive law** to personalize the rehabilitation exercise by adjusting the controller parameters to the patient's performance and required assistance.

The section concludes with a detailed overview of the experiments designed to gather data for validation purposes.

4.2 Experimental Setup

4.2.1 Virtual Reality

The design of the VR environment is presented in Annex B.

4.2.2 Robotic Manipulator I: Kinova MICO^2

Preliminary experiments were conducted using the Kinova MICO^2 manipulator. Subsequently, all results were successfully replicated with the Ufactory xArm7 robotic manipulator.

For readers interested in utilizing the Kinova MICO^2, Annex A provides detailed information on the necessary hardware and software required to operate this device.

4.2.3 Robotic Manipulator II: UFactory xArm7

The UFactory xArm7, a lightweight carbon fiber collaborative robotic manipulator, stands out for its precision, adaptability, and versatility. This robotic arm, weighing 12.2 kg with a reach of 700 mm and a maximum speed of 1 m/s, was chosen for developing an upper limb robotic rehabilitation platform due to its multifaceted features and capabilities, as illustrated in Figure 4.1.



Figure 4.1 xArm7 robotic arm

Hardware

Table 4.1 enumerates the xArm7's specifications. The Denavit-Hartenberg and gravitational parameters are presented in Tables 4.3 and 4.2, respectively. Figure 4.2 present the reference frames used.

Software

The primary interface for the xArm7 is the UFactory Studio, a graphical application equipped with a range of utilities, including parameter setting, motion planning, and direct interaction

Table 4.1 Specifications of the UFactory xArm7 robotic arm

Specification	**Value**
Total Weight	14.5 Kg
Payload	3.5 Kg
Reach	691 mm
Maximum Linear Arm Speed	1 m/s
Power Supply Voltage	24 V DC
Communication Protocol	Modbus TCP

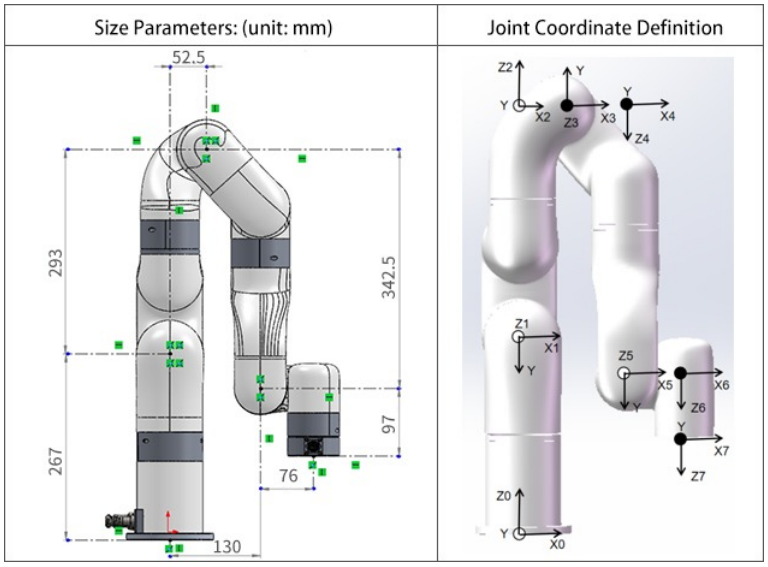


Figure 4.2 Frames for xArm7 DH parameters. [2]

Table 4.2 UFACTORY xArm7-Model Dynamics [2]

Dynamics	**Mass (Kg)**	**Center of Mass (mm)**
Link1	2.177	[0.15, 27.24, -13.57]
Link2	1.716	[0.22, -124.70, 18.90]
Link3	1.485	[46.00, -22.30, -8.47]
Link4	1.574	[69.75, -112.50, 13.20]
Link5	1.209	[-0.35, 17.60, -28.40]
Link6	1.214	[63.65, 30.84, 21.70]
Link7	0.170	[0.00, -6.77, -10.98]

Table 4.3 xArm7 DH parameters

Joint	$\boldsymbol{\theta}$ **(rad)**	**d (mm)**	$\boldsymbol{\alpha}$ **(rad)**	**a (mm)**	**offset (rad)**
Joint1	0	267	$-\pi/2$	0	0
Joint2	0	0	$\pi/2$	0	0
Joint3	0	293	$\pi/2$	52.5	0
Joint4	0	0	$\pi/2$	77.5	0
Joint5	0	342.5	$\pi/2$	0	0
Joint6	0	0	$-\pi/2$	76	0
Joint7	0	97	0	0	0

through the xArm-Python-SDK API. Advanced control and integration are made possible through APIs in ROS, Python, and C++.

Communication Protocol

Using the Modbus-TCP protocol, the xArm7 guarantees robust communication via its Ethernet-enabled control box, as represented in Figure 4.3. Proper IP address configuration is vital to ensure seamless interaction. Within the provided interfaces, angles and coordinates are user-friendly, appearing in degrees and millimeters respectively, while the Control Box supports versatile analog I/O functions.

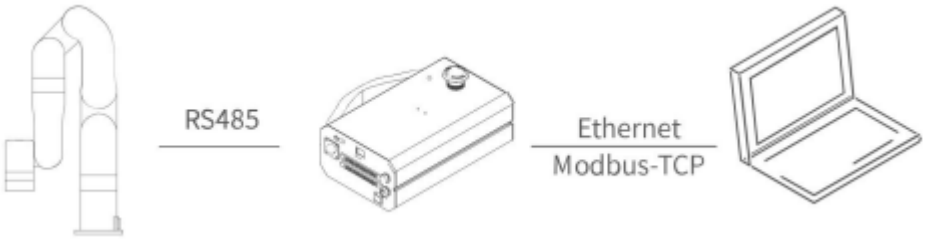


Figure 4.3 Modbus-TCP communication protocol for xArm7. [5]

Python SDK for xArm7

The Python SDK for the xArm7 provides granular control over the robotic arm, allowing for a diverse set of operations. These control modes encompass Position Control, Velocity Control, and Force Control, with the latter being dependent on a 6-axis force/torque sensor. Additionally, developers can retrieve extensive information from the xArm7, aiding in real-time monitoring and diagnostics.

Safety Features

Multiple safety measures are incorporated, including a Safety Boundary to limit its operational range, a Reduced Mode to constrain speed and range, and an Emergency Button to halt all movements immediately.

xArm 6 Axis Force Torque Sensor

The xArm 6 Axis Force Torque Sensor is designed for force and torque data acquisition on xArm. This sensor can simultaneously measure the force and torque in a three-dimensional space. It is specifically installed on the xArm tool flange. To facilitate its functionality, a flexible cable is used for both power and communications.

Below is a detailed specification table that outlines the various technical parameters associated with the xArm 6 Axis Force Torque Sensor.

Table 4.4 Axis Force Torque Sensor specifications

	Fx, Fy	**Fz**	**Tx, Ty, Tz**
Load capacity	150N	200N	4Nm
Resolution	100mN	150mN	5mNm
Hysteresis	2.5%FS	1%FS	1%FS
Crosstalk	3%FS	3%FS	3%FS
Overload capacity	150%	150% (Fz+) 300% (Fz-)	150%
weight	445g		

4.2.4 Arm Support

The effective application of robotic manipulators in rehabilitation requires a specialized interface, termed "Arm Support," which connects the patient's affected arm to the robot's end effector. This critical component was initially designed and prototyped for the MICO robotic manipulator and later customized to fit the xArm7.

The primary considerations in designing the arm support for the MICO included maintaining a lightweight structure, ensuring adaptability to different limb sizes, and providing the option to secure the elbow at specific angles during exercises. A depiction of the arm support can be seen in Figure 4.4.

Further developments led to the adaptation of the MICO^2's arm support for use with the xArm7. This process involved simplifying the original model and modifying the adapter—the

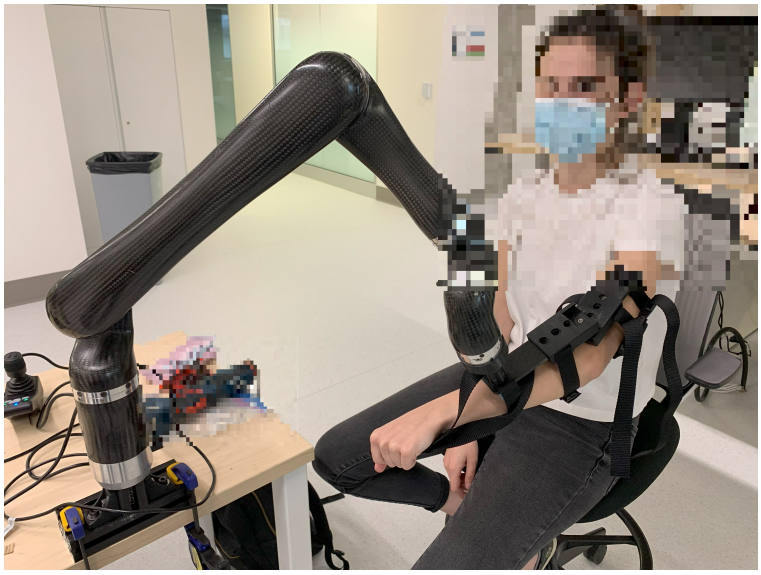


Figure 4.4 Arm Support developed for the MICO 6 DOF

component connecting directly to the robot—so it aligned with the xArm7's tool-head pattern. The revised design for the xArm7 is illustrated in Figure 4.5. The original support, for the MICO, was designed and prototyped by an undergraduate student.

4.2.5 System Integration

This section presents the methodologies employed to integrate the components of the robotic rehabilitation platform, ensuring synchronous operation.

VR and Robotic Device Synchronization

The synchronization between the VR system and the robotic device is pivotal for seamless operation. This was achieved through socket communication over a local port, where a `TcpListener` in Unity listens and processes robot data asynchronously. The foundation of this communication approach is derived from a GitHub repository. [81]

Both Unity and the robot side maintain this communication through specific code snippets. For instance, the Unity side employs a listener, while the robot, based on conditions, initializes and transmits data over the designated socket.

The essence of this methodology is its low latency, typically within milliseconds. Synchronization between the VR system and robot is governed through distinct flags that signal varied operations or phases, such as initialization (*bStandby*), trajectory generation (*bGeneration*),

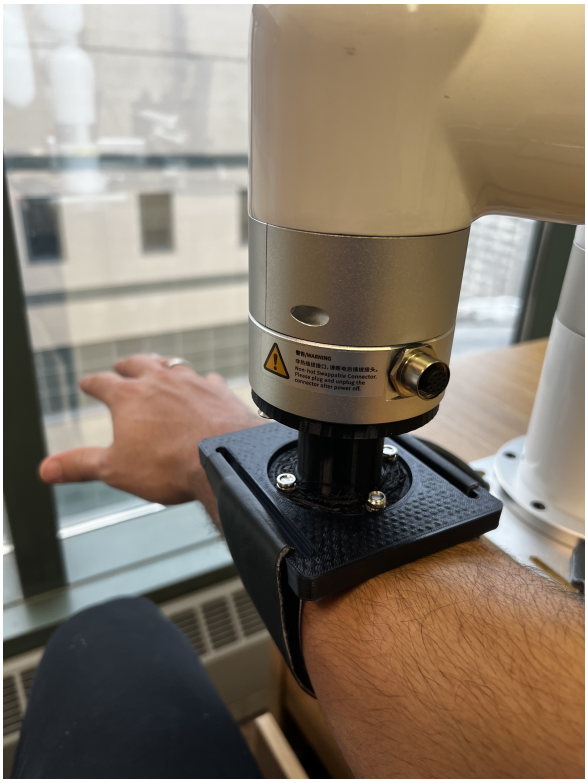


Figure 4.5 Adapted Arm Support for the xArm7

and trajectory tracking (*bTracking*).

The different phases defined and the use of the flags to synchronized the information between the VR environment and the robotic device are as follow:

Initialization (*bStandby = True; bGeneration = False ; bTracking = False*): In this phase, the robot is in "Admittance Mode" (compliant). The VR displays the arm's live position of the patient's arm. Once the therapist has installed and moved the patient's arm to the desired starting point, the **trajectory generation can start**.

Trajectory Generation (*bStandby = True; bGeneration = False ; bTracking = False*): The robot starts recording the patient's arm movement. This movement is mirrored visually within the VR environment.

Trajectory Tracking (*bStandby = True; bGeneration = False ; bTracking = False*): After defining the path, the robot shifts control modes to the trajectory tracking (Impedance Control). The desired path to follow, the real-time position of the patient's arm and the desired actual position are displayed in the VR environment.

The systematic implementation of these flags ensures the VR and robot remain synchronized,

presenting users with immediate and clear visual feedback.

Integration of Robotic Device and Arm Support

The integration of the arm support was primarily driven by design considerations. Specifically, the arm support was adapted to match the tool head pattern of the robotic device.



Figure 4.6 Experimental setup showing the VR, robotic device, and arm support

4.3 Control Strategies

In this section, we outline the core strategies utilized to control our robotic rehabilitation system while overcoming the limitations due to possible singularities. These strategies ensure precise and safe interactions between the robot and the patient. We will discuss the foundational dynamic models that underpin the system's behavior, followed by a deep dive into the admittance and impedance control methodologies. Lastly, we will explore the adaptive law, which allows the system to adjust in real-time based on specific patient needs. In order to design strategies specific to the operation of the robotic system in a rehabilitation context, the use-case scenario is described as below:

- **Trajectory following**: In all types of rehabilitation exercises, the robot should be able to follow a trajectory in Cartesian space. This calls for developing an **inverse kinematics** strategy capable of producing smooth trajectories while avoiding **singularities**.

- **Learning Therapist-induced movements**: In various cases of physical therapy, a therapist prescribes different movements specific to the rehabilitation of the affected limb and applies them with the patient (**Figure 4.7**). To replicate these movements, an **admittance-based controller** is required to make the end effector movements compliant to the therapist's. This admittance control strategy also allows for the implementation of a full active rehabilitation exercise where the patient follows a trajectory shown in the VR environment.
- **Collaborative trajectory following**: Next, an **impedance control** strategy is needed to allow for passive or active exercises when the robot follows a trajectory while allowing for deviations in position due to limitations of a specific joint's range of motion. Essentially, the impedance parameters define the assistive or resistive nature of robot's participation in the rehabilitation exercise.

4.3.1 Dynamic Models

The dynamic model of a robotic manipulator describes its behavior concerning the various forces and dynamic loads interacting with the system. This model is primarily employed in two distinct ways: first, in the context of *Forward Dynamics*, where the interaction forces applied by or on the robot are known, and the resulting movement is computed; Secondly, in the case of *Inverse Dynamics*, which involves determining the torques and forces involved in a particular movement, given a specific state of the robotic manipulator. Figure 4.8 illustrates the input and output of each model.

This concept is particularly relevant when working with force control strategies. Indeed, if the robot does not offer a torque control mode directly, as is the case with the xArm7, the dynamic model of the robot must be used. Using the Forward Dynamics of the xArm7, for a given force command, the equivalent motion command can be computed. However, accurately computing the dynamic of a manipulator is a complexe task, so a "first-order approximation" is used in this work.

First Order Approximation for Forward Dynamics

Generally, the dynamics of the robot manipulator is expressed by:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau, \tag{4.1}$$

where:

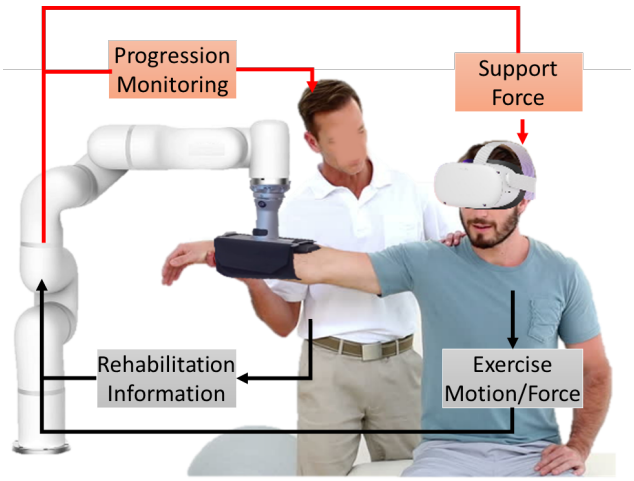


Figure 4.7 Use case representation of the robotic system for rehabilitation

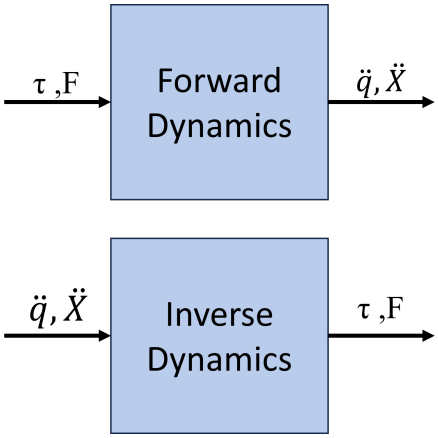


Figure 4.8 Forward and Inverse Dynamics models inputs and outputs in joint and Cartesian space.Here, τ represents the joint torques, F denotes the wrench in Cartesian space, $\ddot{q}$ is the joint's acceleration, and $\ddot{X}$ is the end-effector's acceleration

- $M(q)$ is the mass (inertia) matrix,
- $C(q, \dot{q})\dot{q}$ represents the Coriolis and centrifugal forces,
- $G(q)$ is the gravity vector,
- τ is the vector of torques applied at each joints,

For simplicity, the dynamics described above can be approximated by a second-order linear equation:

$$M\ddot{q} + B\dot{q} = \tau, \tag{4.2}$$

where:

- M is the mass (inertia) matrix approximation,
- B is the damping matrix approximation,
- $\dot{q}$ is the vector of joint velocities,
- $\ddot{q}$ is the vector of joint accelerations,
- τ is the vector of torques applied at each joints,

In this context, the gravitational term is considered known and simplified, and all terms proportional to joint velocities are grouped under one matrix termed the damping matrix, B.

Applying a straightforward order reduction method, where $\Theta = \dot{q}$, and utilizing the Laplace transform, the equation below is derived:

$$M\Theta s + B\Theta = T(s), \tag{4.3}$$

To discretize the above equation, the bilinear transformation is employed:

$$s = \frac{2}{t_s}\frac{1 - z^{-1}}{1 + z^{-1}}, \tag{4.4}$$

where:

- t_s is the sampling time,

- z^{-1} represents the backward shift operator,

With this transformation, the discretized form of the equation is:

$$\left(M\left(\frac{2}{t_s}\frac{1-z^{-1}}{1+z^{-1}}\right)+B\right)\Theta(i)=T(i). \tag{4.5}$$

By defining $\widetilde{M}=\frac{2M}{t_s}$ and $z^{-1}\Theta(i)=\Theta(i-1)$ and $z^{-1}T(i)=T(i-1)$, the terms are rearranged and solved for $\Theta(i)$, resulting in:

$$\Theta(i)=(\widetilde{M}+B)^{-1}(T(i)+T(i-1)+(\widetilde{M}-B)\Theta(i-1)), \tag{4.6}$$

where:

- $\Theta(i)$ is the joint velocity at the current time step,
- $T(i)$ and $T(i-1)$ are the joint torques at the current and previous time steps, respectively,
- $\Theta(i-1)$ is the joint velocity at the previous time step.

4.3.2 External Force Estimation

Estimating external forces is critical for the effective implementation of admittance and impedance control strategies, particularly in cases where the robotic manipulator does not have a force/torque sensor at the end-effector. This estimation relies on the dynamic model of the robotic manipulator, as shown in Equation 2.23. The gravitational matrix $G(q)$ can be computed using the robot's kinematic and dynamic parameters, which helps in understanding the forces acting on the robot.

The process involves calculating the gravitational torques acting on the robot's joints and then subtracting these from the total torques as recorded by the joint torque sensors, thereby isolating the torques due to external forces.

The effects of gravity on each joint are detailed as follows:

$$W_{\mathrm{g}}=\begin{bmatrix}F_{\mathrm{g}}\\ \tau_{\mathrm{g}}\end{bmatrix}. \tag{4.7}$$

In this formation:

- F_{g} denotes the force that gravity exerts on a joint,
- τ_{g} denotes the torque or rotational force that gravity applies at the joint.

The gravity induced torques (τ_g) are then subtracted from the total measured torques, leading to the **gravity compensated interaction torques** (τ_{int}), representing the forces from external factors.

$$\tau_{\mathrm{int}} = \tau_{\mathrm{s}} - \tau_{\mathrm{g}}; \tag{4.8}$$

The **gravity compensated interaction wrench** (W_{int}) applied on the robot is then estimated using the Jacobian matrix through the following equation :

$$W_{int} = J^T \tau_{\mathrm{int}}. \tag{4.9}$$

4.3.3 Admittance Control

Admittance control and impedance control are fundamental approaches in robotics for managing the interaction between a robot and its environment. They are inherently correlated through the concept of mechanical impedance. Mechanical impedance is a measure that describes how a system responds to external forces, encapsulating the relationship between velocity (or displacement) inputs and force outputs, often modeled through resistive (damping), inertial (mass), and elastic (stiffness) properties. Admittance is essentially the inverse of impedance. The mechanical admittance is the relationship between force inputs and velocity outputs. While admittance control guides motion in response to applied force, commonly used in applications requiring high compliance, impedance control modulates the force exerted based on deviation from a desired trajectory, often chosen for tasks requiring precise position control amidst dynamic interactions. Figure 4.9 illustrates these relationships.

Its primary input is the predetermined interaction force between the robot and its environment, and the corresponding output is the robot's motion. When the desired interaction force is zero, it ensures the robotic manipulator's complete admittance, resulting in the robot moving away whenever a non-zero interaction force is detected. This behavior can be adjusted to specific application needs by adjusting the desired mass, damping, and stiffness matrices. The optimization and selection of these parameters depend on the robot's physical attributes and the desired admittance level.

The mathematical representation of the desired dynamic behavior of the robot-environment system or the "Admittance Model" is illustrated by equation 4.10:

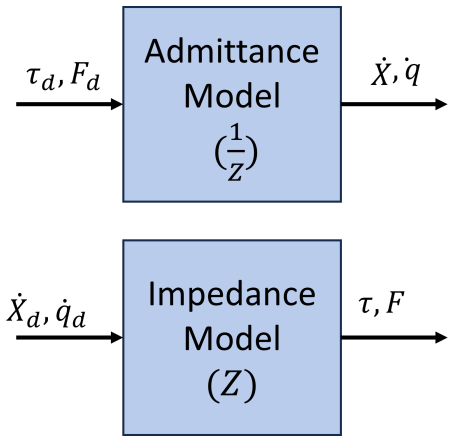


Figure 4.9 Admittance and Impedance models inputs and outputs.Here, τ represents the joint torques, F denotes the wrench in Cartesian space, $\dot{q}$ is the joint's velocities, and $\dot{X}$ is the end-effector's velocity. The subscript d denotes a desired parameter.

$$M_d \cdot \ddot{q} + B_d \cdot \dot{q} + K_d \cdot q = \tau_{int}, \tag{4.10}$$

where :

- M_d: Desired inertia matrix (mass) of the robot
- B_d: Desired damping matrix of the robot
- K_d: Desired stiffness matrix of the robot
- q: Joint positions
- $\dot{q}$: Joint speeds
- $\ddot{q}$: Joint accelerations
- τ_{int}: Interaction torques

The desired stiffness matrix K_d is set to zero to simplify the model. Equation 4.10 becomes :

$$M_d \cdot \ddot{q} + B_d \cdot \dot{q} = \tau_{int}. \tag{4.11}$$

For simplification and reducing the order, let's consider: $\omega = \dot{q}$. This leads to:

$$M \cdot \dot{\omega} + B \cdot \omega = \tau_{int}. \tag{4.12}$$

Applying Laplace Transform to the equation :

$$M \cdot \Omega s + B \cdot \Omega = T_{int}. \tag{4.13}$$

For discretization, the Tustin (trapezoidal) method, known for preserving system stability, is used. The method transforms the s-domain to z-domain as:

$$s \approx \frac{2}{T_s} \cdot \frac{z-1}{z+1}, \tag{4.14}$$

the following equation is obtained :

$$\left(M \cdot \frac{2}{T_s} \cdot \frac{z-1}{z+1} + B\right) \Omega(z) = T_{int}(z). \tag{4.15}$$

Expanding:

$$2M \cdot \frac{z-1}{T_s} \Omega(z) + B \cdot (z+1)\Omega(z) = T_{int}(z) \cdot (z+1). \tag{4.16}$$

From the z-domain equation, translating it to the time domain difference equation, and let $\omega[k]$ be the discretized version of $\Omega(s)$ at the k-th sample, and similarly, $\tau_{int}[k]$ is the sampled external torque.

The difference equation becomes:

$$2M \cdot (\omega[k] - \omega[k-1]) + B \cdot T_s \cdot (\omega[k] + \omega[k-1]) = T_s \cdot (\tau_{int}[k] + \tau_{int}[k-1]). \tag{4.17}$$

Thus, the current joint speed $\omega[k]$ at any discrete time step k is given by:

$$\omega[k] = \frac{T_s \cdot (\tau_{int}[k] + \tau_{int}[k-1]) - 2M \cdot \omega[k-1] - B \cdot T_s \cdot \omega[k-1]}{2M + B \cdot T_s}. \tag{4.18}$$

Admittance Control: Trajectory Generation and Implementation

For precise and consistent trajectory generation, the following systematic procedure was adopted:

1. **Initialization:** Parameters such as recording time, sampling time, and the filename for trajectory data storage are set.
2. **Starting Point:** The robotic manipulator is positioned at the intended starting location.

3. **Device Configuration:** Admittance mode is activated on the robotic device, ensuring full compliance with external inputs.

4. **Data Acquisition:** As the robot follows the desired trajectory, metrics like joint positions, velocities, and end-effector positions are continuously monitored and recorded. At each sampling point, this data is captured using the robot's API functions.

5. **Return Phase:** Once the recording duration concludes, the robot is reverted to its *home* position.

6. **Data Logging:** Acquired trajectory data is stored in the designated .txt file.

The Algorithm 1 provides a top-level depiction of the trajectory generation methodology using Admittance Control.

Algorithm 1 Admittance Control for Trajectory Generation

Require: Desired inertia, damping, and stiffness matrices: M_d, B_d, K_d, Interaction torques τ_{int}, Sampling rate Δt

```
Initialize robot in Admittance Mode.
Position robot at the starting point.
while Recording time not elapsed do
    Compute the desired joint velocity with respect to equation 4.18
    Capture and store joint positions, velocities, and end-effector metrics.
    Wait for Δt
end while
Save accumulated data to .txt file.
Return robot to home position.
```

Figure 4.10 illustrates the admittance control strategy.

4.3.4 Impedance Control

Impedance Control is a strategy employed to regulate the interaction forces between a robot and its environment, crucial for ensuring safe interaction in various settings, including therapeutic applications. Unlike Admittance Control, impedance control takes the desired motion as input and outputs a force.

The proposed controller based on the "Impedance Model" method eliminates the need for an external force sensor. The joint-space impedance relationship between motion and force is reflected in the Equation 4.19:

$$\tau_{int} = M_d\ (\ddot{q}_d - \ddot{q}) + B_d\ (\dot{q}_d - \dot{q}) + K_d\ (q_d - q). \tag{4.19}$$

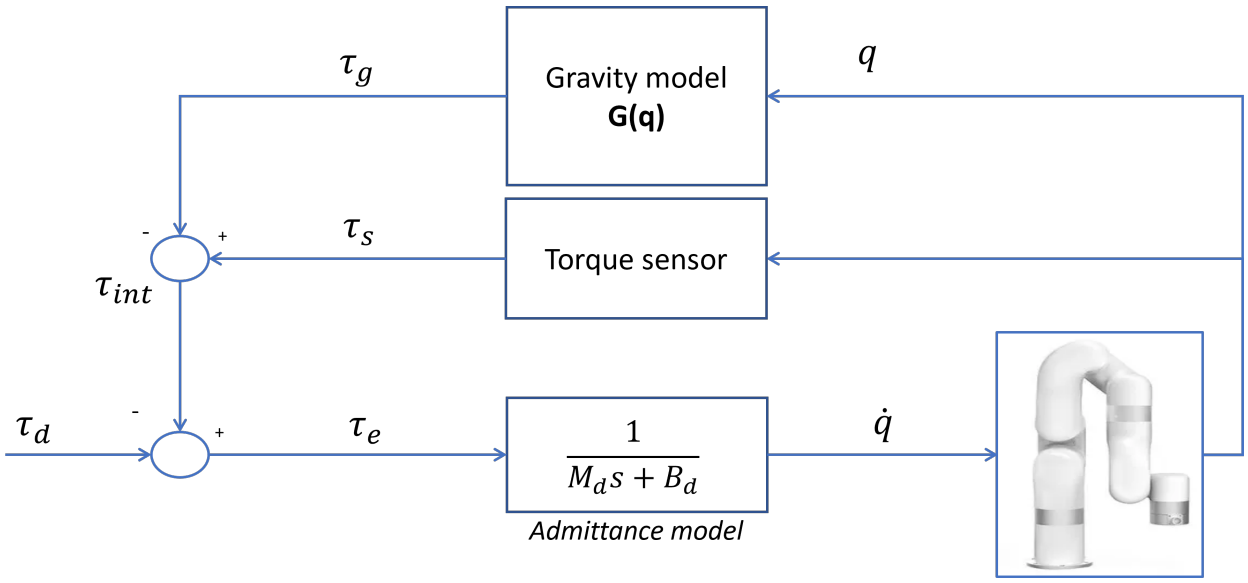


Figure 4.10 Joint space admittance control using the gravity compensated interaction torques

Here, M_d, D_d, and K_d represent the desired inertia, damping, and stiffness matrices, respectively. τ_{int} denotes the interaction torques vector. The dynamic model of the robotic manipulator is represented by Equation 4.20:

$$\tau = M(q)\ddot{q} + C(q,\dot{q}) + V\dot{q} + G(q) - \tau_{int}. \tag{4.20}$$

By setting $\ddot{q}_d$ to zero and isolating $\ddot{q}$ in Equation 4.19, we obtain Equation 4.21:

$$\ddot{q} = M_d^{-1}\,[\tau_{int} - D_d\,(\dot{q}_d - \dot{q}) - K_d\,(q_d - q)]. \tag{4.21}$$

Finally, the control law is obtained by setting M_d as $M(q)$ and replacing $\ddot{q}$ in Equation 4.20 with Equation 4.21. This results in Equation 4.22:

$$\tau = C(q,\dot{q}) + V\dot{q} + G(q) - D_d\,(\dot{q}_d - \dot{q}) - K_d\,(q_d - q). \tag{4.22}$$

Assumptions such as low joint velocities, making Coriolis and friction terms negligible, were made. The simplified final control law is given by Equation 4.23:

$$\tau = G(q) - D_d\,(\dot{q}_d - \dot{q}) - K_d\,(q_d - q). \tag{4.23}$$

This control law aims to balance trajectory tracking accuracy and robot compliance, empha-

sizing safety and adaptability according to specific needs.

The control law derived in Equation 4.23 is in terms of torques. However, some robotic manipulators, like the xArm7, do not provide a direct torque control command in their API. In such cases, the forward dynamics of the robot can be used to compute the equivalent velocity command for a given torque command. A first-order approximation of the forward dynamics, as discussed in the Dynamic Models section, can be used for this purpose.

Impedance Control: Trajectory Tracking and Implementation

The following procedure was adopted for trajectory tracking using impedance-based position control:

1. **File Specification:** Specify filenames for intended trajectory and tracking data.
2. **Data Loading:** Load the desired trajectory data.
3. **Initial Positioning:** Reposition the robot to its starting stance.
4. **Torque Computation:** For each time step, use Equation 4.22 to determine the torque command.
5. **Data Capture:** Concurrently, save data on joint positions, velocities, torques, and end-tool coordinates.
6. **Completion and Storage:** On trajectory completion, save the data in a .txt file and return the robot to its *home* position.

The following pseudo-code provides a high-level representation of the control loop:

Figure 4.11 shows the impedance control loop .

4.3.5 Adaptive Impedance Control

The variability in patient's motor functions throughout the rehabilitation process necessitates a dynamic approach to therapy. Rehabilitation requirements are not only patient-specific but also change as individuals progress through different recovery stages. From initial phases, often needing passive motion exercises, to more advanced stages where activities of daily living (ADL) exercises become crucial, the rehabilitation regime must adapt. ADL exercises mimic everyday actions, like opening a drawer or using utensils, helping patients regain practical motor skills.

Algorithm 2 Impedance Control Loop for the Robotic Manipulator

Require: Desired trajectory q_d, $\dot{q}_d$, Desired impedance parameters : M_d,D_d,K_d, Sampling rate Δt

Retrieve initial state q,$\dot{q}$
while Trajectory not completed **do**
    Compute control torque τ using Eq. (4.23)
    if Robot API provides torque control **then**
        Send control torque τ to the robot joints.
    else
        Compute equivalent velocity command using forward dynamics approximation.
        Send equivalent velocity command to the robot joints.
    end if
    Monitor and update q and $\dot{q}$ for the next cycle.
    Wait for Δt
end while
Save accumulated data to .txt file.
Return robot to *home* position.

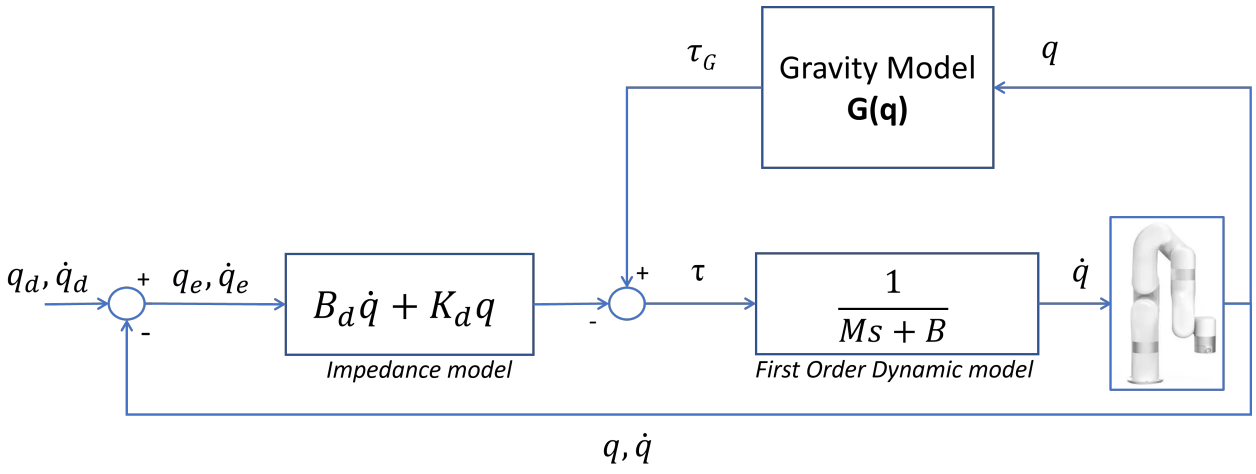


Figure 4.11 Joint-space impedance control strategy

This adaptive approach extends to the robotic controller's operation, where correct impedance model parameters are crucial to ensure the robot applies the proper support force to the patient's limb. Simplifying the use of the robotic rehabilitation platform involves considering primary factors such as the stage of the patient's recovery and the specific exercises prescribed. These factors inform the initial settings of the impedance parameters. Additionally, during exercises, any position error detected in the patient's limb prompts real-time adjustments of these parameters, making the system responsive to the patient's immediate needs.

The recovery stage and exercise type indicate the required assistance level. For example, in early recovery stages, a patient might not be able to exert any effort. However, as recovery progresses, they should provide most of the movement effort. Consequently, three foundational exercise types have been identified, each with preset parameters for the impedance control algorithm. Table 4.5 displays the exercise types alongside the corresponding robotic device support level:

Table 4.5 Types of exercises and corresponding robot support

Exercise Type	**Type of Support**
Passive Exercises	Robot provides complete support to the limb
Active Exercises	Robot provides partial support to the limb
Activities of Daily Living(ADL)	Robot provides minimal to no support to the limb

For real-time tuning of the parameters, the position error between the current desired position and the actual position of the patient's limb is utilized. This error becomes significant when the patient is tasked with following a given trajectory during an exercise. An increase in position error indicates the patient's difficulty in keeping up with the desired trajectory. This metric facilitates the dynamic adjustment of the control strategy's parameters in line with the patient's ongoing performance. Figure 4.12 illustrates this concept, where the vector $\vec{e}$ defines the position error. The desired trajectory is shown as a black dashed line, with the black cross indicating the current desired position. In contrast, the red line represents the patient's actual trajectory, and the red dot marks the current position of the limb.

To modulate the support based on this error, an adaptive law is used. The parameter is proportional to the position error, and its computation is presented in Equation 4.24. The underlying function is chosen for its beneficial properties: it has a 'dead zone' during which the system parameters remain unaffected by minor errors and upper/lower bounds to ensure safety. The dead zone offers a tolerance margin to the patient, allowing minor differences before the system starts amplifying its support. Simultaneously, the upper bound ensures that the system's total impedance remains within safe limits. If the impedance becomes

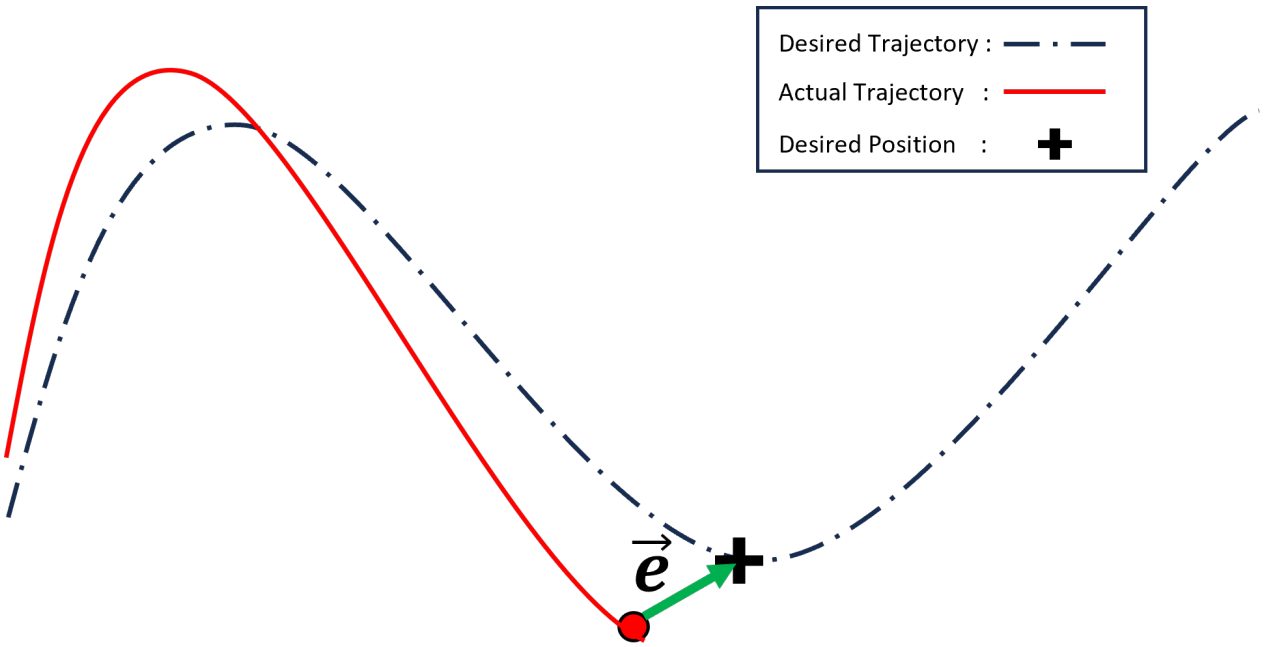


Figure 4.12 Graphical representation of position error in impedance control

excessively high, the resultant stiffness might pose injury risks to the patient. The adaptive law is given by:

$$Z_d = Z \cdot k \cdot \tanh(\gamma \cdot e^3). \tag{4.24}$$

where:

- Z_d is the desired updated impedance parameter,
- Z represents the baseline impedance parameter,
- k and γ are constants that shape the response curve, determined empirically, and
- e is the position error, indicating the deviation from the desired movement or position.

This formula uses the hyperbolic tangent function, ensuring a smooth transition between values, avoiding abrupt changes that could be uncomfortable or unsafe for the patient. The function's response, modulated by k and γ, adjusts the rate at which the impedance changes in response to the position error, e.

The cubic power on e amplifies larger errors, making the system more sensitive and responsive when greater deviations occur. This ensures that the support force is adequate even when the patient's movement significantly strays from the desired trajectory.

Both k and γ are determined through a process of trial and error. Note, different values of gamma are defined for the translation DOF and rotation DOF. The values are set at $k_{rot} = k_{trans} = 2$ and $\gamma_{trans} = 10^{-5}; \gamma_{rot} = 5$. The adaptive law is applied across all DOFs, specifically:

- All 6 DOFs for Cartesian control, adjusting the impedance for three-dimensional space movements.
- All 7 DOFs for joint-space control, modifying the impedance within the robot's articulated structure.

This consistent application ensures the robot provides suitable support, adapting in real-time to the patient's performance and assisting as necessary. The following figure shows the chosen parameters and the graphical representation of the adaptive law for both the translation and orientation DOF :

Adaptive Impedance Control: Implementation

Adaptive Impedance Control introduces dynamic interaction capability, allowing the robot to adapt its behavior based on environmental interactions. By implementing adaptive laws, the system can compute variable impedance parameters to suit the ongoing task requirements.

Implementation Overview:

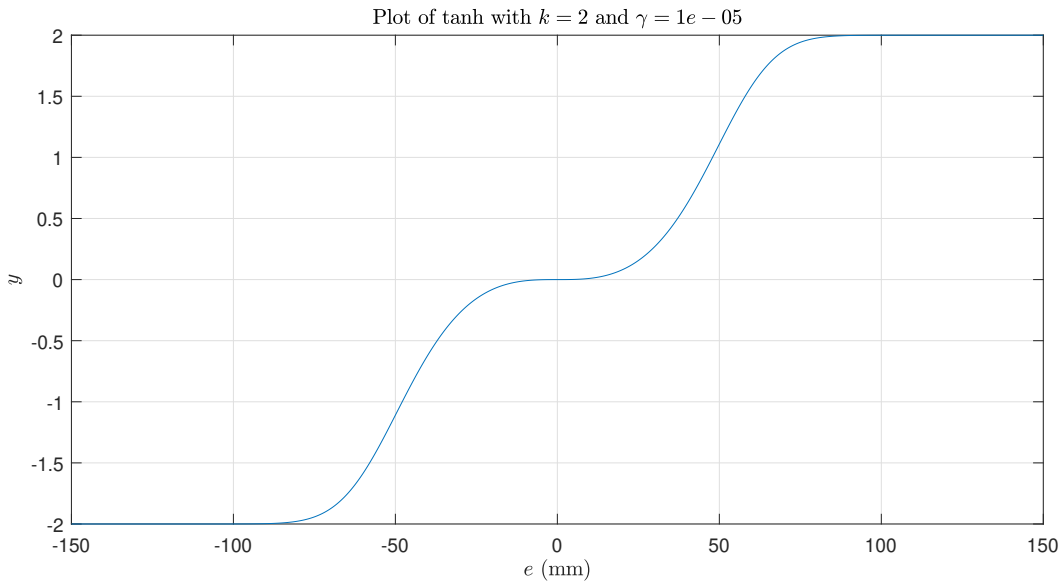
1. **Input Evaluation:** The system determines the state of the adaptive mode through the *bAdaptive* boolean.
2. **Application of Adaptive Law:** Using the hyperbolic tangent function and the error magnitude, the impedance parameters are dynamically adjusted using equation 4.24.

This modular approach ensures that the robot remains compliant and responsive, adapting its behavior to optimize interactions based on the task at hand.

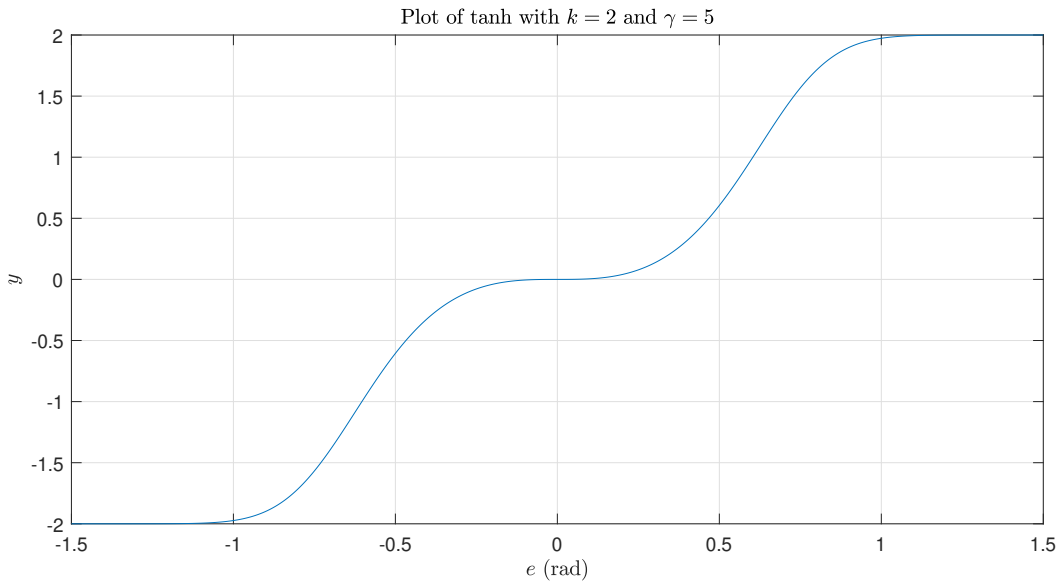
Figure 4.14 shows the adaptive impedance control loop :

4.4 Cartesian Admittance and Impedance Control

The previously presented control strategies are all developed in joint-space. While joint-space control strategies offer a simple and efficient implementation, the interpretation of the



(a) tanh function for the translation DOF



(b) tanh function for the rotation DOF

Figure 4.13 (a) adaptive law parameter for the translation DOF, (b) adaptive law parameter for the rotation DOF

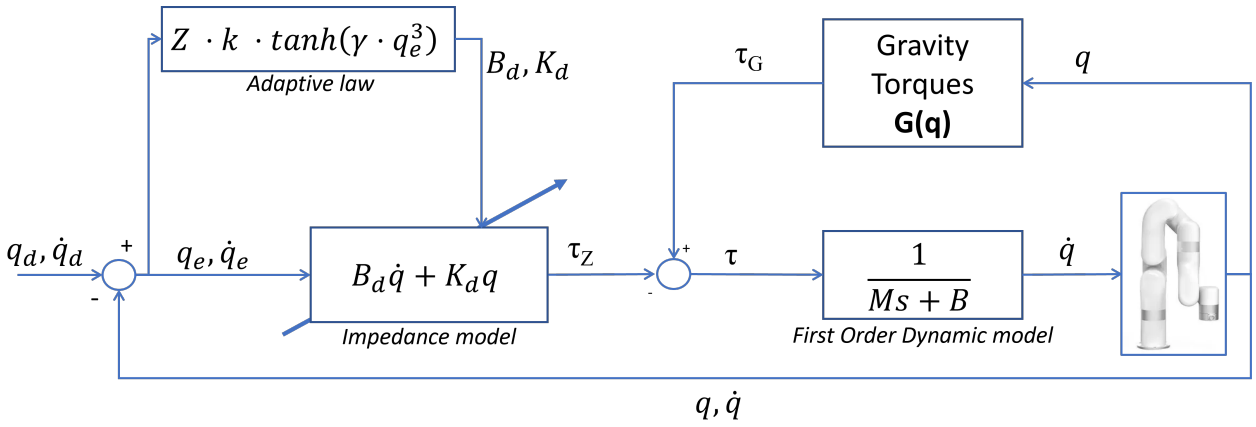


Figure 4.14 Joint space Adaptive Impedance control

different control parameters is complex, if not impossible. In the use case of therapy, the interpretability of the control parameters are often necessary. Therefore, it is often beneficial to implement the same control strategy in the Cartesian space, especially when dealing with tasks that require specific spatial trajectories or when interacting with environments that are defined in terms of Cartesian coordinates. The implementation of the Cartesian space control strategies was done using a 6 axis Force/Torque sensor.

4.4.1 Cartesian Admittance Control

To design the Cartesian Admittance Model, equation 4.18 is modified by replacing the interaction torque τ_{int} and the angular velocity ω by the interaction wrench F_{int} and the Cartesian velocity $\dot{X}$. The Cartesian Admittance equation is as follow :

$$\dot{X}[k] = \frac{T_s \cdot (F_{int}[k] + F_{int}[k-1]) - 2M \cdot \dot{X}[k-1] - B \cdot T_s \cdot \dot{X}[k-1]}{2M + B \cdot T_s}. \tag{4.25}$$

This Cartesian Admittance Control provides a more intuitive way to control the robot in tasks that are spatially defined, ensuring smoother interactions with the environment. The following figure gives a visual representation of the Cartesian Admittance control strategy :

4.4.2 Cartesian Impedance Control

The design of the Cartesian Impedance Model follows the same steps as the joint-space control. The desired position and orientation of the end-effector in the Cartesian space, the **pose**, is expressed as X. The corresponding velocity and acceleration are denoted as as $\dot{X}$ and $\ddot{X}$, respectively. Starting with the impedance model in the Cartesian space, the

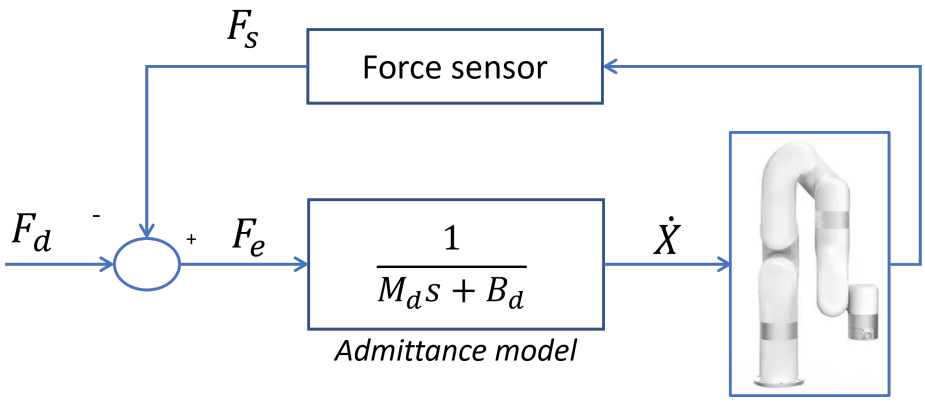


Figure 4.15 Cartesian admittance control using the force sensor

interaction force F_e demanded by the impedance controller is:

$$F_e = M_d\ (\ddot{x}_d - \ddot{x}) + D_d\ (\dot{x}_d - \dot{x}) + K_d\ (x_d - x). \tag{4.26}$$

The dynamic model of a robotic manipulator in the operational space is given by:

$$F = \Lambda(x)\ddot{x} + \mu(x, \dot{x}) + p(x) - F_e. \tag{4.27}$$

Here, $\Lambda(x)$ represents the Cartesian space inertia matrix, $\mu(x, \dot{x})$ captures the Coriolis and centripetal forces in Cartesian space, and $p(x)$ signifies the gravitational forces in the operational space.

By setting the desired inertia matrix M_d to be the robot's inertia matrix $\Lambda(x)$, and considering that we're operating under low velocities (rendering terms related to velocities negligible), the control law simplifies to:

$$F = p(x) - D_d\ (\dot{x}_d - \dot{x}) - K_d\ (x_d - x). \tag{4.28}$$

The following figure gives a visual representation of the Cartesian Impedance control strategy :

4.4.3 Inverse Kinematics

For both Cartesian control strategies, the resulting command is in Cartesian space. The Cartesian velocity command $\dot{X}_c$ cannot be sent directly through the robot API's. Therefore, a general Inverse Kinematics strategy is developed. First the equivalent joint velocities

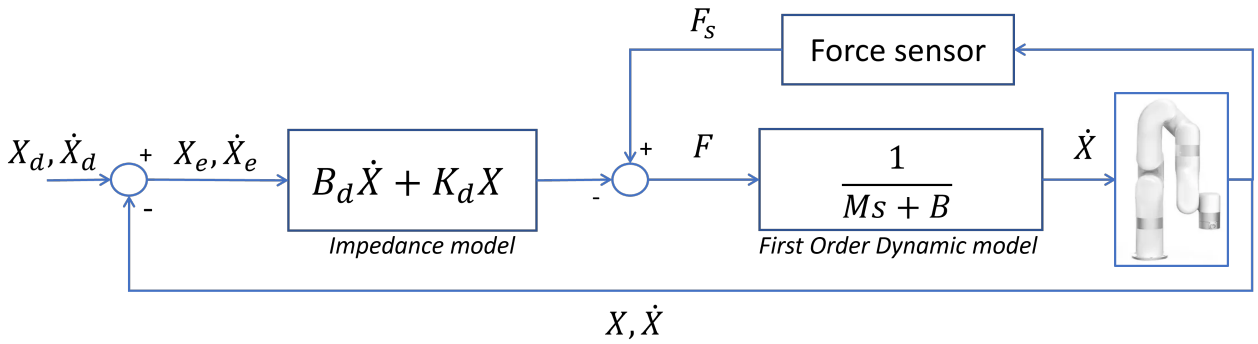


Figure 4.16 Cartesian Impedance control using the force sensor

command, $\dot{q_c}$ is computed using the Jacobian Pseudo-Inverse $J^{\dagger}$:

$$\dot{q_i} = J^{\dagger} \cdot \dot{X}_i. \tag{4.29}$$

This velocity command is then used to compute the new joint position using the following equation:

$$q_i = q_{i-1} + \dot{q_i} \cdot T_s, \tag{4.30}$$

where :

- q_i is the current joint position command,
- q_{i-1} is the previous joint position command,
- $\dot{q_i}$ is the current joint velocity command, and
- T_s is the time step of the control loop.

It is important to note that the Jacobian's pseudo-inverse $J^{\dagger}$ is computed using the **Damped Least Square Pseudo-Inverse Method**. Directly inverting the Jacobian matrix J is less feasible when approaching its singular or near-singular states. In such situations, even small variations in desired velocities can result in amplified joint velocities due to the ill-conditioned Jacobian. This method provides the follow advantages :

- Keeping joint velocities within bounds, especially close to singularities, leading to more controlled robot movements.

- Providing the capability to adjust between system stability and responsiveness through the damping factor.

The damped pseudo-inverse $J^{\dagger}$ is defined by:

$$J^{\dagger} = J^{T}(JJ^{T} + \lambda^{2}I)^{-1}. \tag{4.31}$$

Here, λ represents the damping factor, while I is an identity matrix of matching dimensions. Incorporating this method into the robotic rehabilitation system ensures controlled movements, especially near kinematic singularities. The precise value of the damping factor is typically informed by empirical tests, aiming for an equilibrium between system stability and responsiveness. Through trials and error, the selected damping factor was $\lambda = 0.001$. Figures 4.17 and 4.18 shows the implantation of the Inverse Kinematics for both Cartesian admittance and impedance control.

4.5 Singularity Avoidance

The challenges posed by singularities in robotic control are detailed in Section 2.9.2. To counteract these issues, one passive mechanism and two active methods have been integrated. The passive mechanism involves the use of the **Damped Least Squares Pseudo-Inverse** for inverse kinematics. The active methods, on the other hand, are based on the concept of the manipulability gradient, as presented in Section 2.9.3:

- **Null Space Control:** The xArm7, possessing 7 DOFs, allows the definition of a secondary task for the robot controller. Specifically, the DOFs that are not engaged in the primary task can be allocated to perform a **secondary task**. Employing the manipulability gradient, the secondary task is designed to avoid singularities, which is translated through a **null space joint velocity command** $\dot{q}_{NS}$.
- **Virtual Force Field:** Artificial force fields are frequently employed in robotics to promote certain behaviors, such as collision avoidance. In this context, the manipulability of the robotic manipulator was defined as a performance matrix to be optimized. By combining the performance metric with the manipulability gradient, **a virtual force,** F_v, is calculated to avoid singularities.

Figures 4.17 and 4.18 shows the different singularity avoidance mechanisms implemented for the Cartesian admittance and impedance control. The mechanisms are framed by the red dashed boxes :

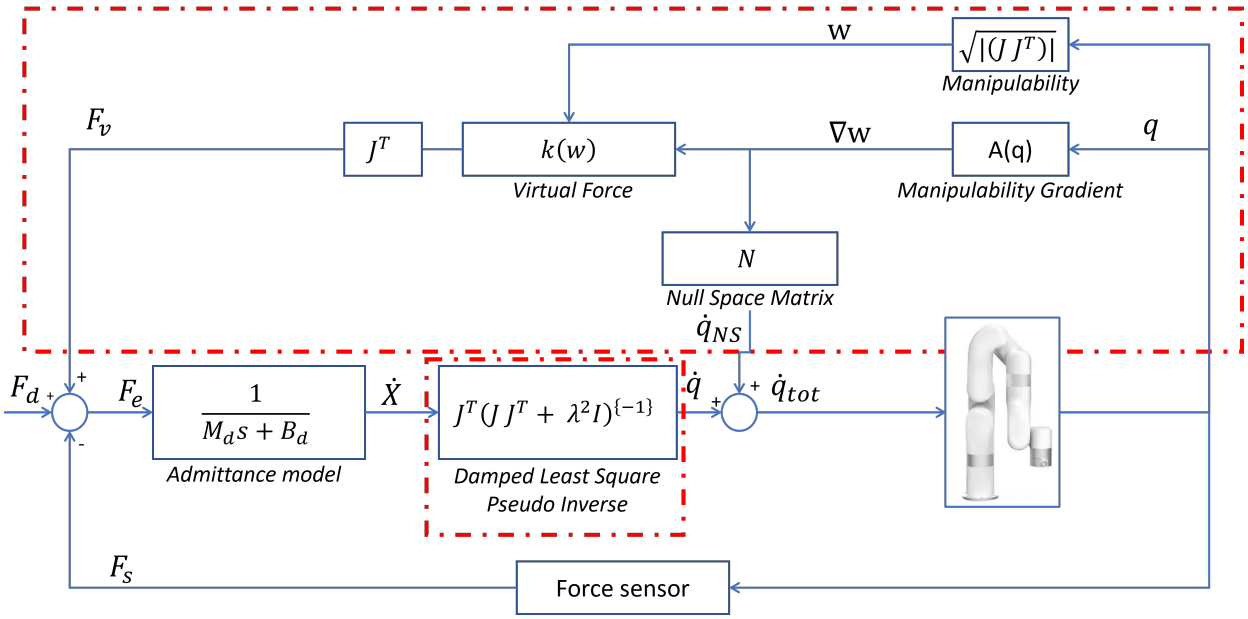


Figure 4.17 Cartesian admittance control loop with singularity avoidance mechanisms framed by the red dashed boxes. Here : w is the manipulability, ∇w is the manipulability gradient, and F_v is the virtual force

4.5.1 Manipulability Gradient

For efficient real-time control in robotics, calculating the manipulability gradient is essential. This study employs a machine learning technique, specifically a Multi-Layer Perceptron (MLP), to facilitate prompt and precise computations. The MLP, known for its effectiveness in approximating non-linear functions, is particularly suitable for this complex task.

The chosen Multi-Layer Perceptron (MLP) model is structured with two hidden layers: one with 100 neurons and another with 50 neurons. This architecture is designed to efficiently process complex, non-linear data relationships. The model incorporates the Rectified Linear Unit (ReLU) activation function, pivotal for introducing non-linearity and aiding quick learning during training phases by preventing the vanishing gradient problem.

For optimization, the 'adam' solver is employed. Known for its effective stochastic optimization, 'adam' enhances convergence speed and overall training stability by adapting learning rates based on the first and second moments of the gradients. This choice of solver, combined with the strategic layer configuration and ReLU activation, equips the MLP model with the robustness and adaptability necessary for its tasks. Figure 4.19 presents the structure of the MLP.

The procedure for training the MLP model involves several stages, outlined as follows:

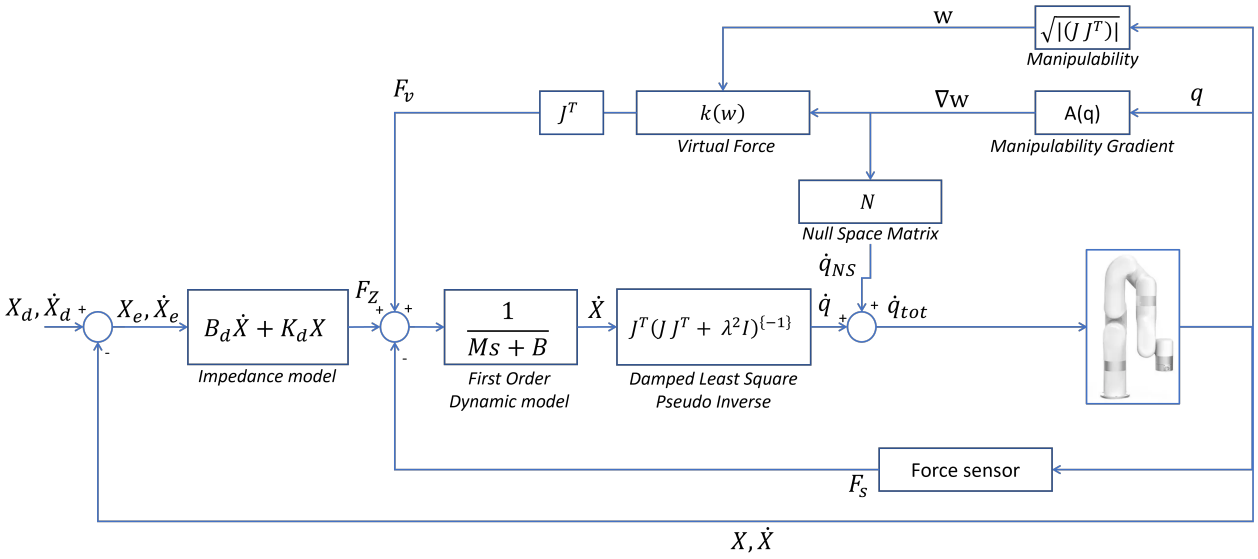


Figure 4.18 Cartesian Impedance control with singularity avoidance mechanisms

- Initially, a dataset was generated by creating 100,000 unique robotic arm configurations through random sampling within the joint limits defined by the manufacturer, referred to as q_i. **Note: The data is synthetically generated; the robotic device does not need to reach those specific configurations.**

- Subsequently, for each configuration q_i, the corresponding Jacobian matrix J_i was computed, leading to the determination of the manipulability measure w_i.**Note : The Jacobian matrix is computed *offline*.**

- The manipulability gradient was then calculated by perturbing each joint angle in q_i slightly and observing the resulting changes in w_i. This calculation was based on the expression $\nabla w_i = (\Delta w_i)/\Delta q_i$, indicating the responsiveness of the manipulability to changes in joint configurations.

- A dataset was compiled, including pairs of q_i and ∇w_i, to serve as the training data for the machine learning model. This dataset is crucial for the MLP to understand the relationship between the joint configurations and the corresponding manipulability gradients.

- The training phase involved the MLP model learning from the inputs (q_i) and the targets (∇w_i). This phase was pivotal for refining the model's ability to predict manipulability gradients accurately in new, unseen configurations.

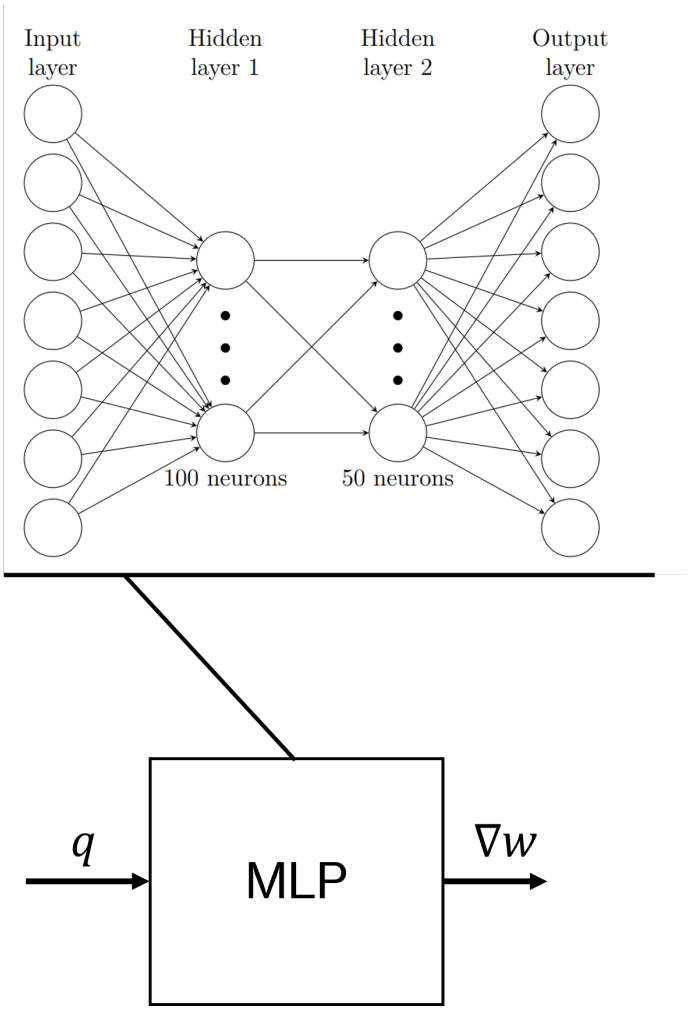


Figure 4.19 Visual representation of the MLP

This methodology ensures the MLP model is capable of generating reliable manipulability gradients without slowing down the control loop, a critical component for the real-time control system of robotic manipulators, enhancing their operational efficiency.

4.5.2 Null-Space Control for Singularity Avoidance

In this section, the Null-Space Control, which uses the unique properties of the Jacobian matrix to maintain the robot's main tasks while defining a secondary task control, is used for singularity avoidance. The null space matrix is computed using the following equation :

$$N = I - J^{\dagger}J. \tag{4.32}$$

The secondary task velocity command ($\dot{q}_N$), to avoid singularities, is directly computed using the gradient of manipulability from the machine learning algorithm :

$$\dot{q}_{\mathrm{N}} = A(q). \tag{4.33}$$

Finally, the joint velocity command $\dot{q}$ is computed as follow :

$$\dot{q}_{\mathrm{c}} = \dot{q}_{\mathrm{task}} + N \cdot \dot{q}_{\mathrm{N}}. \tag{4.34}$$

Here, $\dot{q}_{task}$ represents the joint velocity command from the original control loop, such as the joint velocity command resulting from the admittance controller and $\dot{q}_c$ is the velocity command.

4.5.3 Virtual Force/Torque Computation for Singularity Avoidance

In this section, the computation of virtual forces and torques, applying adjustments based on the manipulability measure's gradient is presented to avoid singularities. This technique ensures real-time trajectory corrections, preventing the manipulator to reach singularity configurations.

The manipulability measure, $w(q)$, is first calculated using the formula:

$$w(q) = \sqrt{\det(J(q)J(q)^T)}, \tag{4.35}$$

where $J(q)$ is the Jacobian matrix of the manipulator, reflecting how changes in joint space affect the end-effector's velocity. A low $w(q)$ signifies a state close to singularity.

When $w(q)$ falls below a predefined threshold, w_{th}, a corrective scalar factor, $k(w)$, activates:

$$k(w) = \begin{cases} \lambda \left(\frac{1}{w(q) - w_{cr}} - \frac{1}{w_{th} - w_{cr}} \right) & \text{if } w(q) \leq w_{th} \\ 0 & \text{otherwise} \end{cases}, \tag{4.36}$$

where λ is a constant, and w_{cr} is the critical lower limit of $w(q)$, below which the system is in singularity. This factor $k(w)$ serves as a repulsive force guiding the system away from undesired configurations. The function $k(w)$ for the values of : $\lambda = 0.4; w_{cr} = 0.006; w_{th} = 0.045$ is represented in Figure 4.20.

The gradient of the manipulability, $A(q)$, is then computed,using the MLP, to understand the direction of required adjustments in joint space:

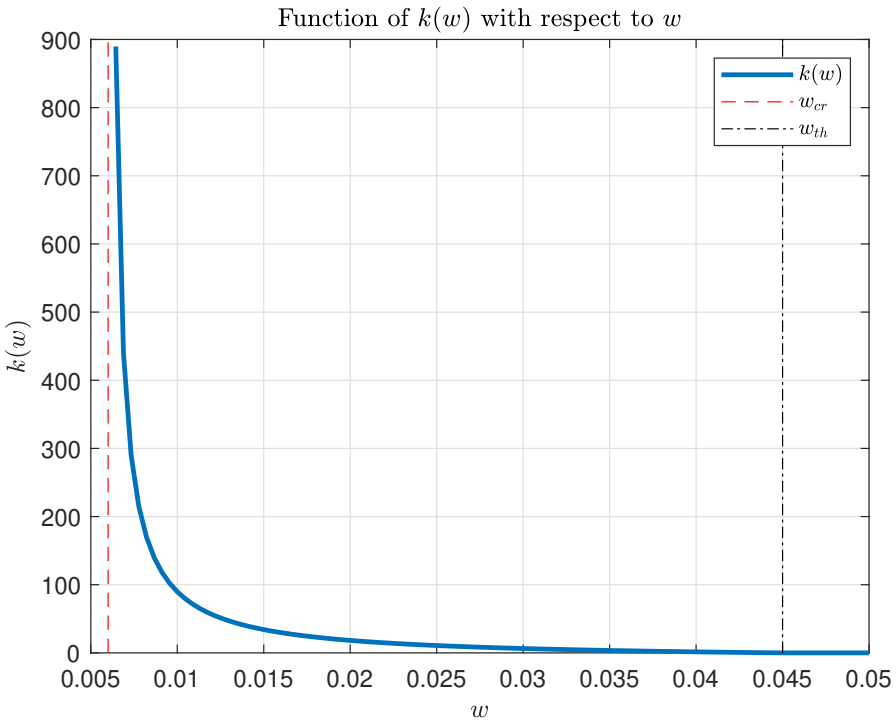


Figure 4.20 Graphical representation of the function $k(w)$, showcasing the scaling behavior as the manipulability measure approaches critical singularity points. The regions corresponding to w_{cr} and w_{th} are distinctly highlighted

$$A(q) = \frac{\partial w(q)}{\partial q}. \tag{4.37}$$

The final step involves calculating the virtual force/torque, F_v, which informs the adjustments in the physical forces/torques at the robotic joints:

$$F_v = J^T \cdot k(w) \cdot A(q). \tag{4.38}$$

4.6 Design of Experiments

This section outlines the experiments conducted to analyze the system's behavior under various operational scenarios. An important aspect of these experiments is the 'compliance vector' used in Cartesian Admittance Control. The compliance vector is a binary array that indicates the active degrees of freedom (DOF) in the control loop. It specifically identifies which translations or orientations are subject to control inputs.

For instance, a compliance vector represented as

$$C = [1, 1, 1, 0, 0, 0], \tag{4.39}$$

in Cartesian space activates the X, Y, and Z translations, while the orientations in X, Y, and Z remain inactive. This selection is crucial in the admittance control setup as it determines the DOFs that respond during control.

In the context of the experiments, the term *perturbation* refers to an external force applied to the robotic device by the user. Perturbations are applied randomly and in different directions. It is important to note that this is possible because the force sensor allows for the characterization of both the magnitude and direction of the *applied perturbation*.

4.6.1 Experiment 1 : Trajectory Generation using Cartesian Admittance Control

The experiments involved using five different compliance vectors. Each of these vectors was used as a basis for generating conditions for trajectory formation, influencing the DOFs that the control loop manages.The compliance vectors were defined to replicate potential use cases for rehabilitation. The first two compliance vectors were defined to assess the translational and rotational DOFs separately. Compliance vectors 3 to 5 were chosen to simulate potential passive exercises performed along a plane, such as passive shoulder extensions. Lastly, the final vector was selected to emulate ADL exercises commonly employed in rehabilitation.

For each compliance vector, five trajectories were randomly constructed using Cartesian Admittance Control. These trajectories were essential in testing the control system's response under different conditions. The following compliance vectors were used:

- **Compliance Vector 1:** [1, 1, 1, 0, 0, 0] - This configuration activates the X, Y, and Z translations, keeping the orientations inactive.
- **Compliance Vector 2:** [0, 0 0, 1, 1, 1] - This configuration activates the X, Y, and Z orientation, keeping the translation inactive.
- **Compliance Vector 3:** [1, 1, 0, 0, 0, 1] - This setup enables the X and Y translations and the Z orientation, with the remaining axes inactive.
- **Compliance Vector 4:** [1, 0, 1, 0, 1, 0] - This configuration activates the X and Z translations and the Y orientation, excluding others.

- **Compliance Vector 5:** [0, 1, 1, 1, 0, 0] - This setup involves the Y, Z translations, and X orientation, leaving out the rest.
- **Compliance Vector 6:** [1, 1, 1, 1, 1, 1] - This configuration represents a fully active state, with all translations and orientations enabled.

During these experiments, several metrics were recorded to provide a detailed analysis of the system's performance. These include:

- Joint positions and velocities,
- Cartesian positions and velocities,
- Null-space command velocities, and
- Virtual force/torque used to prevent singularities.

The collection of these data points contributes to a comprehensive analysis, helping to assess the control strategies implemented. The variety in trajectory and compliance vectors ensures a thorough evaluation, documenting the system's reactions in different scenarios and control settings.

4.6.2 Experiment 2 : Effect of Impedance Parameters

This experiment evaluated how different impedance parameters—stiffness and damping—affect the robot's performance. These parameters affect the robot's response to external forces, trajectory tracking performance and influences its stability and adaptability.

The robot was programmed to reach 10 unique target configurations. For each configuration, 16 impedance parameter settings were used, as detailed in Table 4.6. These settings adjusted the levels of stiffness and damping, ranging from 'Very High' to 'Low' and 'None' to 'High,' respectively. Notably, a setting with no stiffness was avoided to ensure the robot could still move, and an extremely high damping setting was excluded to ensure the system stability.

This experiment aimed to discern how various impedance configurations affected the robot's operational effectiveness, seeking the most favorable settings for different use cases.

4.6.3 Experiment 3 : Response to External Perturbations

This experiment aimed the robot's reaction to external perturbations. This is employed to simulate potential interactions of a patient with the manipulator. The robot held a constant

Table 4.6 Combinations of impedance parameters used during testing

Stiffness \Damping	None	Low	Moderate	High
Very High	Setup 1	Setup 2	Setup 3	Setup 4
High	Setup 5	Setup 6	Setup 7	Setup 8
Moderate	Setup 9	Setup 10	Setup 11	Setup 12
Low	Setup 13	Setup 14	Setup 15	Setup 16

position, while external perturbations were applied to the end-effector. Two impedance settings, Z_d, were chosen from the initial experiment's findings, as shown in Table 4.7.The first set, known as $Z_{d,high}$, had higher values, indicating a more rigid system. In contrast, the second set, referred to as $Z_{d,low}$, had lower values, suggesting a more flexible approach.

Table 4.7 Impedance settings for response testing with external perturbations

Experimental Setting	**Stiffness**	**Damping**
$Z_{d,high}$	High	Moderate
$Z_{d,low}$	Low	Moderate

This experiment allows to understand the behavior of the robotic devices when facing external perturbations for different desired impedance parameters.

4.6.4 Experiment 4 : Impedance Control Evaluation with and without Perturbations

The experiment focused on assessing the tracking accuracy and adaptability of the robotic manipulator using impedance control in different conditions. The two sets of impedance parameters presented in 4.7 were used. Five reference trajectories were established for the manipulator to follow. Each trajectory was followed, with and without perturbations, using both the $Z_{d,high}$ and $Z_{d,low}$ parameter sets. In total, each trajectory was tracked four times, twice in a controlled environment with no disturbances and twice in a setting with external disturbances. Both high and low impedance settings were applied in each type of environment.

This method allowed for a direct comparison between the high and low impedance configurations in different scenarios. The main goal was to observe the differences in tracking accuracy and the system's ability to adapt when disturbances were present. The results would help determine which impedance parameter set, high or low, is more effective in maintaining accurate trajectory tracking when external perturbations are encountered.

4.6.5 Experiment 5 : Adaptive Control for Rehabilitation Purposes

The experiments aimed to asses the effectiveness of the control **adaptive control strategy** in rehabilitation exercises. Four exercises were included in this phase, with three being passive movements - shoulder extension, shoulder abduction, and horizontal shoulder flexion - and one being an active daily living task (ADL), mimicking the action of "Taking a Can off a Shelf." The passive exercises are shown on Figure 4.21.

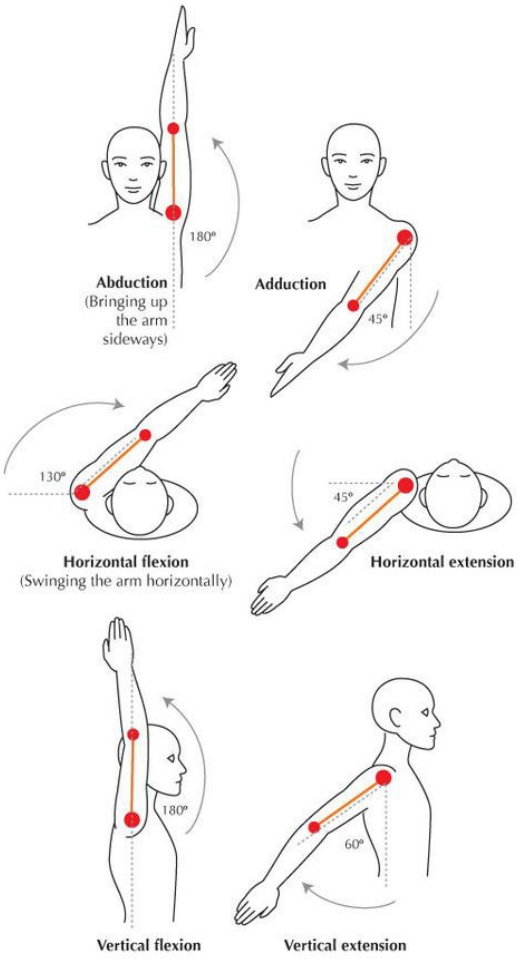


Figure 4.21 Shoulder joint passive exercises [6]

The exercises were conducted in a structured manner, detailed as follows:

1. Initially, the robot performed the specific exercise trajectories with the participant's arm attached. This phase was repeated four times for each exercise, applying the appropriate impedance parameters. For passive movements, $Z_{d,passive}$ parameters were used, and for the active task, $Z_{d,active}$ parameters were chosen, assuring an interaction

suitable for each type of exercise.

2. Next, the adaptive impedance control approach was used. This strategy allowed the robot to adjust its parameters dynamically based on real-time feedback from the user's interactions. Each exercise trajectory was repeated four times using this adaptive control, highlighting the robot's ability to adjust and maintain precision according to the specific requirements of each exercise.

Note: Only one healthy user conducted the experiment. The participant's role during the various exercises was to adjust the level of activity of their limb. For passive movements, the user was instructed not to exert any force, allowing their arm to move freely. In contrast, for the Activities of Daily Living (ADL) task, active participation was required, with the user following the reference trajectory. This setup was crucial in assessing the adaptive control system's ability to effectively identify and respond to varying degrees of user participation.

CHAPTER 5 RESULTS AND DISCUSSION

5.1 Results

5.1.1 Experiment 1 : Trajectory Generation

In this experiment,the Cartesian Admittance Controller was used with the singularity avoidance strategies presented in the methodology section. This approach facilitated the generation of trajectories for various degrees of freedom (DOFs), defined by different compliance vectors.

Visual Representation of Trajectory Generation using the Admittance Control

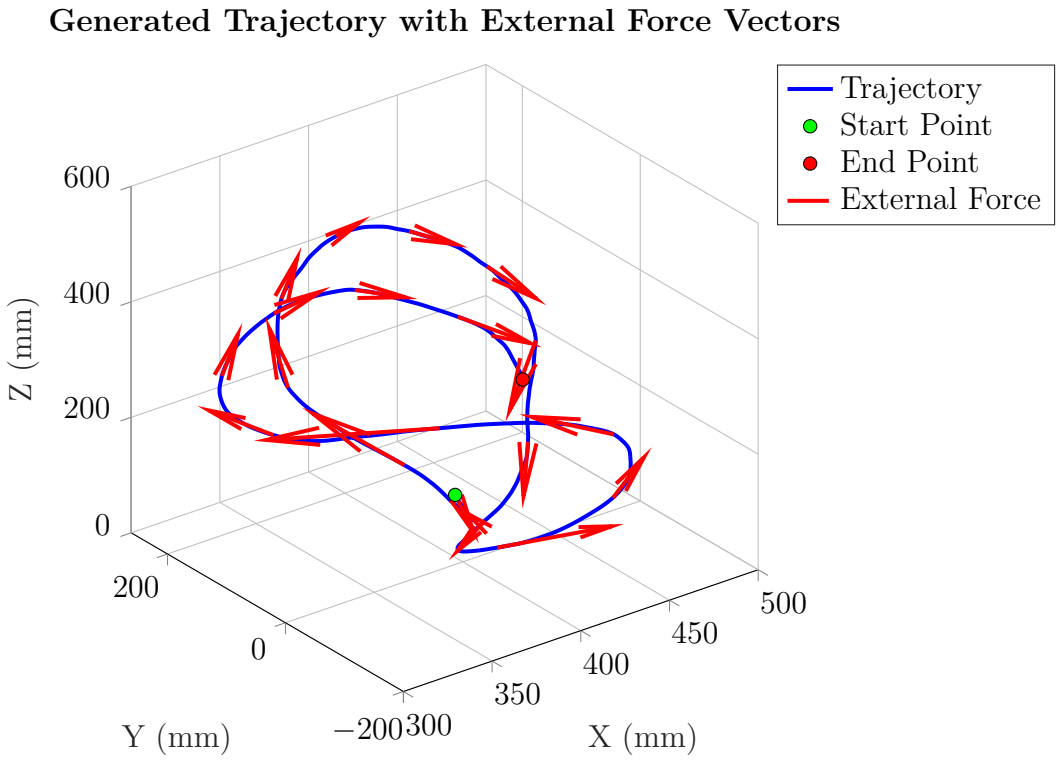


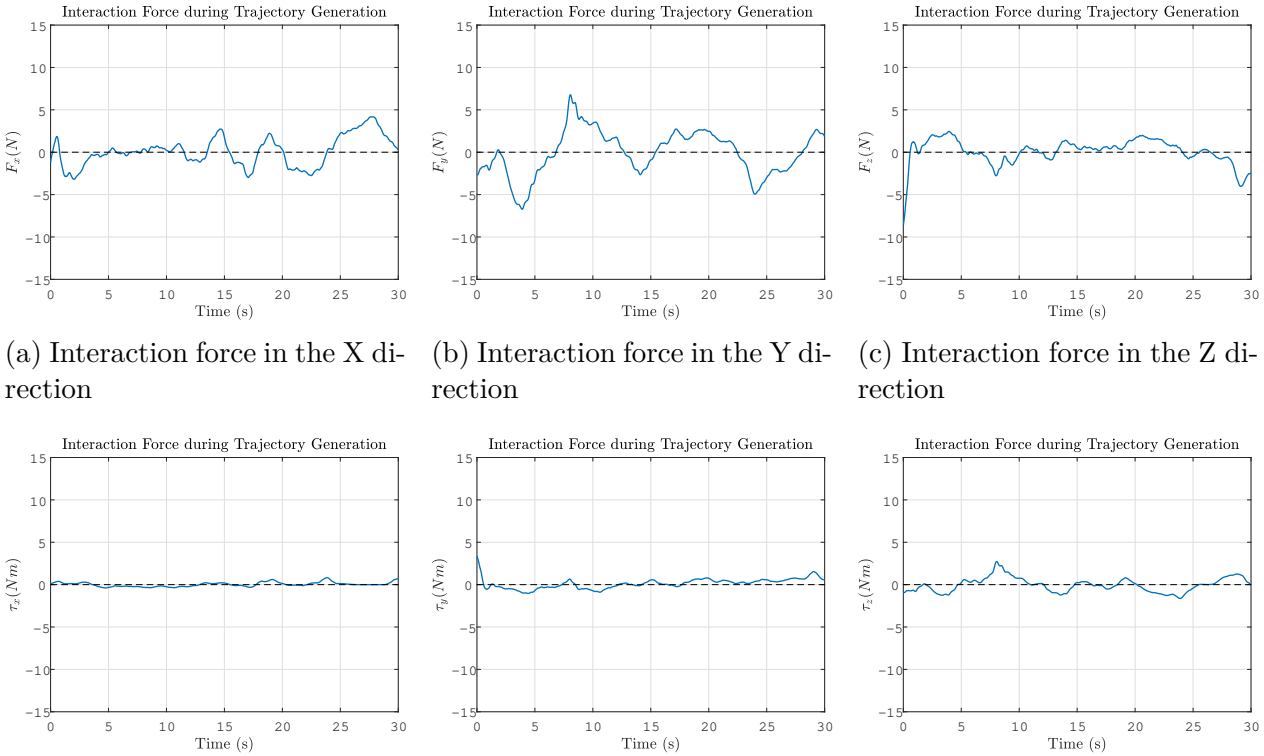
Figure 5.1 Generated trajectory using the Cartesian Admittance Controller

Figure 5.1 shows the trajectory generated from an experiment employing a compliance vector of [1,1,1,0,0,0]. The illustration emphasizes both the trajectory and the external forces at recorded between the initial and final position. Similar results are observed for every trial.

Interaction Force during Trajectory Generation

During trajectory generation using the admittance controller, the **interaction force** between the robotic manipulator and its environment is set to be **null** $(F_d = 0)$. Therefore, when the user interacts with the robot, the device is constantly trying to compensate the interaction force by moving away. Figure 5.2 shows the interaction force recorded throughout the trajectory generation, using the fully compliant vector $(C = [1, 1, 1, 1, 1, 1])$.

Throughout the trajectory development phase, the admittance controller maintains the interaction force with the environment at zero $(F_d = 0)$ for the robotic manipulator. Therefore, any physical user interference prompts the robot to adjust its position, effectively negating the applied force. Figure 5.2 captures the interaction forces throughout the trajectory formation with a fully compliant vector $(C = [1, 1, 1, 1, 1, 1])$.



(a) Interaction force in the X direction

(b) Interaction force in the Y direction

(c) Interaction force in the Z direction

(d) Interaction torque in the Roll direction

(e) Interaction torque in the Pitch direction

(f) Interaction torque in the Yaw direction

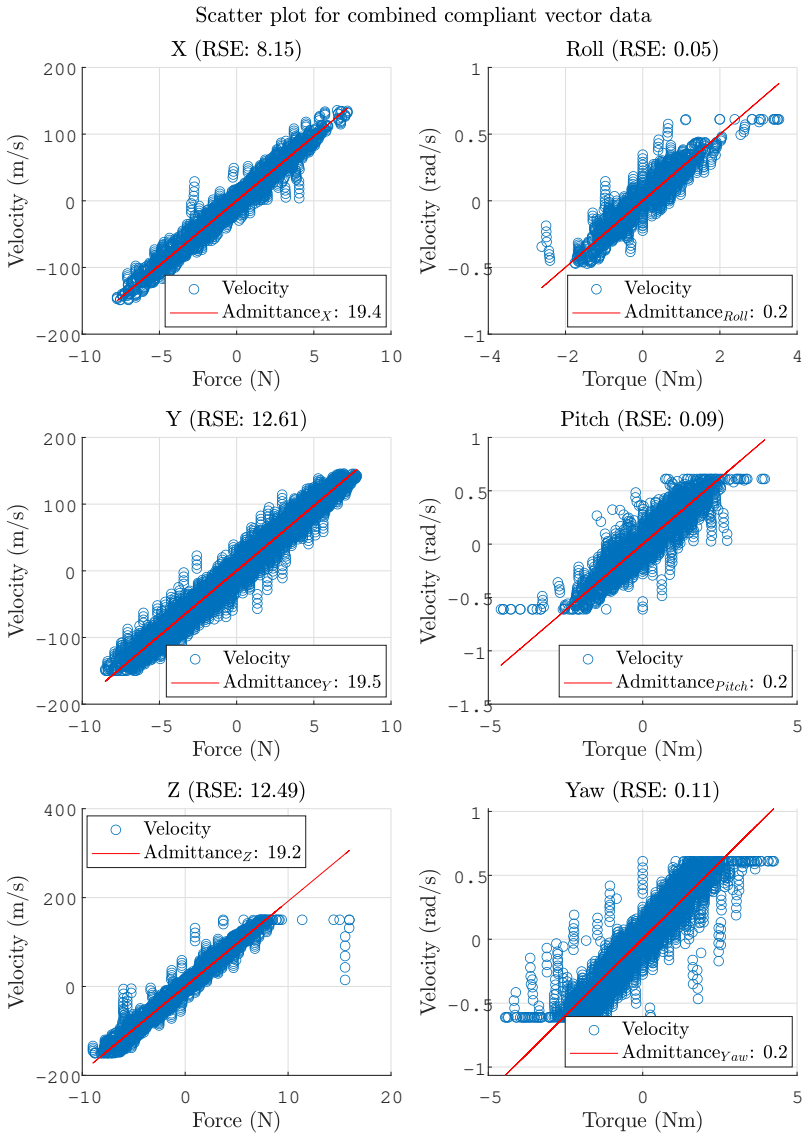
Figure 5.2 Interaction forces during trajectory generation for different degrees of freedom (DOFs)

Interaction Force and Velocity in Cartesian Admittance Control

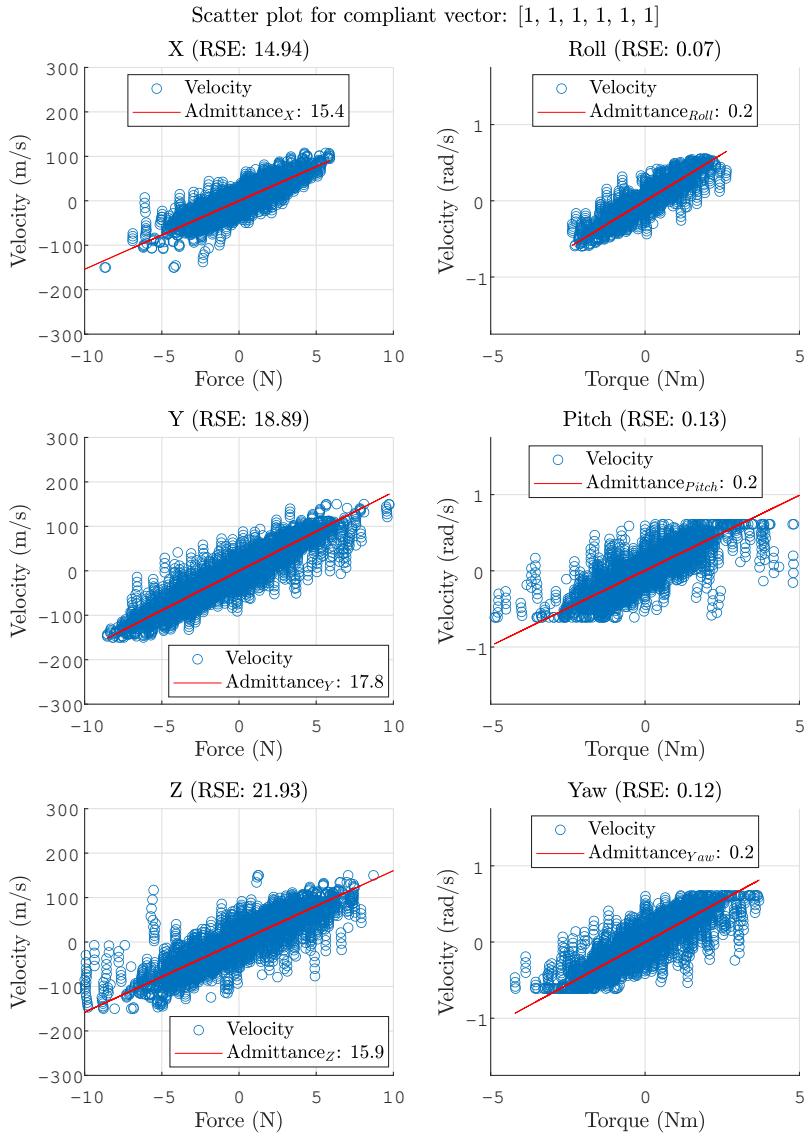
Figure 5.3 demonstrates the relationship between the external forces and the resulting velocities under different compliance vector. In fact, Figures 5.3a and 5.3b show the velocity command with respect to the interaction force, highlighting how force influences velocity in both translational and orientational DOF. The analysis involved computing the Residual Standard Error (RSE) to estimate the accuracy of the linear fit, which represents the admittance measure ($\frac{1}{Z}$) of the admittance model, between the interaction forces and the resulting velocities. The RSE represents the standard deviation of the residuals, which are the differences between the observed velocities and the velocities predicted by the linear model. This error estimation is crucial as it provides insight into the variability, indicating the consistency of the robot's response to external forces. For each direction component, a linear fit was applied to the scatter plot of the external forces versus the resulting velocities. The residuals were calculated as the differences between the actual velocities and those predicted by the linear model. The RSE was then computed as follows:

$$RSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(v_i - \hat{v}_i)^2} \tag{5.1}$$

where v_i represents the observed velocities, $\hat{v}_i$ are the velocities predicted by the linear model, and n is the number of observations. This metric was calculated for each compliance vector across all experiments.



(a) External forces and velocities for compliance vector [1,1,1,0,0,0] (X,Y,Z), and compliance vector [0,0,0,1,1,1] (RPY)



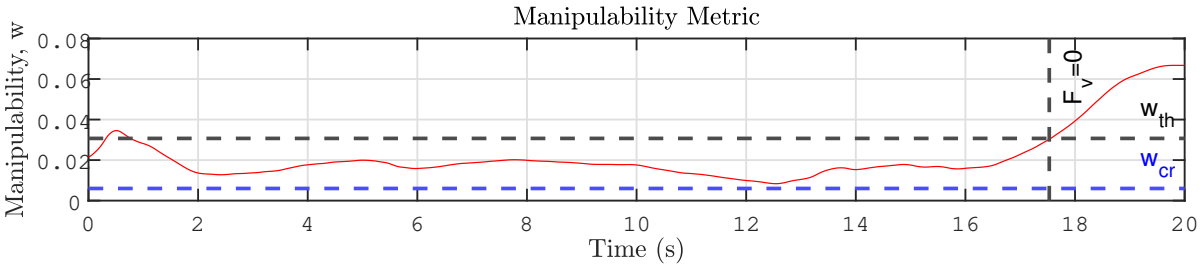
(b) External forces and velocities for compliance vector [1, 1, 1, 1, 1, 1]

Figure 5.3 Force-Velocity Relationship in Cartesian Admittance Control: This figure illustrates the relation between external forces and velocities command during trajectory generation using the Cartesian admittance controller. The calculated slope, a fit through the data points, represents the compliance level of the admittance model

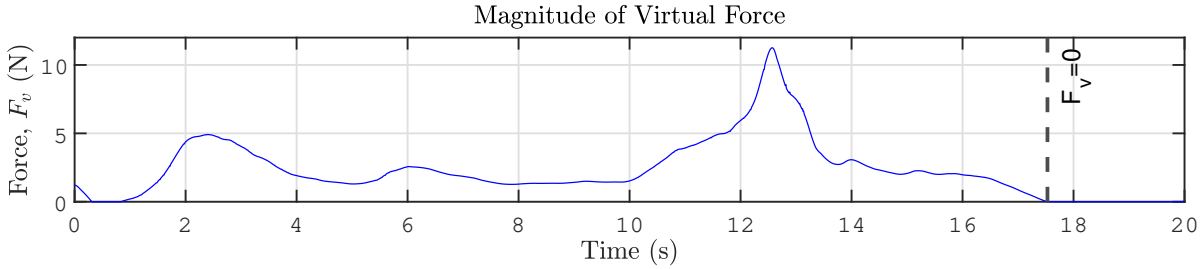
Singularity Avoidance Active Mechanism

To optimize the manipulator's functionality, two active singularity avoidance techniques were implemented. The first, Null-Space control, computes secondary joint velocity for singularity avoidance. Simultaneously, the Virtual Force method introduces a force vector, guiding the manipulator away from potential singularity configurations.

Figure 5.4 demonstrates the relationship between the manipulability of the robotic device and the corresponding virtual force applied during the trajectory generation. Subfigure 5.4a illustrates the manipulability metric over time, highlighting the critical manipulability value (w_{cr}) and the manipulability threshold (w_{th}). It is evident that as the manipulability decreases, approaching the critical value, there is an increase in the virtual force magnitude exerted to avoid potential singularities. In fact, the maximum magnitude is atteined when the manipulability is closer to the critical value. On the other hand, as the manipulability increases, this results in a reduction of the magnitude of the virutal force. Note that the virtual force is equal to zero once the manipulability is over the threshold value ($F_v = 0$). This can be observed at $t = 18(s)$ on the figure. This phenomenon highlights the system's self-regulatory measures to maintain operational efficiency and safety.



(a) Manipulability during trajectory generation, demarcating critical and threshold values



(b) Corresponding virtual force dynamics in response to manipulability variations

Figure 5.4 Analysis of manipulability and virtual force during robotic trajectory generation, highlighting the active mechanism of the robotic system to avoid singularity configurations

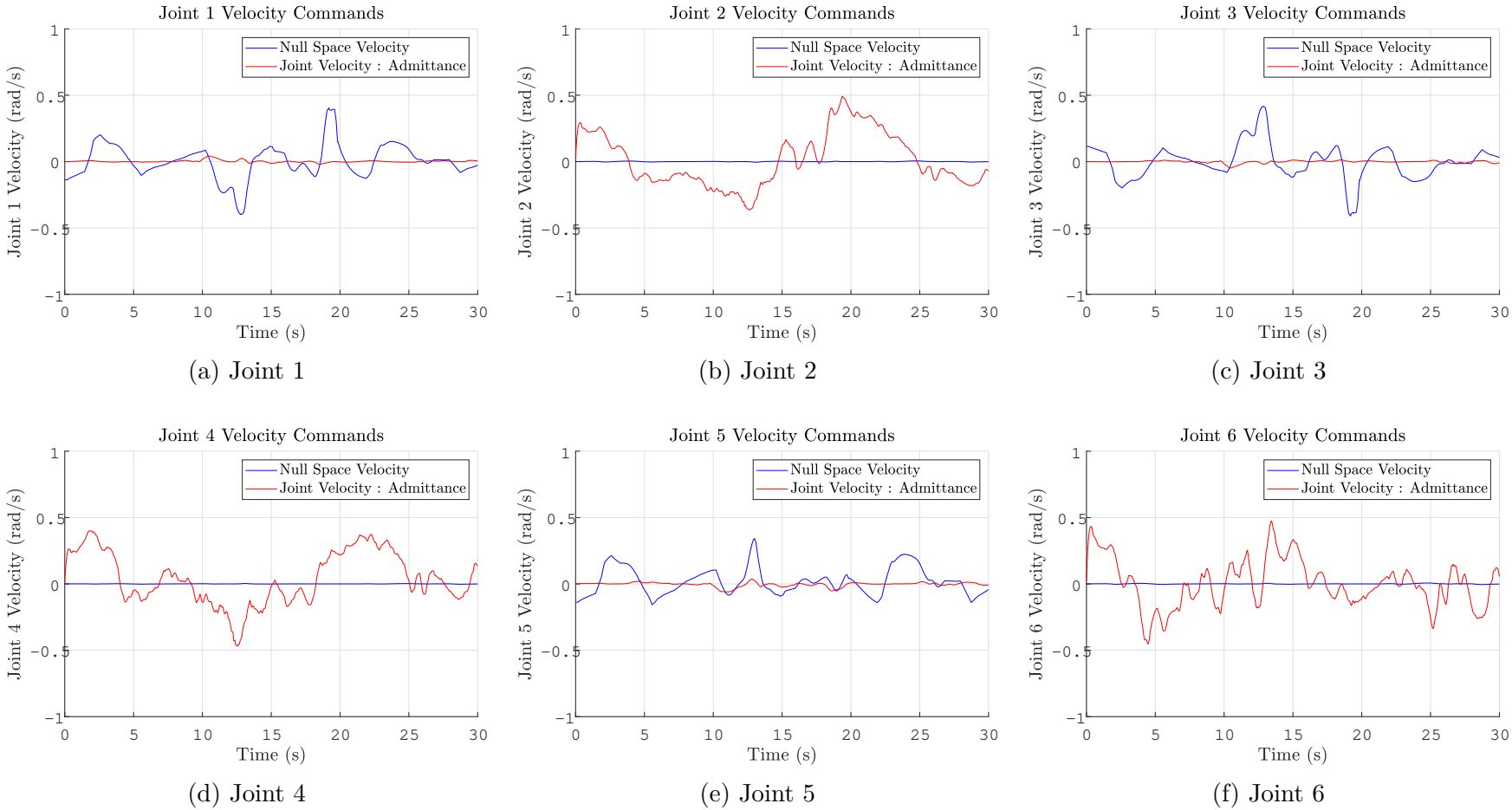
Figure 5.5 shows the behavior of each joint under null space and admittance control com-

mands, offering insights into their functions during robot operation.

In the figure, joints 2, 4, and 6 have no null space commands, indicating that these joints are crucial for the robot's primary task. This importance is evident as these specific joints receive high admittance commands, directing them to perform the main task (see subfigures 5.5b, 5.5d, and 5.5f).

On the other side, joints 1, 3, and 5 display low or virtually no admittance commands, which suggests that these joints are not actively involved in the primary task. Instead, they receive significant null space commands, highlighting their role in secondary tasks or adjustments that don't affect the main operation (refer to subfigures 5.5a, 5.5c, and 5.5e). Note that the results from joint 7 were consistent with the presented data and have been excluded for brevity and clarity in the graphical representation.

This distinction in the joint commands helps us understand how the robot prioritizes the main task (trajectory generation) during task execution, demonstrating an organized allocation of joint roles.



(a) Joint 1 (b) Joint 2 (c) Joint 3

(d) Joint 4 (e) Joint 5 (f) Joint 6

Figure 5.5 Velocity commands for each joint, comparing the null space velocity and the admittance joint velocity commands

5.1.2 Experiment 2: Impact of Impedance Parameters on Performance

This subsection discusses the results from Experiment 2, focusing on the role of impedance parameters in the robot's joint-space impedance controller. The experiment involved moving the robot from a starting point to a target position. The performance during these movements was evaluated based on several criteria.

The metrics used for evaluation included:

- **Response Time**: This is calculated as the time it takes for the joint position to stabilize within a range of $+/-2\%$ of the target position. It is a measure of how quickly the system responds to the target.
- **Overshoot**: This represents the maximum peak value recorded beyond the desired position and is expressed as a percentage. It indicates the control system's tendency to exceed the target.
- **Position Error**: This is determined by the difference between the final stabilized position and the desired target, represented in percentage points. It assesses the accuracy of the system in reaching and maintaining the target position.

These metrics are crucial for understanding the system's responsiveness, precision, and stability under various impedance settings. Figure 5.6 demonstrates the step response of the system. Each response curve corresponds to a specific set of control parameters, showing the system's reactions under different conditions. It is evident from the diverse response behaviors that the impedance parameters significantly influence the system's performance. Figure 5.7 shows the results, providing a comprehensive view of performance across all test conditions. By analyzing these metrics, it can observe how changes in impedance settings affect each joint's performance and the system's overall stability and accuracy.

The insights gained from these experiments are vital for determining the optimal impedance parameters, contributing to improved control strategies in robotic tasks.

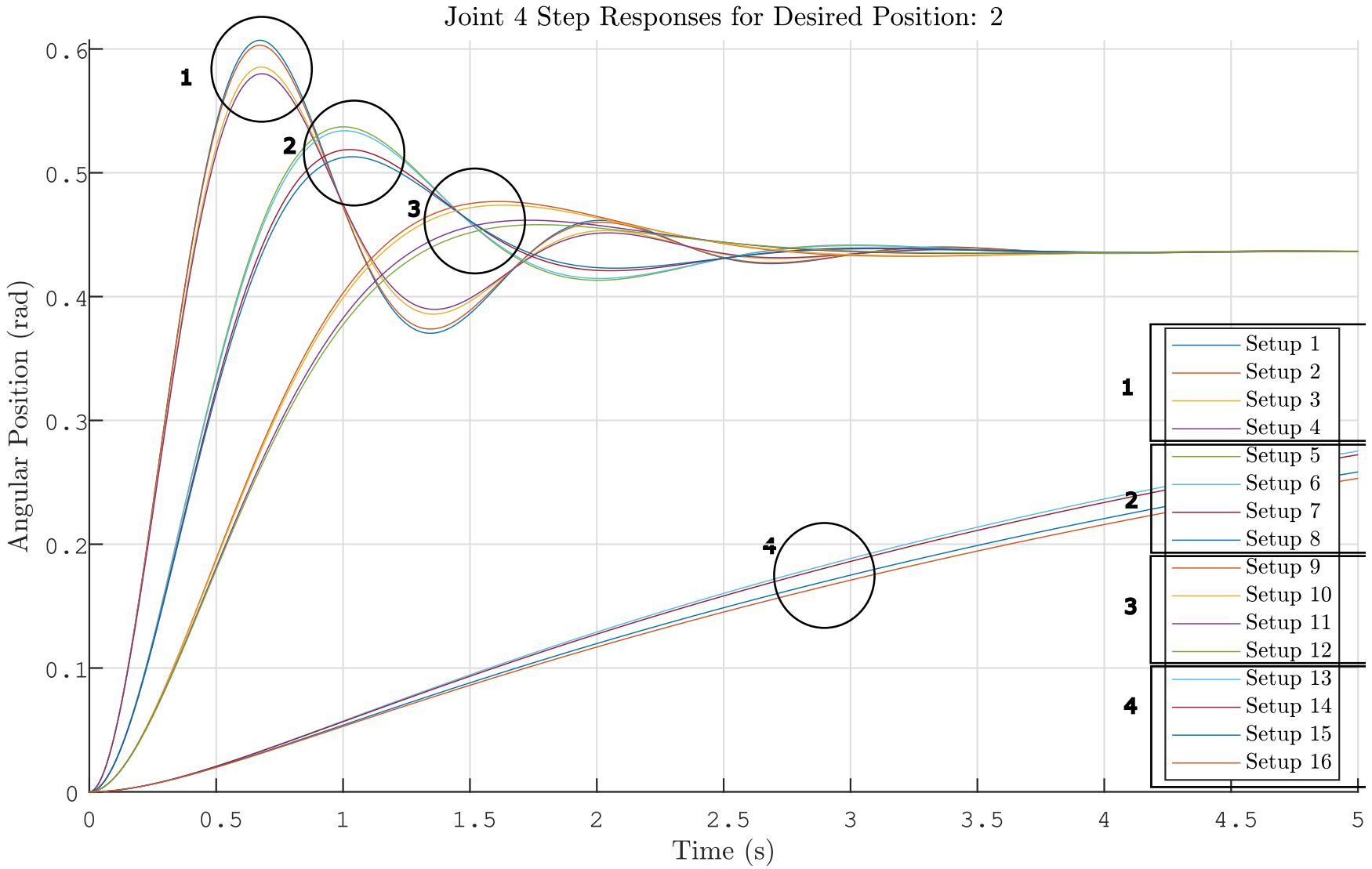


Figure 5.6 Analysis of 4th Joint Step Response: The figure displays the step response of the robot's 4th joint, grouped into four categories based on shared response characteristics. These responses help understand the impact of different impedance control settings on the joint's movement behavior

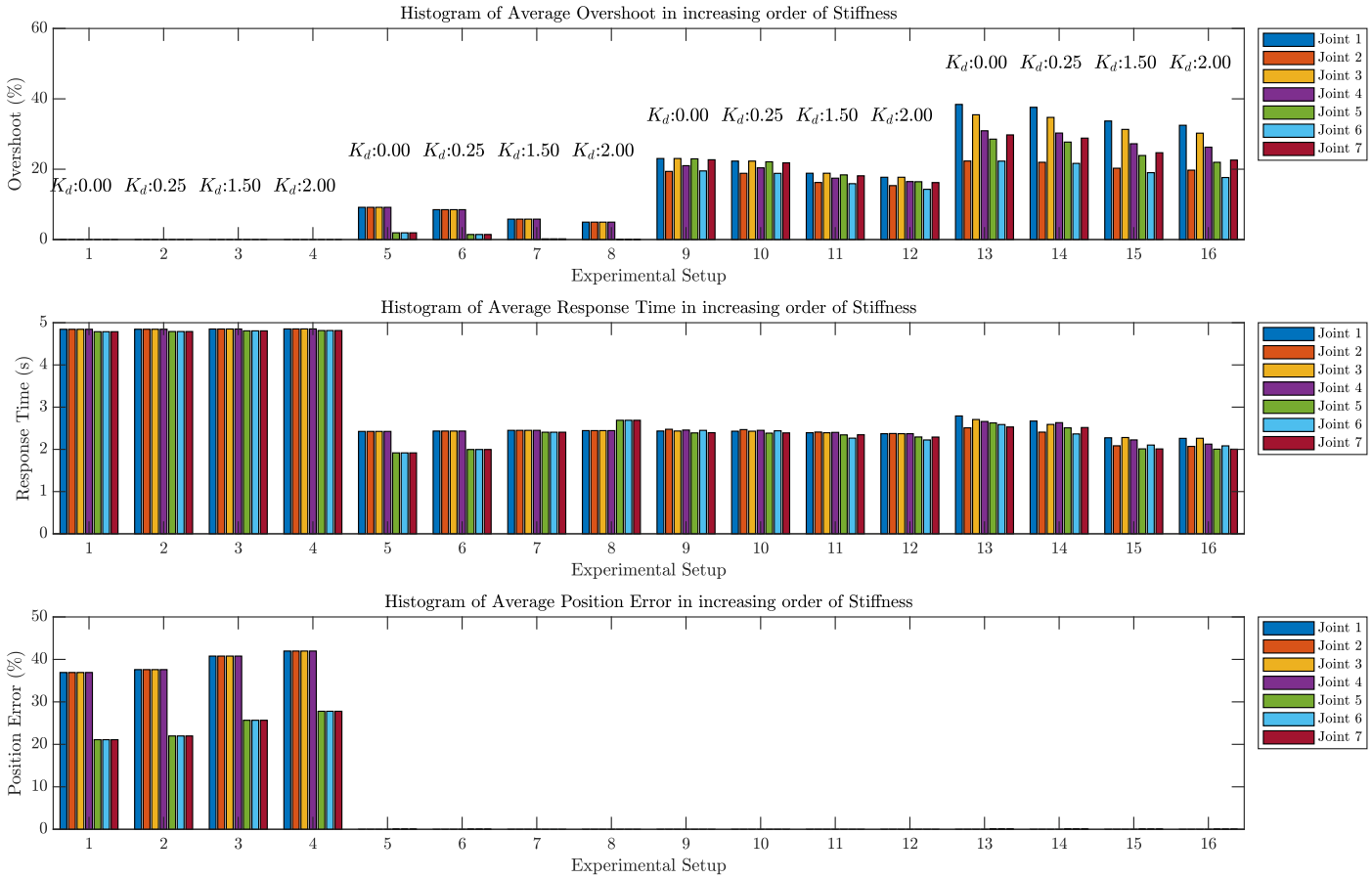


Figure 5.7 Performance Metrics Overview: This figure summarizes the performance metrics, including Overshoot(%), Response Time(s), and Position Error(%), across different experimental conditions. The metrics, broken down by joint, draw attention to the relationship between impedance settings, particularly stiffness (K_d), and overall system behavior in task execution

5.1.3 Experiment 3: Analyzing Response to External Perturbations

Experiment 3 is designed to evaluate the performance of the robotic manipulator when it encounters external perturbations. The control strategy employed here is Cartesian impedance control. The experiment involves applying different impedance parameters, specifically $Z_{d,low}$ and $Z_{d,high}$, to understand how the system's position error dynamics react when an external force is applied. The primary measure of interest is how effectively the control strategy manages to maintain the designated position amidst these disturbances.

Figure 5.8 presents a critical comparison between the two different impedance settings. Initial observations indicate that the system exhibits a higher position error when the impedance is low. This error decreases at a gradual rate. In contrast, with high impedance settings, the system not only experiences a generally lower position error, but the compensation mechanism also acts more rapidly, promptly restoring the position error back to zero. This behavior underscores the influence of impedance parameters on the system's resilience to external forces and its ability to maintain a steady position. The rapid recovery with $Z_{d,high}$ suggests a more robust control mechanism in scenarios where disturbances occur.

These observation allows to understand the manipulator's behavior under different levels of impedance, allowing to optimize the control strategies by adjusting the parameters while considering the real-world applications. In the use case of therapy, one should consider the patient recovery stage and the type of exercise to define the appropriate impedance parameters.

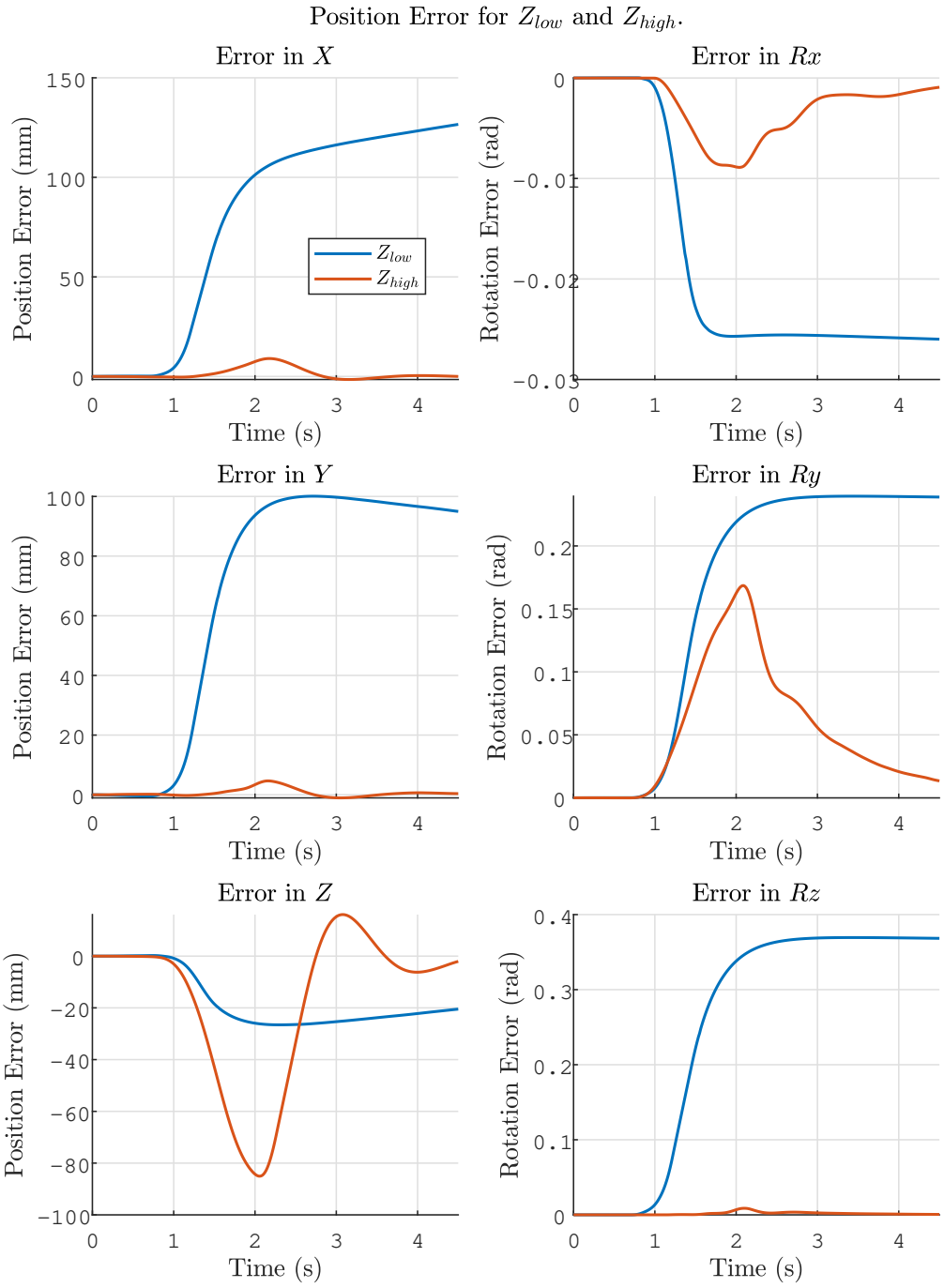


Figure 5.8 Comparative Analysis of External Perturbation : This graph depicts the response of the position error under Cartesian impedance control when subjected to external perturbations, contrasting the responses between $Z_{d,low}$ and $Z_{d,high}$ settings

5.1.4 Experiment 4: Trajectory Tracking Using Cartesian Impedance Control

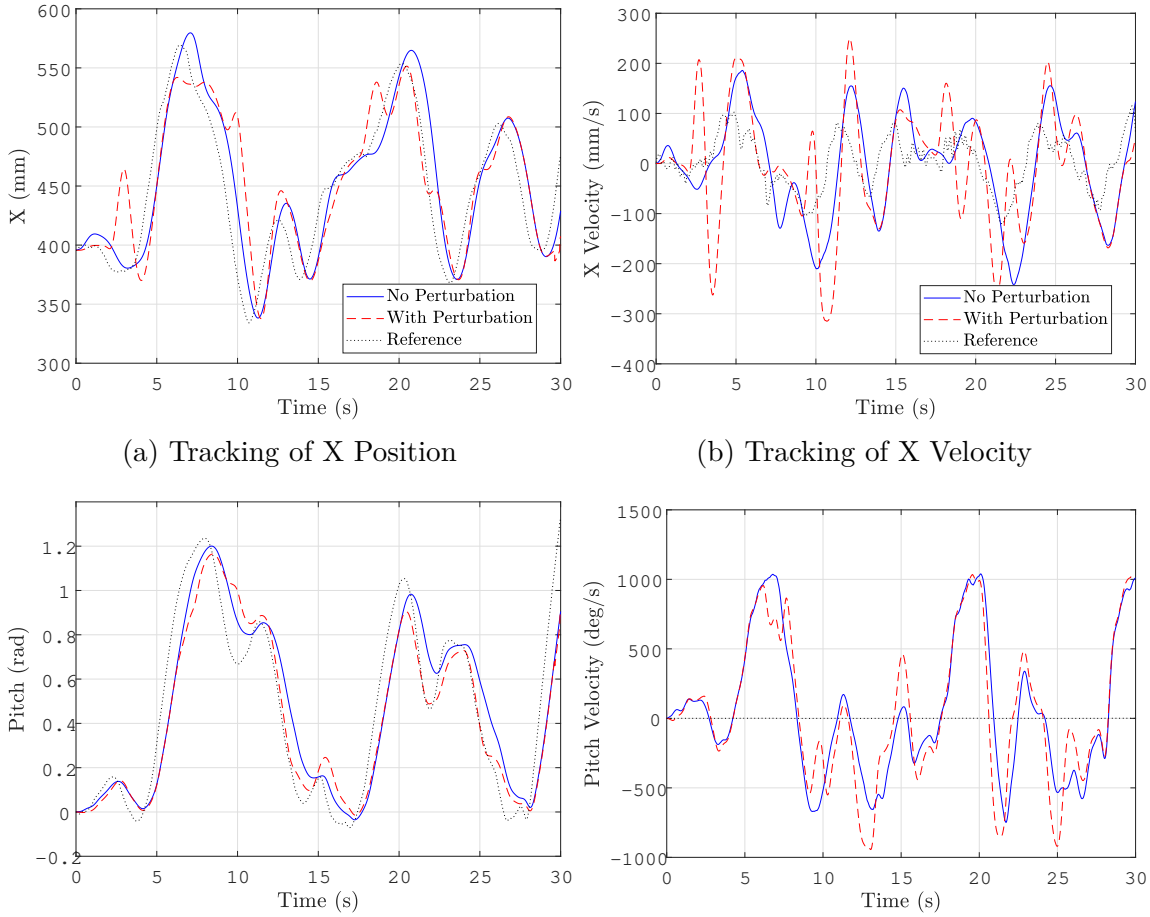
This section shows the results from the trajectory tracking experiments conducted with the Cartesian Impedance Controller under two distinct settings: $Z_{d,high}$ and $Z_{d,low}$. The experiments were designed to assess the controller's performance in maintaining the predefined trajectory, both in the presence and absence of external disturbances.

Figures 5.9 and 5.10 illustrate the performance outcomes for the $Z_{d,high}$ setting. These visuals provide a comparative analysis of the controller's capacity to follow the desired trajectory and the different forces magnitude throughout the process. A similar analysis for the $Z_{d,low}$ parameters is available in Figures 5.11 and 5.12.

The actual trajectory recorded during the experiments, compared to the reference trajectories, are presented in Figures 5.9 and 5.11. These comparisons are crucial in evaluating the controller's effectiveness under varying conditions, demonstrating its accuracy and capacity to admit external forces in trajectory tracking.

Further observations are drawn from Figures 5.10 and 5.12, where the magnitude of command forces, external forces, and position errors are detailed. These elements highlight the controller's adaptive responses to external disturbances and its adjustments to maintain trajectory tracking. The data presented allows to understand the controller's behavior,such as the relationship between applied forces and trajectory deviation, and between the command force and the position error.

Similar results are observed across the different DOFs. Therefore, one DOF has been selected for presentation. Annex C presents the results for all DOFs.



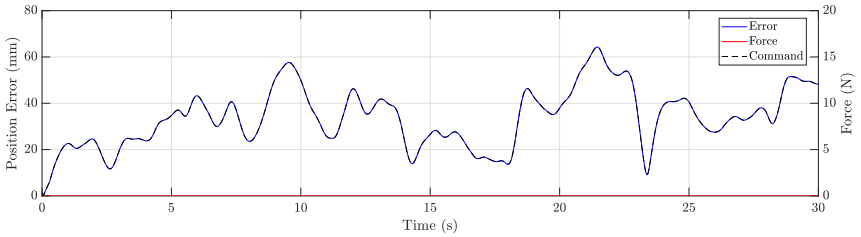
(a) Tracking of X Position

(b) Tracking of X Velocity

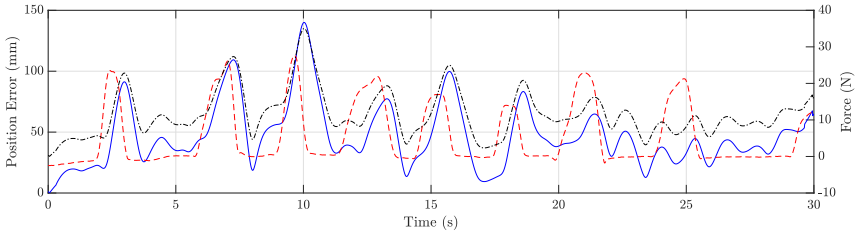
(c) Tracking of Pitch Position

(d) Tracking of Pitch Velocity

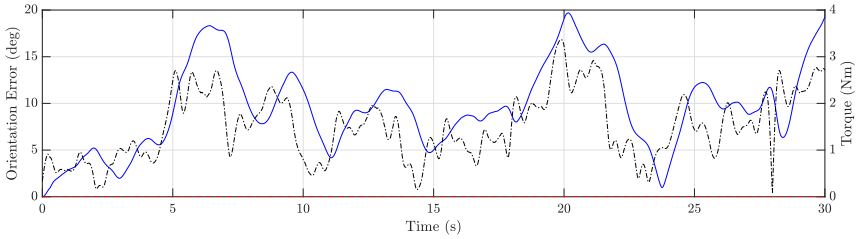
Figure 5.9 Tracking performance in the X direction and Pitch orientation using the Cartesian impedance controller. **(a), (b)** Position and velocity tracking for the X translation. **(c), (d)** Position and angular velocity tracking for the Pitch orientation. These results highlight the controller's ability to maintain the desired trajectory in these specific degrees of freedom



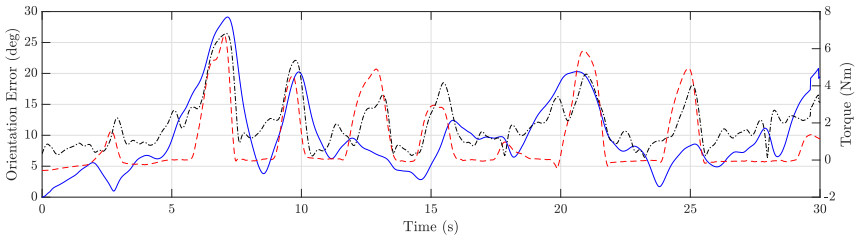
(a) Position Error and Forces-No perturbation



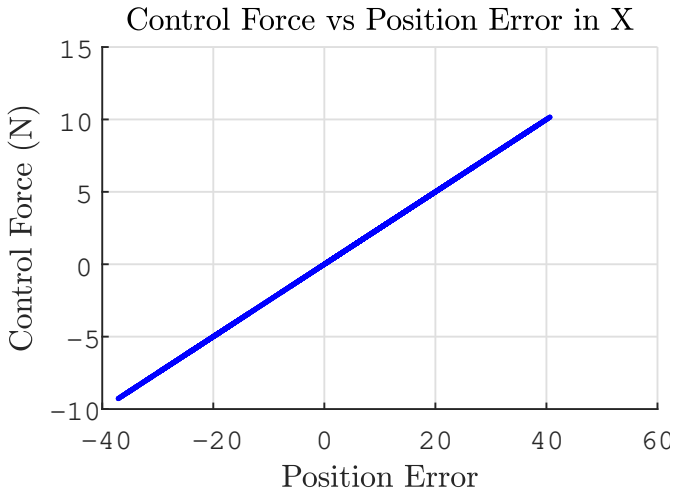
(b) Position Error and Forces and External Force-Perturbation



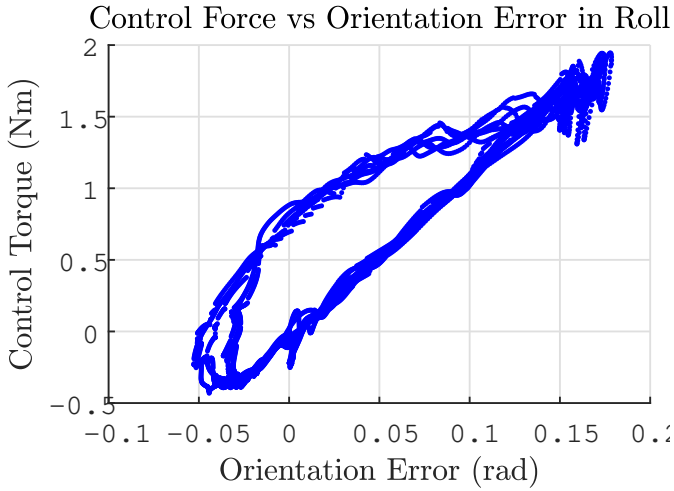
(c) Orientation Error and Forces-No perturbation



(d) Orientation Error and Forces-Perturbation

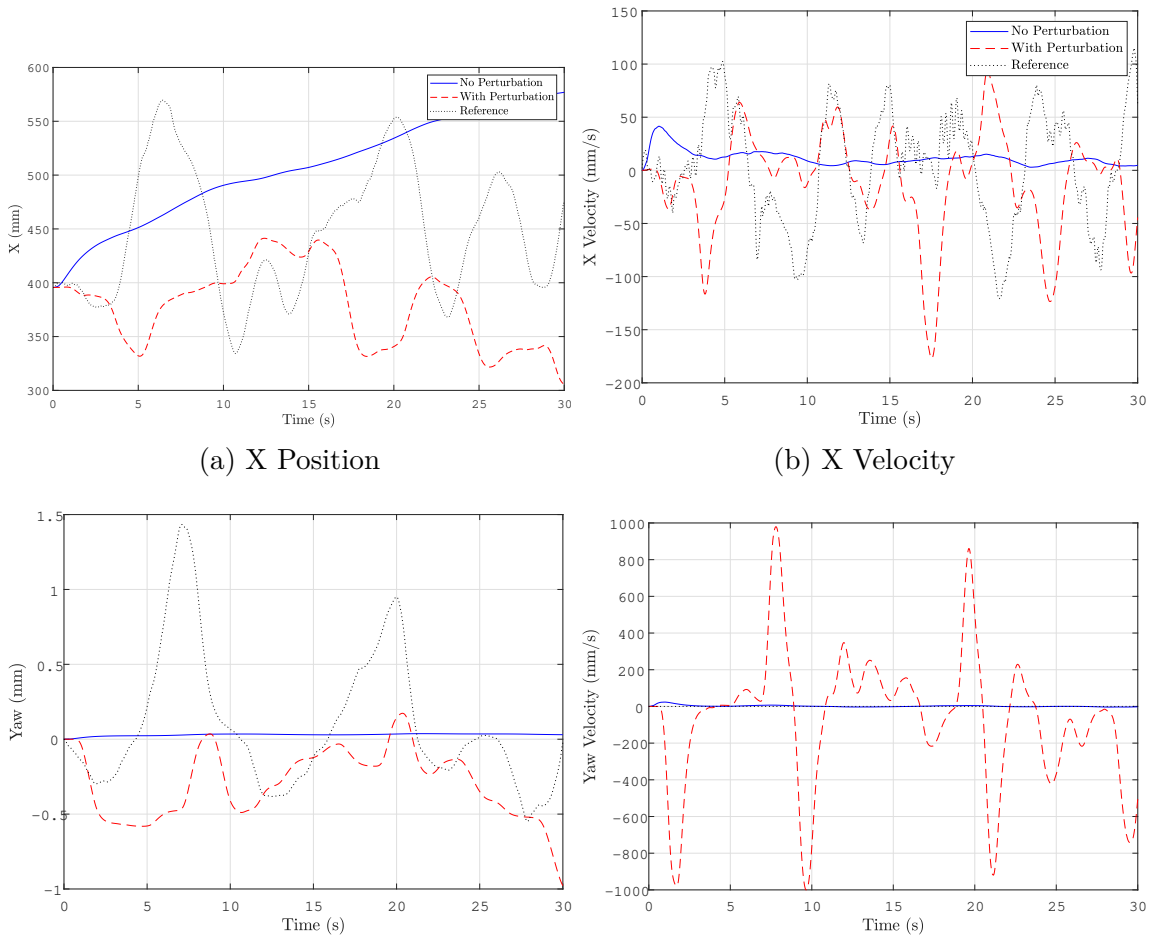


(e) F_X vs. e_X



(f) τ_{Roll} vs. e_{Roll}

Figure 5.10 Analysis of the control force and position error relationship during trajectory tracking with the Cartesian impedance controller under the $Z_{d,high}$ setting. **(a-d)**: Comparative insights under no perturbation and under external perturbations for both translational and orientational control. **(e-f)**: Control force with respect to the position error for the X direction and the Roll orientation



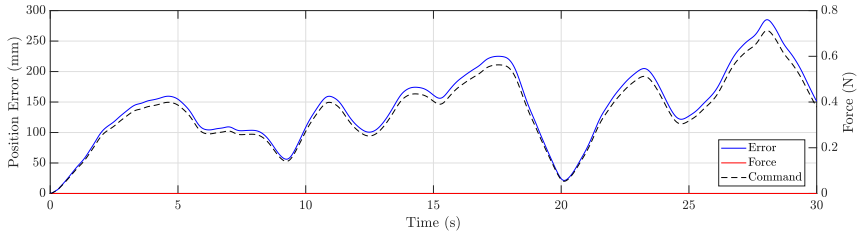
(a) X Position

(b) X Velocity

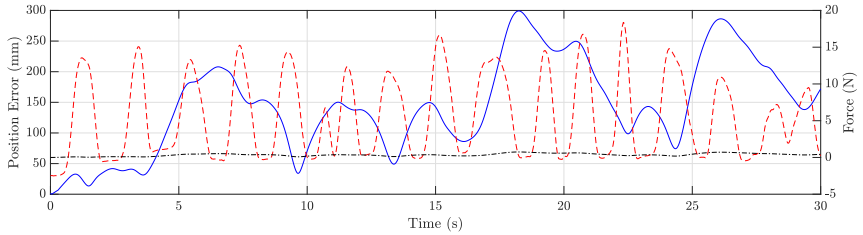
(c) Pitch Position

(d) Pitch Velocity

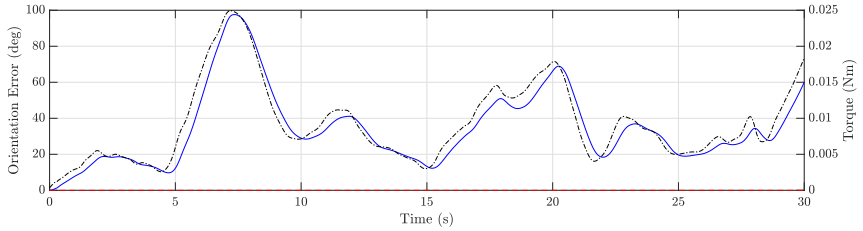
Figure 5.11 Tracking performance in the X direction and Pitch orientation using the Cartesian impedance controller with the Z_{low} parameters. **(a), (b)** Position and velocity tracking for the X translation. **(c), (d)** Position and angular velocity tracking for the Pitch orientation. These results highlight the controller's performance in maintaining the desired trajectory in these specific degrees of freedom



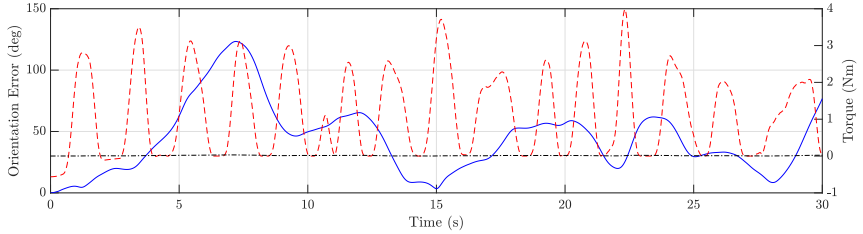
(a) Position Error and Forces-No perturbation



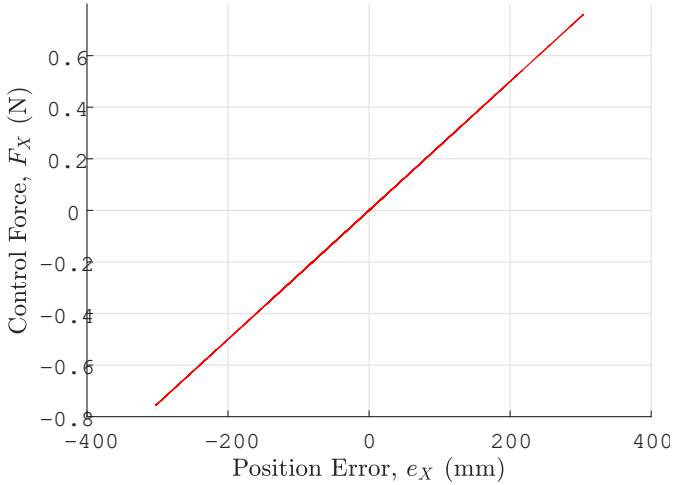
(b) Position Error and Forces and External Force-Perturbation



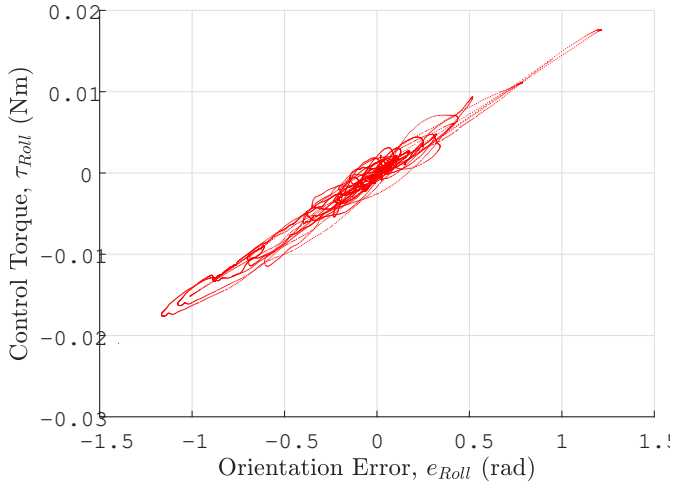
(c) Orientation Error and Forces-No perturbation



(d) Orientation Error and Forces-Perturbation



(e) F_X vs. e_X



(f) τ_{Roll} vs. e_{Roll}

Figure 5.12 Analysis of the command force and position error relationship during trajectory tracking with the Cartesian impedance controller under the $Z_{d,low}$ setting. **(a-d)**: Forces and error under no perturbation and under external perturbations for both translational and orientational control. **(e-f)**: Control force with respect to the position error for X translation and Roll orientation

5.1.5 Experiment 5: Adaptive Impedance Control Rehabilitation Exercises

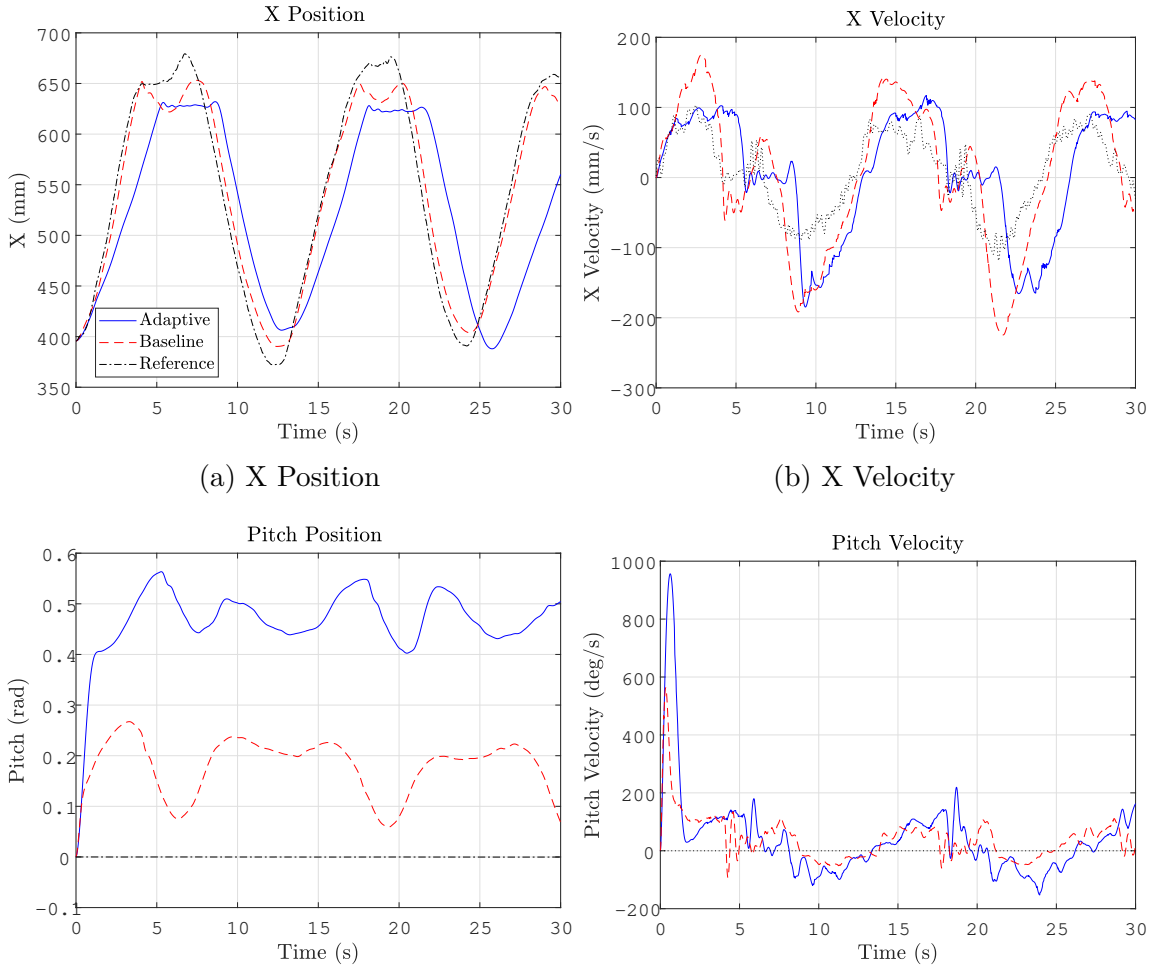
This section evaluates the adaptive impedance controller during various rehabilitation exercises, comprising three passive movements and one Activity of Daily Living (ADL). Each exercise was performed four times by a healthy adult male to ensure consistent arm positioning across the trials, adapting his limb's movements according to the specific exercise requirements.

Results for the *Passive Shoulder Horizontal Flexion* exercise are depicted in Figures 5.13 and 5.14. The exercise trajectory was generated using an admittance compliant vector of [1, 1, 0, 0, 0, 1] (X, Y, R_Z), facilitating motion in the horizontal plane. The parameters from the $Z_{d,high}$ setting in the previous experiment were adopted as the baseline for this exercise. Conversely, the *Can Off Shelf* exercise, which enabled all degrees of freedom for the admittance control, used the $Z_{d,low}$ setting as its baseline. The outcomes are illustrated in Figures 5.15 and 5.16.

Similar results are observed across the different DOFs. Therefore, one DOF has been selected for presentation. Annex D presents the results for all DOFs.

The data presented offers a comparative analysis between the adaptive Cartesian impedance controller's performance during different exercises, both passive and active. Utilizing the baseline values established in prior experiments allows for an assessment of the controller's efficiency with static parameters. In contrast, the adaptive data points represent the controller's responsiveness and adaptability according to the adaptive control law.

Moreover, Figures 5.18 and 5.19 show the variation in control parameters by the adaptive impedance controller, highlighting its dynamic adjustment capabilities during various rehabilitation exercises. Only the translational DOFs are presented. Similar results were observed for the orientational DOFs and are presented in Figures D.4 and D.5.



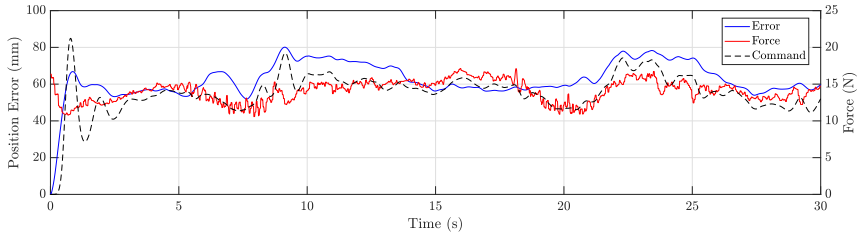
(a) X Position

(b) X Velocity

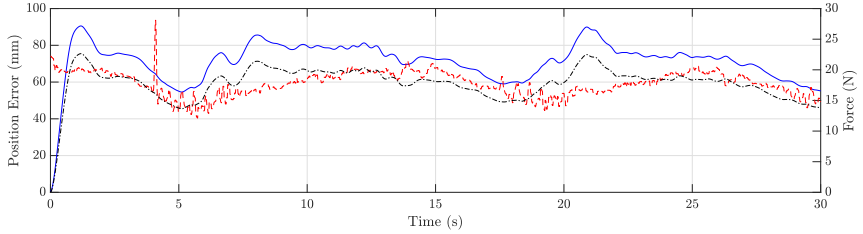
(c) Pitch Position

(d) Pitch Velocity

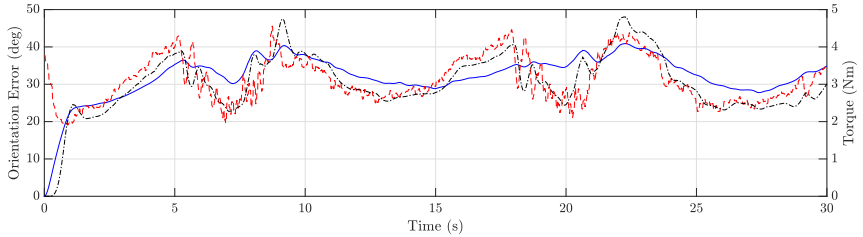
Figure 5.13 Tracking performance in the X direction and Pitch orientation using the Cartesian impedance controller with adaptive and passive parameters. **(a), (b)** Position and velocity tracking for the X translation. **(c), (d)** Position and angular velocity tracking for the Pitch orientation. These results illustrate the controller's ability to maintain the desired trajectory in passive rehabilitation exercises using adaptive impedance control



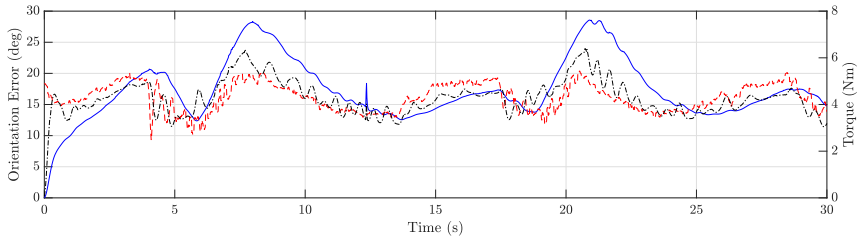
(a) Position Error and Forces-Adaptive



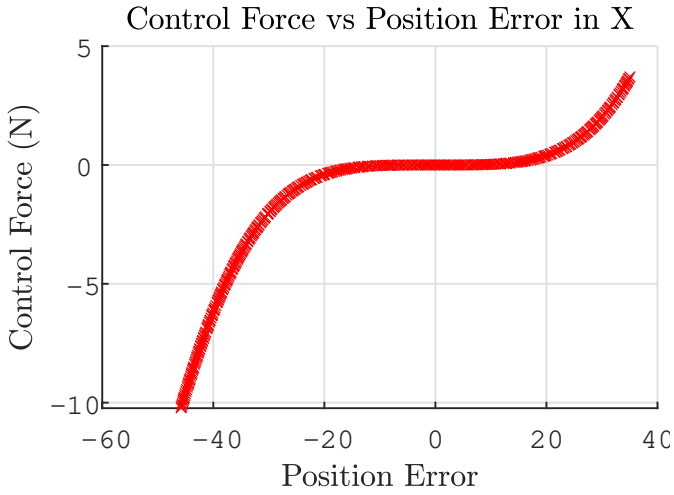
(b) Position Error and Forces and External Force-Passive



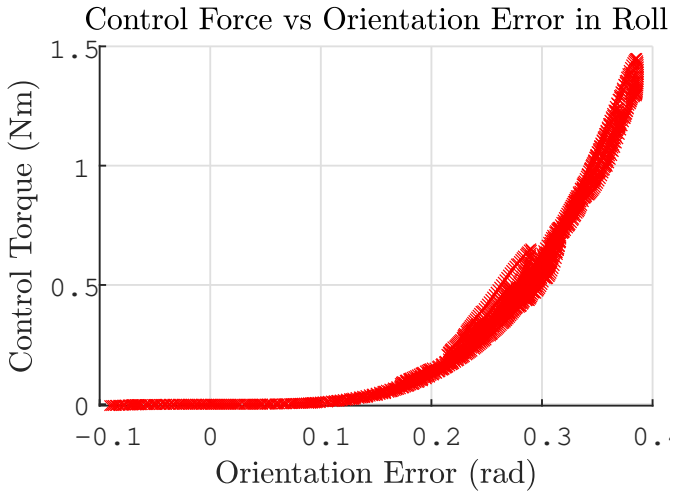
(c) Orientation Error and Forces-Adaptive



(d) Orientation Error and Forces-Passive

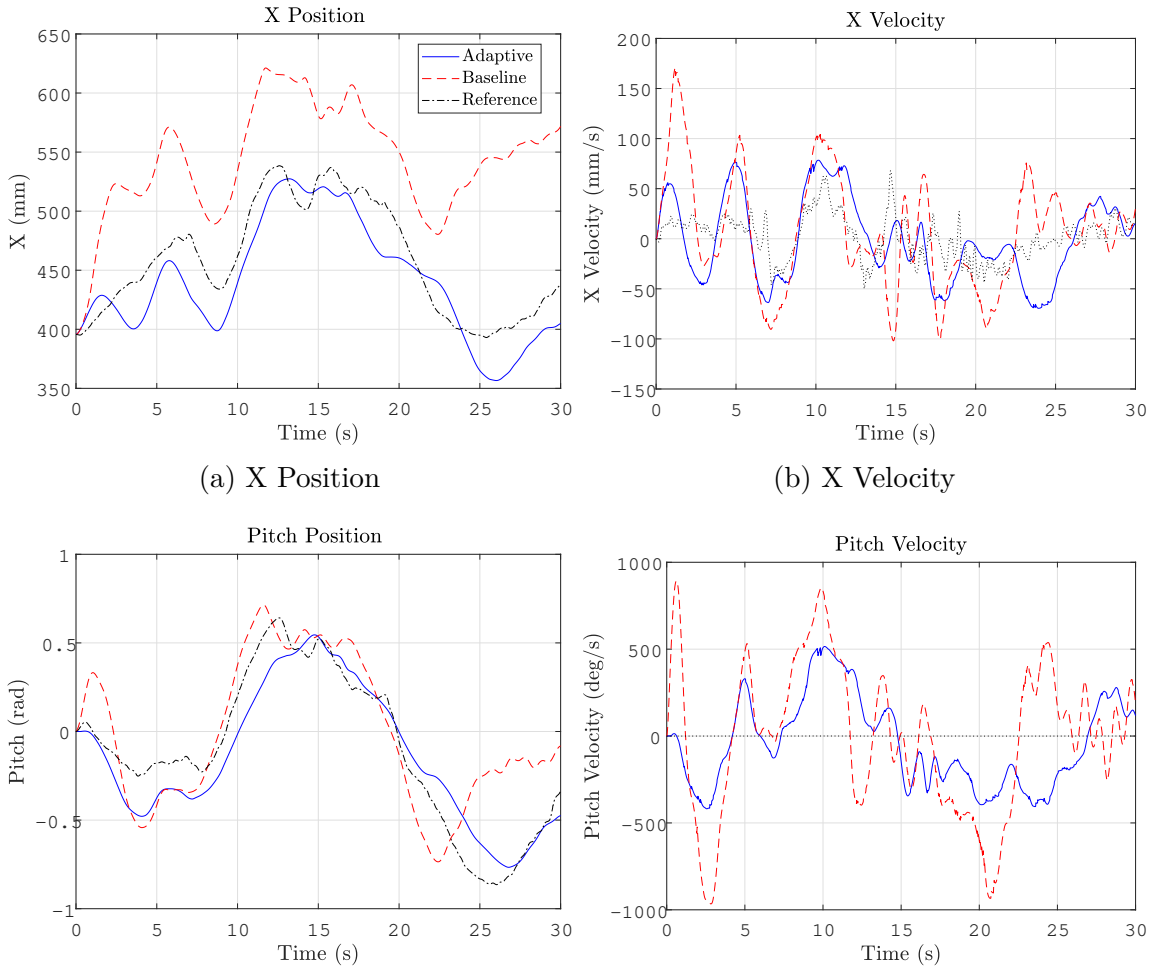


(e) F_X vs. e_X



(f) τ_{Roll} vs. e_{Roll}

Figure 5.14 Analysis of the control force and position error relationship during trajectory tracking with the adaptive Cartesian impedance controller for passive rehabilitation exercises. **(a-d)**: Comparative insights between the adaptive parameters and passive parameters control for both translational and orientational control. **(e-f)**: Control force with respect to the position error relationship for adaptive parameters in passive exercises



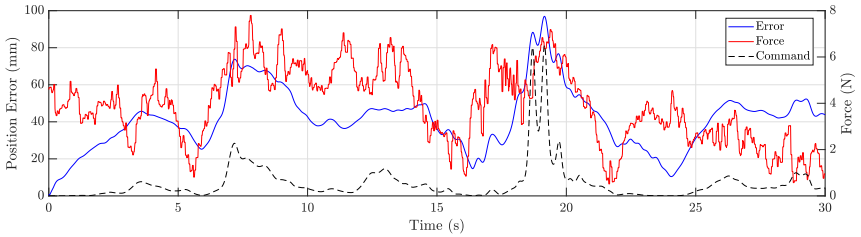
(a) X Position

(b) X Velocity

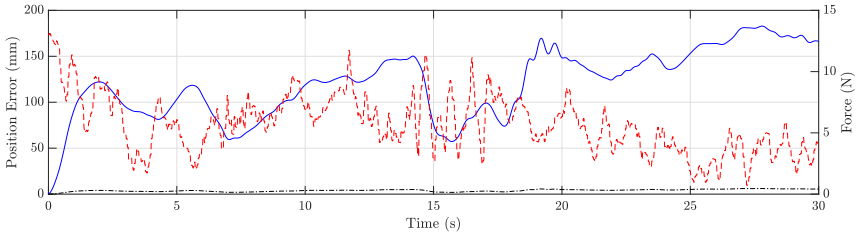
(c) Pitch Position

(d) Pitch Velocity

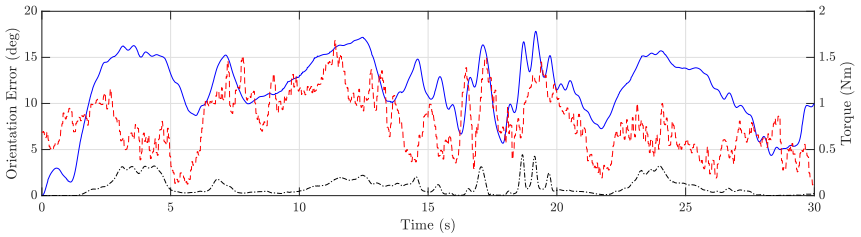
Figure 5.15 Tracking performance in the X direction and Pitch orientation using the Cartesian impedance controller with adaptive and active parameters for ADL exercises. **(a), (b)** Position and velocity tracking for the X translation. **(c), (d)** Position and angular velocity tracking for the Pitch orientation. These results illustrate the controller's ability to maintain the desired trajectory in Activities of Daily Living (ADL) exercises



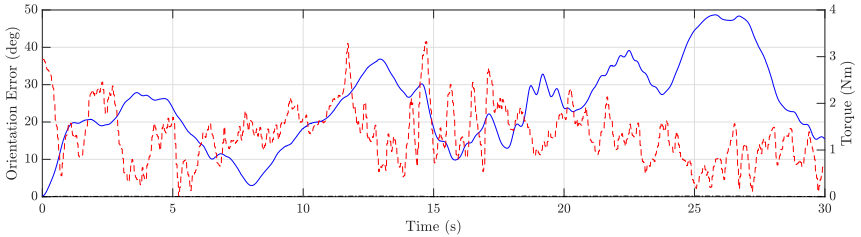
(a) Position Error and Forces-Adaptive



(b) Position Error and Forces and External Force-Active



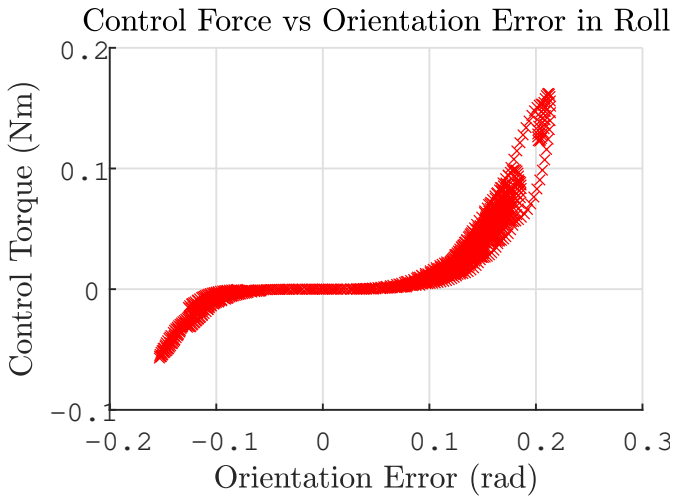
(c) Orientation Error and Forces-Adaptive



(d) Orientation Error and Forces-Active

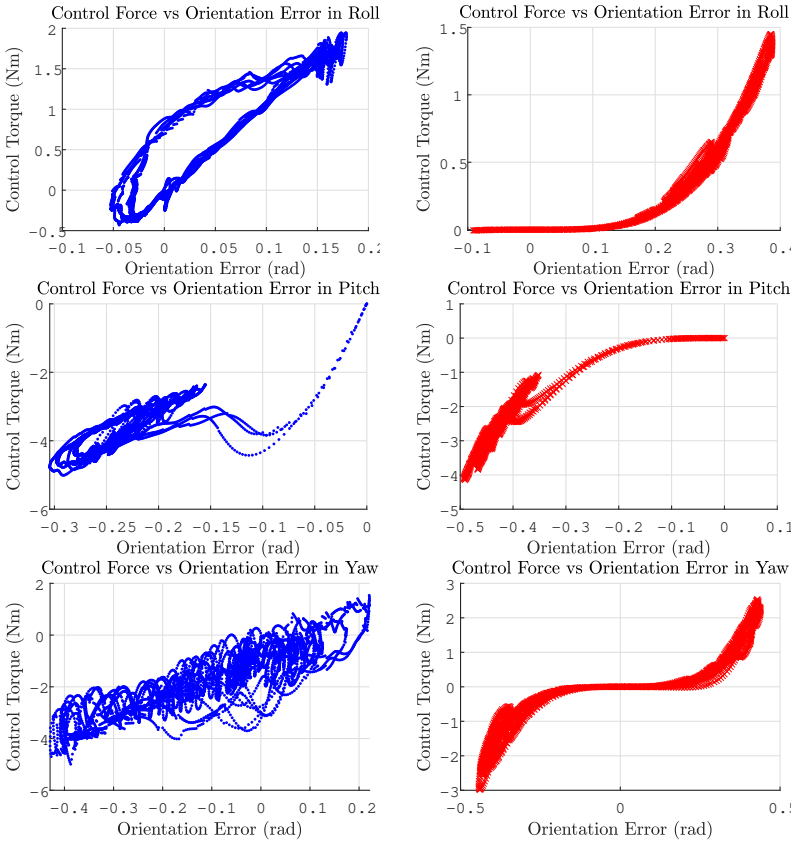


(e) F_X vs. e_X

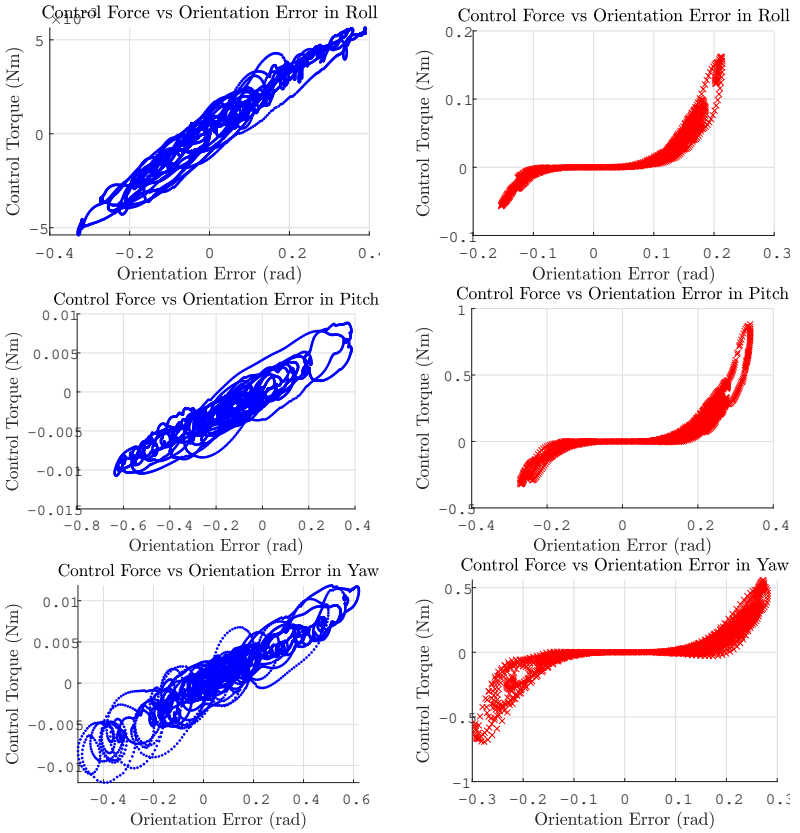


(f) τ_{Roll} vs. e_{Roll}

Figure 5.16 Analysis of the control force and position error relationship during trajectory tracking with the adaptive Cartesian impedance controller for ADL exercises. **(a-d)**: Comparison between the adaptive parameters and active parameters control for both translational and orientational control. **(e-f)**: Control force with respect to the position error relationship for adaptive parameters in ADL exercises



(a) Impedance control command torque with respect to the orientation error for ADL exercises using constant and adaptive parameters



(b) Impedance control command torque with respect to the orientation error for passive exercises using constant and adaptive parameters

Figure 5.17 Analysis of the impact relationship between the command torque and the orientation error for different experimental setup. This figure presents the command torque from the Cartesian impedance controller using constant parameters-passive and ADL- and variable parameters

5.2 Discussion

5.2.1 Cartesian Admittance Control

The objective of the admittance-based controller is to generate a reference trajectory for the patient's limb. This trajectory minimizes interaction forces between the patient's arm and the end-effector of the robotic manipulator.

Figure 5.1 displays a 3D trajectory generated using Cartesian admittance control. The trajectory aligns with the direction of the external force exerted on the end-effector by the therapist. Therefore, the therapist can guide the robotic device along a specific trajectory simply by interacting with it. This is achieved by setting the desired interaction force between the robotic manipulator and its environment to zero. Figure 5.2 illustrates the interaction force during trajectory generation using the admittance controller. The interaction force's magnitude is notably low, fluctuating between +/- 5 N and +/- 2 Nm for torque. As anticipated, when an interaction force is detected, the robotic device attempts to reduce it to zero. The forces detected are not completely null. This can be attributed to the total desired mechanical impedance (Z) of the admittance model, which determines the robot's sensitivity to external forces. As the total impedance increases, the required interaction force to displace the robot also increases proportionally. The values chosen for trajectory generation were determined experimentally, prioritizing both stability and the therapist's ability to maneuver the robotic device. The goal is for the robot to have a total impedance that is low enough to allow easy movement but high enough to filter out noise and provide feedback to the therapist. These parameters can be adapted to the therapist's preferences.

Figure 5.3 shows the relationship between the computed velocity command and the external force. A linear fit has been applied to the scatter plots in Figure 5.3b. This fit represents the admittance of the model upon which the controller is based. As detailed in the Methodology section, the admittance model correlates a force with a velocity output. The linear fit provides a measurement of the model's admittance. However, while the model is linear, the implementation of the controller introduces non-linearity, especially due to factors like unmodeled dynamics. Implementing this in Cartesian space further complicates the controller, particularly concerning orientation DOFs. To quantify the introduced non-linearities, the residual standard error (RSE) was calculated against the fitted linear curve for various compliance vectors.

The data in Figure 5.3b was generated using two distinct compliance vectors: [1,1,1,0,0,0] for translational DOFs and [0,0,0,1,1,1] for rotational DOFs. Conversely, Figure 5.3 presents results from trajectory generation using the fully compliant vector [1,1,1,1,1,1]. A comparison

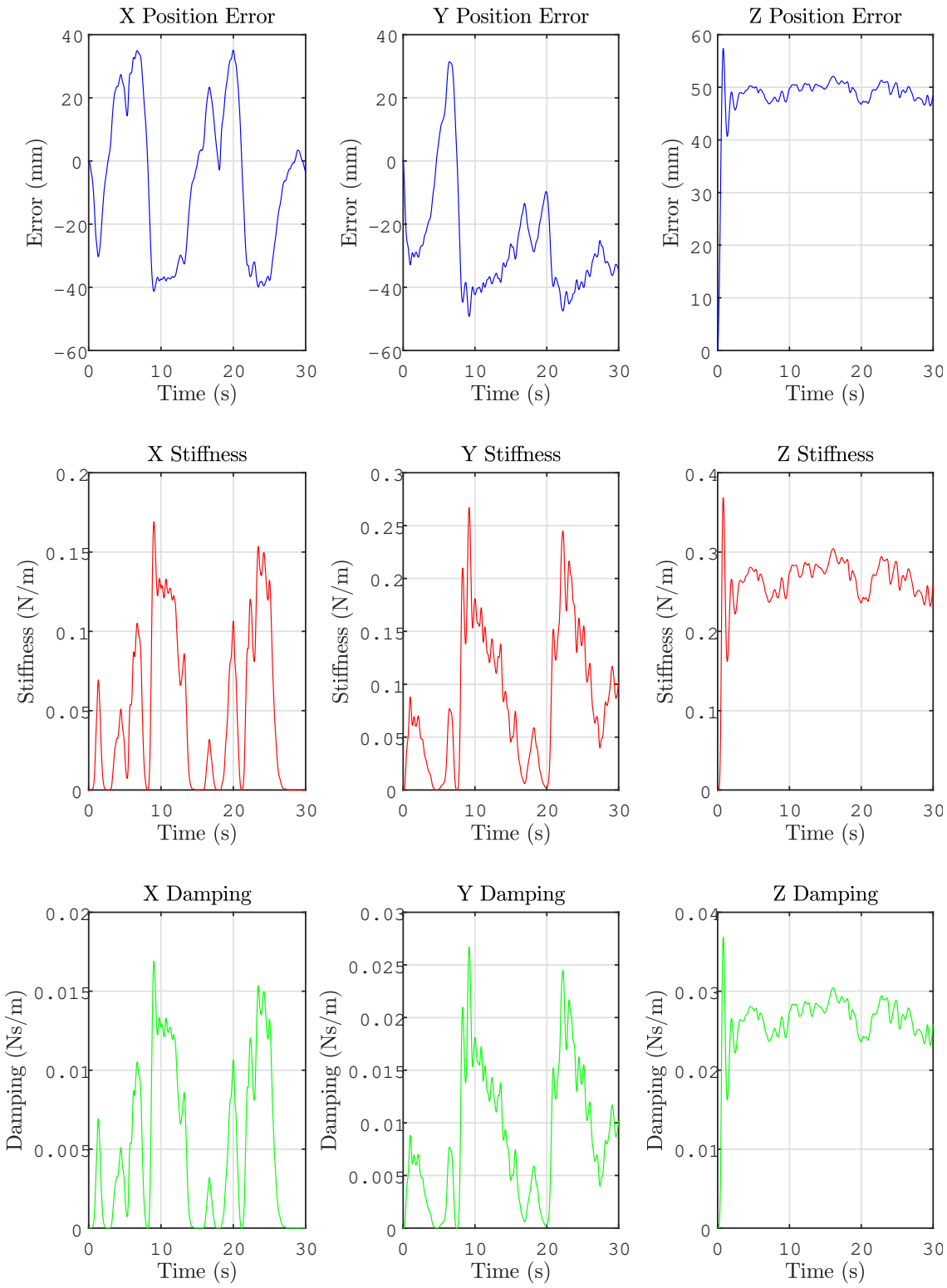


Figure 5.18 Variable Impedance Control Parameters : this figure shows the variation of the impedance parameters throughout the shoulder passive exercise

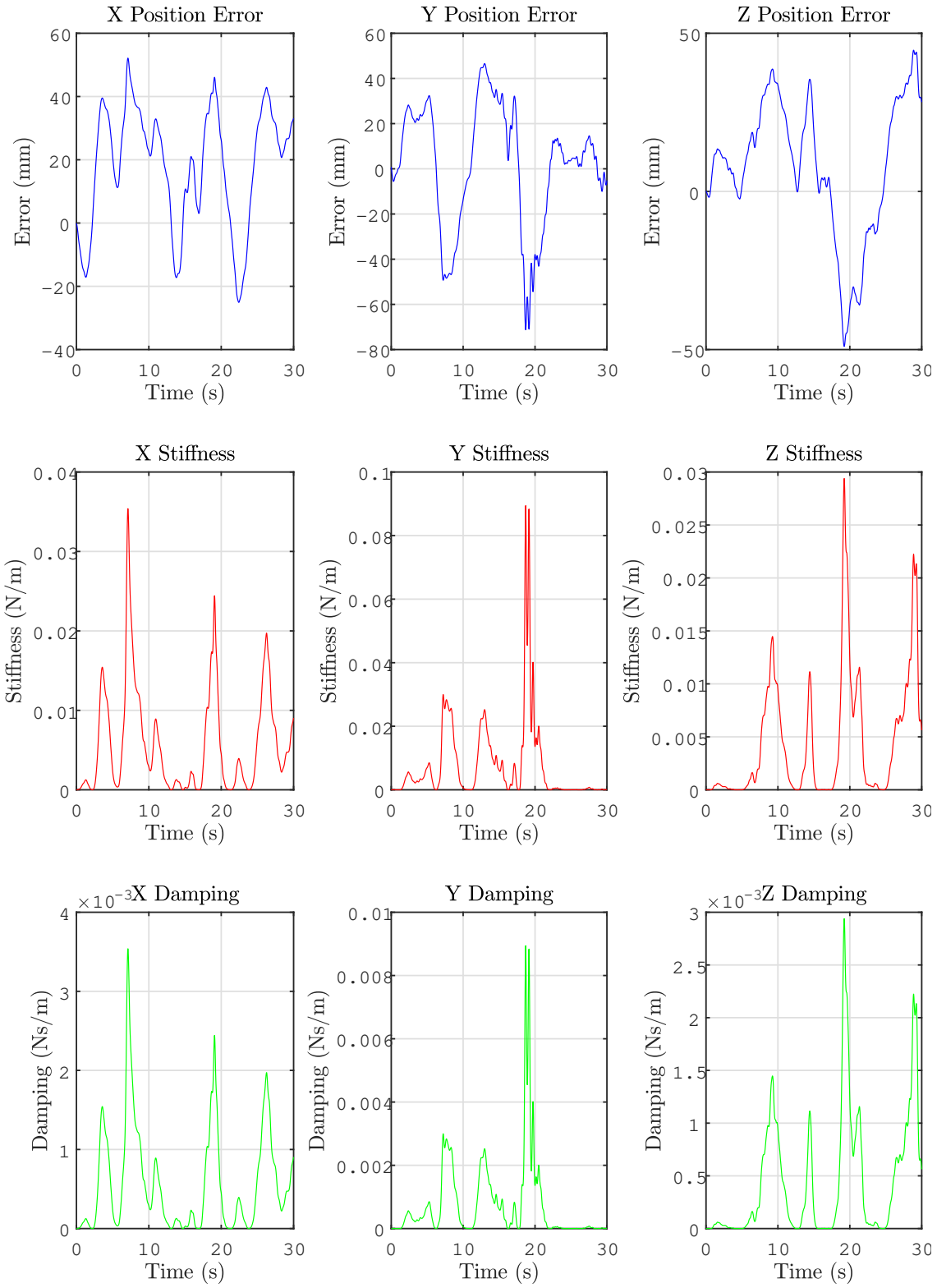


Figure 5.19 Variable Impedance Control Parameters : this figure shows the variation of the impedance parameters throughout the *Can Off Shelf* exercise

reveals that including more DOFs increases the system's complexity and non-linearity. This translates into an higher value of RSE. For the X direction, the RSE nearly doubles upon introducing rotational DOFs. A similar pattern is apparent for rotational DOFs, underscoring the influence of compliance settings on the controller performance.

A method to estimate total impedance in this system involves forming a complex number where the real part represents the damping impedance and the imaginary part represents the reactive (mass) impedance: $Z_d = B_d + j\omega M_d = 0.07 + j(0.01 \cdot \omega)$ (since $K_d = 0$). This is a frequency domain representation. For a step response at $t = 0$, a rough approximation of the mechanical impedance could be made using the damping coefficient alone, as $Z(0) = B = 0.07$. This simplification provides a rough estimate of the impedance at the exact moment the step input is applied, based solely on the damping term. Thus, the total admittance as a reference value at $t = 0$ is given by $\frac{1}{Z}$, which simplifies to $\frac{1}{B} \approx 14.29\,\frac{m}{Ns}$. The admittance value derived from the linear fit, around $16\,\frac{m}{Ns}$, depending on the compliance vector, may align closely with the system's estimated total admittance. Following the same approach for the rotational DOFs, with a $B_d = 7\frac{Nm \cdot s}{rad}$, the approximated admittance of the system is $\frac{1}{Z} = 0.14\frac{rad}{Nm \cdot s}$, which is consistent with the approximated value.

From a rehabilitation standpoint, the admittance controller provides a flexible approach for therapists to design patient-specific reference trajectories. In fact, the exact path and the range of motion (ROM) of the patient can be easily adapted to their need and capacities. Moreover, the compliance vector's integration allows therapists to determine the necessary DOFs for a given exercise.

To enhance the controller's efficacy, two singularity avoidance mechanisms were implemented, as described in the Methodology section. The outcomes are showcased and deliberated in Figures 5.4 and 5.5. The virtual force offers real-time protection against singularity configurations, which, if encountered, could hurt the patient's arm with sudden movements. It's vital to calibrate parameters—the threshold value w_{th}, the critical value w_{cr}, and the gain k—to suit the application. The dead zone, critical value, and force magnitude are all governed by these parameters. High magnitude virtual forces must be avoided, especially during trajectory tracking, to prevent injuries. Moreover, fine-tuning the critical value and threshold can yield a more precise singularity avoidance mechanism. The secondary mechanism, null-space control, designates a secondary task to avoid singularity. The velocities of this secondary task are kept minimal to prevent sudden motions, and not all joints are involved, indicating some are essential for the primary task.

In therapeutic settings, this increasing virtual force acts as a safeguard. It only attains significant values when the robot is deliberately forced into specific configurations, directing

the robot away from singularities, thus limiting excessive joint velocities and guaranteeing user safety and therapeutic efficacy.

5.2.2 Impedance Control: The Effect of Parameters

The impact of controller parameters, specifically desired stiffness K_d and damping B_d, was studied through a step response analysis. An examination of Figure 5.6 shows that the 16 experimental setups from 4.6 can be grouped into four main categories based on stiffness values. Within each group, the setups are sorted by increasing damping values.

A clear trend is observed: higher stiffness results in increased overshoot. When considering individual groups, overshoot reduces with increasing damping. For response time, as stiffness increases, it decreases; while with increased damping, response time grows. This indicates that as the system's damping rises and stiffness falls, the system slows down. Moreover, position error largely depends on stiffness. By the fourth group, a noticeable position error is evident, likely because the system becomes less sensitive to position errors with lower stiffness. This reduced sensitivity diminishes the focus on adjusting commands. These trends are further supported by a histogram (shown in Figure 5.7) that organizes various metrics by increasing stiffness. These patterns are consistent across all tests and throughout the robot's joints. The behavior of the robotic manipulator to external perturbations are presented on 5.8. As the impedance value increases, the position error due to the external perturbations are smaller and the position error is rapidly corrected. This means that the manipulator is *stiffer*, meaning that it will not allow high deviation from the reference trajectory.

5.2.3 Cartesian Impedance Control: Trajectory Tracking

The impedance controller, as outlined in Section 5.2.2, is engineered to guide the user's hand movements along predefined trajectories, making real-time adjustments in response to deviations. This controller operates under two distinct parameter sets: one tailored for passive exercises, requiring considerable support, and another for active exercises, catering to users with improved motor skills.

In the high impedance mode $Z_{d,high}$, illustrated in Figure 5.9, the robot diligently follows the reference trajectory, swiftly correcting any disturbances. An observable offset in the Roll direction, attributed to discontinuities stemming from the Roll Pitch Yaw representation, was identified and rectified during the system's implementation.

A thorough evaluation of the impedance controller's performance is presented in Figure 5.10. Sub-figures **a-d** showcase the control force's adaptive nature to changes in position error and

external perturbations, highlighting its rapid command force adjustments to counteract discrepancies. Sub-figures **e-f** elucidate a strong correlation between control force and position error for translational motions. This direct relationship, however, becomes less evident in rotational movements. The complexities of rotational dynamics, influenced by factors like non-commutativity of rotations and the challenges in accurately representing 3D orientation, bring about nuances not typically encountered in translational behaviors. Hence, the controller's reactions to rotational errors slightly deviate from the consistent responses seen in translational scenarios.

In the low impedance setting $Z_{d,low}$, as shown in Figure 5.11, the controller struggles to maintain adherence to the reference trajectory. This observation is further accentuated in Figure 5.12, where the control force remains relatively subdued (around 0.5 N), even when confronted with substantial position errors. Moreover, when exposed to external disturbances, the position error amplifies, signifying heightened admittance.

A notable insight from analyzing the controller's behavior across different impedance values is the seemingly more linear relationship between orientation error and control torque at lower impedance value. This trend can be associated with the system's lower stiffness and inertia. A compliant system, due to lower impedance, is inherently less responsive to disturbances and errors. This results in a more predictable response to orientation errors, mitigating the rotational complexities usually present in higher impedance scenarios. Essentially, the challenges of rotational dynamics are somewhat subdued in a more accommodating control setup. Therefore, at reduced impedance values, the controller demonstrates a clearer linear relationship between orientation errors and the resulting control torque (Figure 5.14 and 5.16).

These insights highlight an intrinsic trade-off within the impedance-based controller: balancing trajectory tracking precision with the robot's sensitivity to external forces. Enhancing the system's overall impedance improves trajectory tracking accuracy but reduces its compliance. In contrast, reducing impedance fosters greater robot compliance, albeit with potential susceptibility to disturbances. Striking an optimal balance between these factors is essential in therapeutic contexts, ensuring the robot's versatility across various rehabilitation exercises, from passive support to activities of daily living (ADL).

5.2.4 Adaptive Cartesian Impedance Control

An adaptive law was introduced to adjust the control parameters dynamically based on the patient's performance, providing real-time corrections according to the position error.

Figure 5.13 compares the performance of the adaptive law during a passive shoulder exercise with a baseline impedance controller (non-adaptive) that uses predetermined passive parameters. In both position and velocity tracking, the adaptive controller's performance closely matches that of the baseline. Notable differences appear in the Z translation and the *Pitch* orientation, DOFs that were inactive during trajectory generation. Analysis of Figure 5.14.(**a-d**) indicates similar patterns between the baseline and adaptive controllers. However, in the adaptive controller, there is a clear offset between position error and command force, unlike the baseline where they align closely.

Figures 5.15 and 5.16 further evaluates the adaptive law's effectiveness during an activity of daily living (ADL). Compared again to the non-adaptive baseline, the two controllers display similar performance in translation and orientation as shown on Figure 5.15. The primary difference emerges in Figure 5.16**(a-d)**. While the baseline controller's command force is nearly zero, the adaptive controller applies guiding forces on the user's arm upon significant deviations from the trajectory.

Moreover, Figures 5.19 and 5.18 illustrates the relationship between the desired stiffness K_d and damping B_d parameters and the position errors for both translation and orientation DOFs. These parameters adjust dynamically according to the adaptive law. It should be highlighted that, employing the identical adaptive law, the parameter magnitudes are substantially lower for active exercises compared to passive ones.

The figures 5.14**(e-f)** and 5.16**(e-f)** represents the relationship between the command force and the position error for variable impedance control. In the context of translational degrees of freedom (DOFs), this relationship was linear before the integration of the adaptive law, as evidenced in Figure 5.12**(e-f)** and 5.10**(e-f)**. However, with the introduction of the adaptive law, which employs the tanh function to adjust parameters based on the position error, the relationship transitions to a cubic curve. A similar pattern is observed for orientation DOFs, with both passive and ADL exercises demonstrating that the relationship between command torque and orientation error adheres to the hyperbolic tangent form. It's important to highlight that the data related to rotational DOFs is less consistent compared to that of the translational DOFs.

In conclusion, the introduction of the adaptive law to the impedance controller allows for real-time adjustments based on user performance. As users follow the desired trajectories accurately, the command forces exerted remain minimal. However, if users deviate significantly from the path, the controller applies stronger forces to guide them back. These findings support the use of the adaptive law for rehabilitation applications.

5.2.5 Adaptive Cartesian Impedance Control: Advantages over Non-Adaptive Controller

The benefits and behavior of the adaptive Cartesian impedance control compared to the non-adaptive variant can be discerned in Figure 5.17, which gathers the results of the control torque with respect to the orientation error for different control parameters.

Interestingly, the adaptive law's influence on orientation DOFs diverged from initial expectations. Instead of creating variances, the adaptive law made the correlation between command torque and orientation error more consistent, shaping it into a cubic curve as seen on Figure 5.17. A possible rationale behind this is that the gradual adjustment of parameters in relation to the orientation error allows the controller to operate with enhanced stability, sidestepping abrupt shifts in command forces. This consistent behavior mirrors the observations made in different impedance settings, namely Z_{d_low} and Z_{d_high}. In these settings, the less aggressive nature of the low impedance controller resulted in a more consistent relationship.

Adaptive control introduces two primary advantages. Firstly, its inherent nature of real-time adjustments, guided by live feedback, eliminates the risk of excessive reactions that could jeopardize both user and equipment safety. Secondly, this adaptive mechanism guarantees a consistent performance, even when faced with unforeseen disturbances or shifts in the operating conditions.

To conclude, adaptive control ensures uniformity and safety across all DOFs, underscoring the merits of integrating adaptive strategies into impedance-based robotic control.

CHAPTER 6 CONCLUSION

6.1 Conclusion

Summary of Works

This study presents a comprehensive investigation into admittance and impedance-based controllers in the context of robotic-assisted rehabilitation. The primary objective was to develop controllers that facilitate safe and effective interaction between the robotic manipulator and a patient's limb. Results showed that the admittance-based controller effectively produced a reference trajectory for the patient's limb, with both orientation and translation control exhibiting satisfactory outcomes. The impedance controller was designed to aid hand movement along predefined trajectories, revealing a significant interplay between trajectory fidelity and robot compliance.

Furthermore, the adaptive Cartesian impedance control was introduced, enabling real-time adjustments based on the user's behavior, enhancing consistency across all degrees of freedom. The integration of a virtual reality (VR) environment aimed to enhance patient engagement during therapy sessions. An essential aspect of this integration was the implementation of a communication protocol between the VR system and the robotic manipulator's API.

Limitations

Several limitations were identified in the study:

- The dynamics of the arm were not incorporated into the control loop, potentially influencing the interaction forces as recorded by the force sensor.
- A simplified and approximated dynamic model was employed for the robotic manipulator, which could introduce systemic errors.
- The arm support mechanism was basic, not providing full support to the entire arm. The design also lacked features for isolating specific joints, such as the capability to block the elbow joint to focus on the shoulder.
- Utilizing *Forward Kinematics* in Cartesian control behaves like a black-box and might influence the efficacy of the control strategies, especially for the rotational DOFs.

Future Research

Areas for future research include the development of more complex adaptive laws leveraging additional sensory information and alternative learning models. There is potential in employing machine learning algorithms to better understand the robotic device's dynamic model. Moreover, the fine tuning of the different parameters, such as the virtual force field for singularity avoidance could potentially increase the performance of the robotic manipulator. Collaborative efforts with healthcare professionals will be crucial to guide the design and optimization of the robotic platform further. Enriching the VR environment and considering the addition of haptic feedback also presents an avenue worth exploring to enhance the therapeutic experience.

In conclusion, this research contributes significantly to the growing body of knowledge on robotic-assisted rehabilitation, laying the groundwork for further advancements in this critical domain.

REFERENCES

[1] M. Babaiasl, S. H. Mahdioun, P. Jaryani, and M. Yazdani, "A review of technological and clinical aspects of robot-aided rehabilitation of upper-extremity after stroke," *Disability and Rehabilitation: Assistive Technology*, vol. 11, no. 4, pp. 263–280, 2016.

[2] D. Wang. (2023) Kinematic and dynamic parameters of ufactory xarm series.

[3] C. for Disease Control and Prevention, "About stroke," https://www.cdc.gov/stroke/about.htm, 2023, accessed: 2023-07-10.

[4] E. Blume, "Of exoskeletons and service robots – the future of rehabilitation," 2023. [Online]. Available: https://www.medica-tradefair.com/en/physio-tech/Of_exoskeletons_and_service_robots_%E2%80%93_the_future_of_rehabilitation

[5] UFATORY, *xArm User Manual V2.0.0*, 2023, accessed: Insert the date you accessed the manual. [Online]. Available: https://www.ufactory.cc/wp-content/uploads/2023/05/xArm-User-Manual-V2.0.0.pdf

[6] B. Hochreiter, S. Wyss, and C. Gerber, "Extension of the shoulder is essential for functional internal rotation after reverse total shoulder arthroplasty," *Journal of Shoulder and Elbow Surgery*, vol. 31, no. 6, pp. 1166–1174, 2022.

[7] C. Nguiadem, M. Raison, and S. Achiche, "Motion planning of upper-limb exoskeleton robots: a review," *Applied Sciences*, vol. 10, no. 21, p. 7626, 2020.

[8] A. Tapus, C. Tapus, and M. J. Matarić, "User—robot personality matching and assistive robot behavior adaptation for post-stroke rehabilitation therapy," *Intelligent Service Robotics*, Feb 2008.

[9] D. Webster and O. Celik, "Systematic review of kinect applications in elderly care and stroke rehabilitation," *Journal of Neuroengineering and Rehabilitation*, Jul 2014.

[10] D. P. Losey, C. G. McDonald, E. Battaglia, and M. K. O'Malley, "A review of intent detection, arbitration, and communication aspects of shared control for physical human–robot interaction," *Applied Mechanics Reviews*, Jan 2018.

[11] F. Molteni, G. Gasperini, G. Cannaviello, and E. Guanziroli, "Exoskeleton and end-effector robots for upper and lower limbs rehabilitation: Narrative review," *Pmr*, Sep 2018.

[12] Y. Bouteraa, I. B. Abdallah, A. ElMogy, A. Ibrahim, U. Tariq, and T. Ahmad, "A fuzzy logic architecture for rehabilitation robotic systems," *International Journal of Computers Communications & Control*, vol. 15, no. 4, 2020.

[13] P. Shah Nazar and P. P. Pott, "Ankle rehabilitation robotic systems for domestic use–a systematic review," *Current Directions in Biomedical Engineering*, vol. 8, no. 2, pp. 65–68, 2022.

[14] T. Noritsugu and T. Tanaka, "Application of rubber artificial muscle manipulator as a rehabilitation robot," *IEEE-ASME Transactions on Mechatronics*, Dec 1997.

[15] P. Lum, D. Reinkensmeyer, R. Mahoney, W. Z. Rymer, and C. Burgar, "Robotic devices for movement therapy after stroke: current status and challenges to clinical acceptance," *Topics in stroke rehabilitation*, vol. 8, no. 4, pp. 40–53, 2002.

[16] D. Simonetti, L. Zollo, E. Papaleo, G. Carpino, and E. Guglielmelli, "Multimodal adaptive interfaces for 3d robot-mediated upper limb neuro-rehabilitation: An overview of bio-cooperative systems," *Robotics and Autonomous Systems*, vol. 85, pp. 62–72, 2016.

[17] R. Iandolo, F. Marini, M. Semprini, M. Laffranchi, M. Mugnosso, A. Cherif, L. De Michieli, M. Chiappalone, and J. Zenzeri, "Perspectives and challenges in robotic neurorehabilitation," *Applied Sciences*, vol. 9, no. 15, p. 3183, 2019.

[18] M. Zhang, T. C. Davies, and S. Xie, "Effectiveness of robot-assisted therapy on ankle rehabilitation–a systematic review," *Journal of neuroengineering and rehabilitation*, vol. 10, no. 1, pp. 1–16, 2013.

[19] L. Piggott, S. Wagner, and M. Ziat, "Haptic neurorehabilitation and virtual reality for upper limb paralysis: a review," *Critical Reviews™ in Biomedical Engineering*, vol. 44, no. 1-2, 2016.

[20] Z. Li, Z. Huang, W. He, and C.-Y. Su, "Adaptive impedance control for an upper limb robotic exoskeleton using biological signals," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 2, pp. 1664–1674, 2016.

[21] S. Hussain, S. Q. Xie, and P. K. Jamwal, "Adaptive impedance control of a robotic orthosis for gait rehabilitation," *IEEE transactions on cybernetics*, vol. 43, no. 3, pp. 1025–1034, 2013.

[22] S. Geyh, A. Cieza, J. Schouten, H. Dickson, P. Frommelt, Z. Omar, N. Kostanjsek, H. Ring, and G. Stucki, "Icf core sets for stroke," *Journal of rehabilitation medicine*, vol. 36, no. 0, pp. 135–141, 2004.

[23] P. Raghavan, "Upper limb motor impairment after stroke," *Physical Medicine and Rehabilitation Clinics*, vol. 26, no. 4, pp. 599–610, 2015.

[24] R. W. Livingstone and G. S. Paleg, "Use of overground supported-stepping devices for non-ambulant children, adolescents, and adults with cerebral palsy: A scoping review," *Disabilities*, vol. 3, no. 2, pp. 165–195, 2023.

[25] J. Fisher Rebecca, C. Gaynor, M. Kerr *et al.*, "A consensus on stroke," *Stroke*, vol. 42, no. 5, pp. 1392–1397, 2011.

[26] C. W. Tsao, A. W. Aday, Z. I. Almarzooq, A. Alonso, A. Z. Beaton, M. S. Bittencourt, A. K. Boehme, A. E. Buxton, A. P. Carson, Y. Commodore-Mensah *et al.*, "Heart disease and stroke statistics—2022 update: a report from the american heart association," *Circulation*, vol. 145, no. 8, pp. e153–e639, 2022.

[27] K. A. Almhdawi, V. G. Mathiowetz, M. White, and R. C. delMas, "Efficacy of occupational therapy task-oriented approach in upper extremity post-stroke rehabilitation," *Occupational therapy international*, vol. 23, no. 4, pp. 444–456, 2016.

[28] K. B. Lee, S. H. Lim, K. H. Kim, K. J. Kim, Y. R. Kim, W. N. Chang, J. W. Yeom, Y. D. Kim, and B. Y. Hwang, "Six-month functional recovery of stroke patients: a multi-time-point study," *International journal of rehabilitation research. Internationale Zeitschrift fur Rehabilitationsforschung. Revue internationale de recherches de readaptation*, vol. 38, no. 2, p. 173, 2015.

[29] J. R. Shiber, E. Fontane, and A. Adewale, "Stroke registry: hemorrhagic vs ischemic strokes," *The American journal of emergency medicine*, vol. 28, no. 3, pp. 331–333, 2010.

[30] Heart and S. F. of Canada, "Types of stroke," 2023, accessed: 2023-10-12. [Online]. Available: https://www.heartandstroke.ca/stroke/what-is-stroke/types-of-stroke

[31] B. H. Dobkin and A. Dorsch, "New evidence for therapies in stroke rehabilitation," *Current Atherosclerosis Reports*, vol. 15, no. 6, p. 331, Jun 2013.

[32] J. W. Krakauer, S. T. Carmichael, D. Corbett, and G. F. Wittenberg, "Getting neurorehabilitation right: what can be learned from animal models?" *Neurorehabilitation and neural repair*, vol. 26, no. 8, pp. 923–931, 2012.

[33] S. L. Wolf, C. J. Winstein, J. P. Miller, E. Taub, G. Uswatte, D. Morris, C. Giuliani, K. E. Light, D. Nichols-Larsen, f. t. EXCITE Investigators *et al.*, "Effect of constraint-induced movement therapy on upper extremity function 3 to 9 months after stroke: the excite randomized clinical trial," *Jama*, vol. 296, no. 17, pp. 2095–2104, 2006.

[34] F. Ferrarello, M. Baccini, L. A. Rinaldi, M. C. Cavallini, E. Mossello, G. Masotti, N. Marchionni, and M. Di Bari, "Efficacy of physiotherapy interventions late after stroke: a meta-analysis," *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 82, no. 2, pp. 136–143, 2011.

[35] X. Chen, D. Hu, R. Zhang, Z. Pan, Y. Chen, L. Xie, J. Luo, and Y. Zhu, "Interpretable evaluation for the brunnstrom recovery stage of the lower limb based on wearable sensors," *Frontiers in Neuroinformatics*, vol. 16, 2022. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fninf.2022.1006494

[36] L. Provinciali, L. Binaretti, A. Bortone, S. Cappa, M. Ceravolo, C. Cerri, P. Consolmagno, R. Delsanto, C. Flosi, R. Frediani *et al.*, "Trattamento riabilitativo e continuità dell'assistenza," in *SPREAD-Ictus cerebrale: linee guida italiane di prevenzione e trattamento*, 2007, pp. 427–482.

[37] J. Johnson, V. Pearson, and L. McDivitt, "Stroke rehabilitation: Assessing stroke survivors' long-term learning needs," *Rehabilitation Nursing*, vol. 22, no. 5, pp. 243–248, 1997.

[38] D. X. Cifu and D. G. Stewart, "Factors affecting functional outcome after stroke: a critical review of rehabilitation interventions," *Archives of physical medicine and rehabilitation*, vol. 80, no. 5, pp. S35–S39, 1999.

[39] H. I. Krebs, N. Hogan, M. L. Aisen, and B. T. Volpe, "Robot-aided neurorehabilitation," *IEEE transactions on rehabilitation engineering*, vol. 6, no. 1, pp. 75–87, 1998.

[40] R. Loureiro, F. Amirabdollahian, M. Topping, B. Driessen, and W. Harwin, "Upper limb robot mediated stroke therapy—gentle/s approach," *Autonomous Robots*, vol. 15, pp. 35–51, 2003.

[41] T. Proietti, V. Crocher, A. Roby-Brami, and N. Jarrasse, "Upper-limb robotic exoskeletons for neurorehabilitation: a review on control strategies," *IEEE reviews in biomedical engineering*, vol. 9, pp. 4–14, 2016.

[42] B. D. Schmit, J. P. Dewald, and W. Z. Rymer, "Stretch reflex adaptation in elbow flexors during repeated passive movements in unilateral brain-injured patients," *Archives of physical medicine and rehabilitation*, vol. 81, no. 3, pp. 269–278, 2000.

[43] N. Vitiello, T. Lenzi, S. Roccella, S. M. M. De Rossi, E. Cattin, F. Giovacchini, F. Vecchi, and M. C. Carrozza, "Neuroexos: A powered elbow exoskeleton for physical rehabilitation," *IEEE transactions on robotics*, vol. 29, no. 1, pp. 220–235, 2012.

[44] J. Ueda, D. Ming, V. Krishnamoorthy, M. Shinohara, and T. Ogasawara, "Individual muscle control using an exoskeleton robot for muscle function testing," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, no. 4, pp. 339–350, 2010.

[45] A. Subasi, "Electromyogram-controlled assistive devices," in *Bioelectronics and Medical Devices*. Elsevier, 2019, pp. 285–311.

[46] Y. Ren, S. H. Kang, H.-S. Park, Y.-N. Wu, and L.-Q. Zhang, "Developing a multi-joint upper limb exoskeleton robot for diagnosis, therapy, and outcome evaluation in neurorehabilitation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 21, no. 3, pp. 490–499, 2012.

[47] X. L. Hu, K.-y. Tong, R. Song, X. J. Zheng, and W. W. Leung, "A comparison between electromyography-driven robot and passive motion device on wrist rehabilitation for chronic stroke," *Neurorehabilitation and neural repair*, vol. 23, no. 8, pp. 837–846, 2009.

[48] S. E. Fasoli, H. I. Krebs, J. Stein, W. R. Frontera, R. Hughes, and N. Hogan, "Robotic therapy for chronic motor impairments after stroke: Follow-up results," *Archives of physical medicine and rehabilitation*, vol. 85, no. 7, pp. 1106–1111, 2004.

[49] B. Sheng, Y. Zhang, W. Meng, C. Deng, and S. Xie, "Bilateral robots for upper-limb stroke rehabilitation: State of the art and future prospects," *Medical engineering & physics*, vol. 38, no. 7, pp. 587–606, 2016.

[50] I. Doumas, G. Everard, S. Dehem, and T. Lejeune, "Serious games for upper limb rehabilitation after stroke: a meta-analysis," *Journal of neuroengineering and rehabilitation*, vol. 18, pp. 1–16, 2021.

[51] L. B. G. S. D. J. S. G. Laver, KE and M. Crotty, "Virtual reality for stroke rehabilitation," *Cochrane Database of Systematic Reviews*, no. 11, 2017. [Online]. Available: https://doi.org//10.1002/14651858.CD008349.pub4

[52] "The 3 types of virtual reality," https://heizenrader.com/the-3-types-of-virtual-reality/, July 2023, accessed: 2023-07-01.

[53] M. Maier, B. R. Ballester, A. Duff, E. D. Oller, and P. F. M. J. Verschure, "Effect of specific over nonspecific vr-based rehabilitation on poststroke motor recovery: A systematic meta-analysis," *Neurorehabilitation and Neural Repair*, vol. 33, no. 2, pp. 112–129, 2019, pMID: 30700224. [Online]. Available: https://doi.org/10.1177/1545968318820169

[54] "Gripable game helping stroke rehabilitation," https://www.ns-businesshub.com/technology/gripable-game-helping-stroke-rehabilitation/, 2023, accessed: 2023-07-10.

[55] "Reaplan combines robot-assisted upper limb rehabilitation with vr technology," https://www.fitness-gaming.com/news/health-and-rehab/reaplan-combines-robot-assisted-upper-limb-rehabilitation-with-vr-technology.html, 2023, accessed: 2023-07-10.

[56] "The science behind virtual reality in rehabilitation," https://blog.vive.com/us/virtual-reality-science-rehabilitation/, 2023, accessed: 2023-07-10.

[57] P. Kiper, A. Szczudlik, M. Agostini, J. Opara, R. Nowobilski, L. Ventura, P. Tonin, and A. Turolla, "Virtual reality for upper limb rehabilitation in subacute and chronic stroke: A randomized controlled trial," *Archives of Physical Medicine and Rehabilitation*, vol. 99, no. 5, pp. 834–842.e4, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0003999318300996

[58] W.-S. Kim, S. Cho, J. Ku, Y. Kim, K. Lee, H.-J. Hwang, and N.-J. Paik, "Clinical application of virtual reality for upper limb motor rehabilitation in stroke: Review of technologies and clinical evidence," *Journal of Clinical Medicine*, vol. 9, no. 10, 2020. [Online]. Available: https://www.mdpi.com/2077-0383/9/10/3369

[59] G. Kwakkel, R. C. Wagenaar, T. W. Koelman, G. J. Lankhorst, and J. C. Koetsier, "Effects of intensity of rehabilitation after stroke: a research synthesis," *Stroke*, vol. 28, no. 8, pp. 1550–1556, 1997.

[60] A. Gupta and M. K. O'Malley, "Design of a haptic arm exoskeleton for training and rehabilitation," *IEEE/ASME Transactions on mechatronics*, vol. 11, no. 3, pp. 280–289, 2006.

[61] B. T. Volpe, D. Lynch, A. Rykman-Berland, M. Ferraro, M. Galgano, N. Hogan, and H. I. Krebs, "Intensive sensorimotor arm training mediated by therapist or robot improves hemiparesis in patients with chronic stroke," *Neurorehabilitation and neural repair*, vol. 22, no. 3, pp. 305–310, 2008.

[62] H. I. Krebs and N. Hogan, "Robotic therapy: the tipping point," *American journal of physical medicine & rehabilitation/Association of Academic Physiatrists*, vol. 91, no. 11 0 3, p. S290, 2012.

[63] D. Lynch, M. Ferraro, J. Krol, C. M. Trudell, P. Christos, and B. T. Volpe, "Continuous passive motion improves shoulder joint integrity following stroke," *Clinical rehabilitation*, vol. 19, no. 6, pp. 594–599, 2005.

[64] T. Platz, "Evidenzbasierte armrehabilitation: Eine systematische literaturübersicht," *Der Nervenarzt*, vol. 74, pp. 841–849, 2003.

[65] A. Blank, M. K. O'Malley, G. E. Francisco, and J. L. Contreras-Vidal, "A pre-clinical framework for neural control of a therapeutic upper-limb exoskeleton," in *2013 6th International IEEE/EMBS Conference on Neural Engineering (NER)*. IEEE, 2013, pp. 1159–1162.

[66] X. Hu, K. Tong, X. Wei, W. Rong, E. Susanto, and S. Ho, "The effects of post-stroke upper-limb training with an electromyography (emg)-driven hand robot," *Journal of Electromyography and Kinesiology*, vol. 23, no. 5, pp. 1065–1074, 2013.

[67] M. Sarac, E. Koyas, A. Erdogan, M. Cetin, and V. Patoglu, "Brain computer interface based robotic rehabilitation with online modification of task speed," in *2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR)*. IEEE, 2013, pp. 1–7.

[68] R. Gassert and V. Dietz, "Rehabilitation robots for the treatment of sensorimotor deficits: a neurophysiological perspective," *Journal of neuroengineering and rehabilitation*, vol. 15, no. 1, pp. 1–15, 2018.

[69] A. AG, "Eduexo pro," 2023. [Online]. Available: https://www.auxivo.com/eduexo-pro

[70] A. Toth, G. Fazekas, G. Arz, M. Jurak, and M. Horvath, "Passive robotic movement therapy of the spastic hemiparetic arm with reharob: report of the first clinical test and the follow-up system improvement," in *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.* IEEE, 2005, pp. 127–130.

[71] C. Delvosalle, C. Fievez, A. Pipart, and B. Debray, "Aramis project: A comprehensive methodology for the identification of reference accident scenarios in process industries," *Journal of Hazardous Materials*, vol. 130, no. 3, pp. 200–219, 2006.

[72] Q. Qiu, D. A. Ramirez, S. Saleh, G. G. Fluet, H. D. Parikh, D. Kelly, and S. V. Adamovich, "The new jersey institute of technology robot-assisted virtual rehabilitation (njit-ravr) system for children with cerebral palsy: a feasibility study," *Journal of neuroengineering and rehabilitation*, vol. 6, pp. 1–10, 2009.

[73] S. Kemna, P. R. Culmer, A. E. Jackson, S. Makower, J. F. Gallagher, R. Holt, F. Cnossen, J. A. Cozens, M. C. Levesley, and B. B. Bhakta, "Developing a user interface for the ipam stroke rehabilitation system," in *2009 IEEE International Conference on Rehabilitation Robotics*. IEEE, 2009, pp. 879–884.

[74] A. Mohebbi, "Human-robot interaction in rehabilitation and assistance: a review," *Current Robotics Reports*, vol. 1, pp. 131–144, 2020.

[75] L. Peng, Z.-G. Hou, N. Kasabov, L. Peng, J. Hu, and W. Wang, "Implementation of active training for an upper-limb rehabilitation robot based on impedance control," in *The 27th Chinese Control and Decision Conference (2015 CCDC)*. IEEE, 2015, pp. 5453–5458.

[76] J. Bai, A. Song, T. Wang, and H. Li, "A novel backstepping adaptive impedance control for an upper limb rehabilitation robot," *Computers & Electrical Engineering*, vol. 80, p. 106465, 2019.

[77] G. Xu, A. Song, and H. Li, "Adaptive impedance control for upper-limb rehabilitation robot using evolutionary dynamic recurrent fuzzy neural network," *Journal of Intelligent & Robotic Systems*, vol. 62, pp. 501–525, 2011.

[78] L. Pan, A. Song, G. Xu, H. Li, H. Zeng, and B. Xu, "Safety supervisory strategy for an upper-limb rehabilitation robot based on impedance control," *International Journal of Advanced Robotic Systems*, vol. 10, no. 2, p. 127, 2013.

[79] L. Sciavicco and B. Siciliano, *Modelling and control of robot manipulators*. Springer Science & Business Media, 2001.

[80] M. G. Rodd, "Introduction to robotics: Mechanics and control: John j. craig," 1987.

[81] Y. Elashry. (2022) Python-unity-socket-communication. GitHub repository. [Online]. Available: https://github.com/Siliconifier/Python-Unity-Socket-Communication

APPENDIX A VIRTUAL REALITY

The design of the VR environment was centered around three main pillars: user engagement, integration, and feedback. Translating these principles, the following key specifications were formulated:

1. **Trajectory Representation:** The VR platform displays the intended movement path for each specific exercise, allowing users to understand the movement goals and adjust their actions in real-time.
2. **Real-time Movement Feedback:** To enhance immersion and engagement, the VR environment provide the users with immediate visual feedback on their real-world movements, bridging the virtual and the physical realms.
3. **Real-time Performance Feedback:** An in-built mechanism presents users with instantaneous visual feedback on their performance. This not only serves as a performance metric but also as a motivational tool.
4. **External Device Integration:** The VR software communicates in real-time with the external devices used in the rehabilitation process, ensuring a seamless experience.
5. **User Performance Metrics:** This feature records, analyzes, and showcases their performance metrics post-session, crucial for therapists and patients to track progress.

Hardware

This research used the VIVE Pro 2, an advanced virtual reality headset, to materialize the above specifications. This headset's capabilities guarantee a fluid and immersive user experience. The technical specifications of the VIVE Pro 2 are detailed below:

Environment

The developed virtual reality environment tailored for the robotic rehabilitation platform is depicted in Figure A.2. Addressing the specifications mentioned earlier, the environment incorporates the following components:

1. The target trajectory: **black** 3D line.



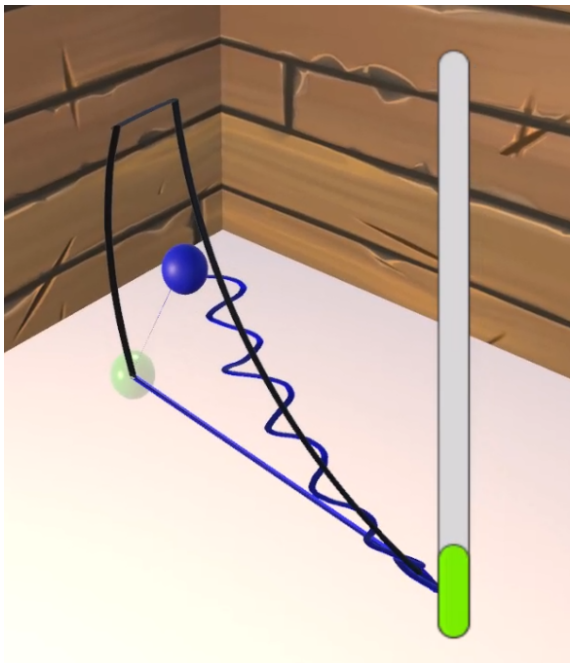
Figure A.1 The VIVE Pro 2 Virtual Reality headset

Specification	**Details**
Display	Dual RGB low persistence LCD
Resolution	2448 × 2448 pixels per eye (4896 x 2448 pixels combined)
Refresh Rate	90/120 Hz
Field of View	Up to 120 degrees (horizontal)
Audio	Hi-Res certified headset and headphones
Connections	Bluetooth, USB-C port for peripherals
Sensors	G-sensor, gyroscope, proximity, IPD sensor, SteamVR Tracking V2.0
Ergonomics	Adjustable IPD 57-70mm, Adjustable headphones and headstrap
Minimum Computer Specs	
Processor	Intel® Core™ i5-4590 or AMD Ryzen 1500 equivalent or greater
Graphics	NVIDIA® GeForce® GTX 1060 or AMD Radeon RX 480 equivalent or greater
Memory	8 GB RAM or more
Operating System	Windows® 11 / Windows® 10

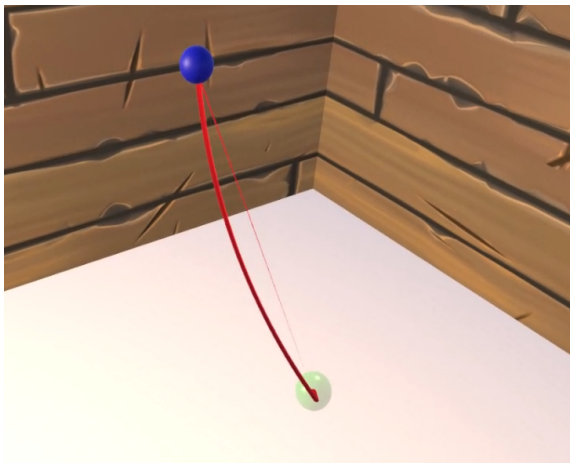
Table A.1 VIVE Pro 2 specifications

2. The current desired position: **green** sphere.
3. The user's arm's real-time position: **blue** sphere.
4. The trajectory resulting from the user's motion: **blue** 3D line.
5. User performance: position error gauge, which fills up and changes color from green to red as the position error increases.

During **trajectory generation**, the generated trajectory is represented by a **red** 3D line, with a **green** sphere representing the starting point and a **blue** sphere for the real-time position of the user's arm (Figure A.2b).



(a) VR environment during trajectory tracking



(b) VR environment during trajectory generation

Figure A.2 Robotic Rehabilitation Virtual Reality Environment

APPENDIX B KINOVA MICO²

The MICO² is a robotic manipulator designed by Kinova Robotics, a Canadian-based company specializing in the development of robotic arms for a wide range of applications. These applications span from industrial uses such as machine tending, to medical applications including surgical robots and assistive robots. The MICO² is a compact 6 DOF robotic arm equipped with a three-fingered end-effector. The general specifications of this robotic manipulator are presented in Table B.1. Figure B.1 displays the MICO² alongside the JACO², which is a larger version of the former.



Figure B.1 Kinova's MICO² (left) and JACO² (right) 6 DOF robotic manipulator

Specification	**Value**
Total Weight	without gripper: 3.6 Kg with gripper: 5.2 Kg
Payload	mid-range continuous payload : 2.1 Kg full-reach peak : 1.5 Kg
Reach	70 cm
Maximum Linear Arm Speed	20 cm/s
Power Supply Voltage	18 to 29 VD
Communication Protocol	RS485

Table B.1 Specifications of the Kinova MICO² 6 DOF robotic arm

The following tables presents the DH parameters of the MICO (Tables B.2 and B.3).

Table B.2 MICO 6 DOF curved wrist DH parameters

Joint	$\boldsymbol{\alpha}$ **(rad)**	$\mathbf{a_{i-1}}$ **(mm)**	$\mathbf{d_i}$ **(mm)**	$\boldsymbol{\theta_i}$
1	$\pi/2$	0	275.5	$q1$
2	π	290.0	0	$q2$
3	$\pi/2$	0	-7.0	$q3$
4	$\pi/3$	0	-166.1	$q4$
5	$\pi/3$	0	-85.6	$q5$
6	π	0	-202.8	$q6$

Table B.3 Transformation from DH algorithm to robotic arm physical angles

Transformation
$q_{1,\text{physical}} = -q_1$
$q_{2,\text{physical}} = q_2 + 90°$
$q_{3,\text{physical}} = q_3 - 90°$
$q_{4,\text{physical}} = q_4$
$q_{5,\text{physical}} = q_5 + 180°$
$q_{6,\text{physical}} = q_6 - 90°$

Hardware

As previously mentioned, the MICO robotic arm has 6 DOF. Its physical structure comprises six actuators, five links, a base, and an end-effector. Initially designed as a service device affixed to a wheelchair, the MICO's structure utilizes carbon fiber, a material that ensures robustness while maintaining a lightweight profile.

Starting from the base, the actuator models for joints #1, #2, and #3 are the K-75, while joints #4, #5, and #6 utilize the K-58 model. These actuators facilitate unrestricted joint rotations. Each joint is equipped with sensors that measure current, temperature, position, acceleration, and torque. The base of the robot houses the embedded controller, which operates at a CPU frequency of 360 MHz.

The base also serves as the communication hub, offering three different ports: a USB 2.0 port, an Ethernet 1 MBps port, and a joystick port. The system is designed to withstand various environmental conditions, offering IPX2 water resistance and operating efficiently in temperatures ranging from -10°C to 40°C.

Software

Kinova offers multiple ways to control the robot. The first is the joystick control, which allows the user to directly control the manipulator using the controller, the Software Development Kit (SDK) and an API. The first option will not be discussed.

Software Development Kit (SDK) The SDK offers a complete GUI to control and interact with the MICO by connecting the robot to a computer using the communication hub. Two main programs can be used : the Development Center and the Torque Console.

As the name implies, the latest allows to change the control mode of the robot to torque control. The command sent to the robot are torques command (in joint space) and force commands (in Cartesian space). Different control parameters can be adjusted. The safety parameters is related to how sensible the robot is to interaction forces and velocities.If the parameter is low (0), the robot will, for low interaction forces and velocities, switch from torque control to position control as a safety measure. On the other hand, if the parameter is high (1), the control mode will never switch, meaning that the safety measure is turned off. The vibration controller allows to eliminate the undesired vibration (e.g. jerk) when the robot is operating. The gravity vector and the payload applied to the end-effector can be defined. The actuator damping parameter allows to define the damping effect applied to the actuator.

The Development Center is a GUI that allows the user to change some control parameters, such as the PID values of the controller, monitor the different sensory data available and control the robotic device. In the *Monitoring* window, data from the sensors at each joint can be found under the *joint* tab. Using this data, the information is mapped in Cartesian space under the *Cartesian* tab. Both position and velocity control can be found for Angular and Cartesian mode under the *Virtual Joystick* window. This allows the user to control the robotic manipulator using the desired mode. Finally, some examples of API codes can be found under the *Examples* section on the main page. The SDK was used to control the robot and to get familiar with the robot's behavior under different control modes.

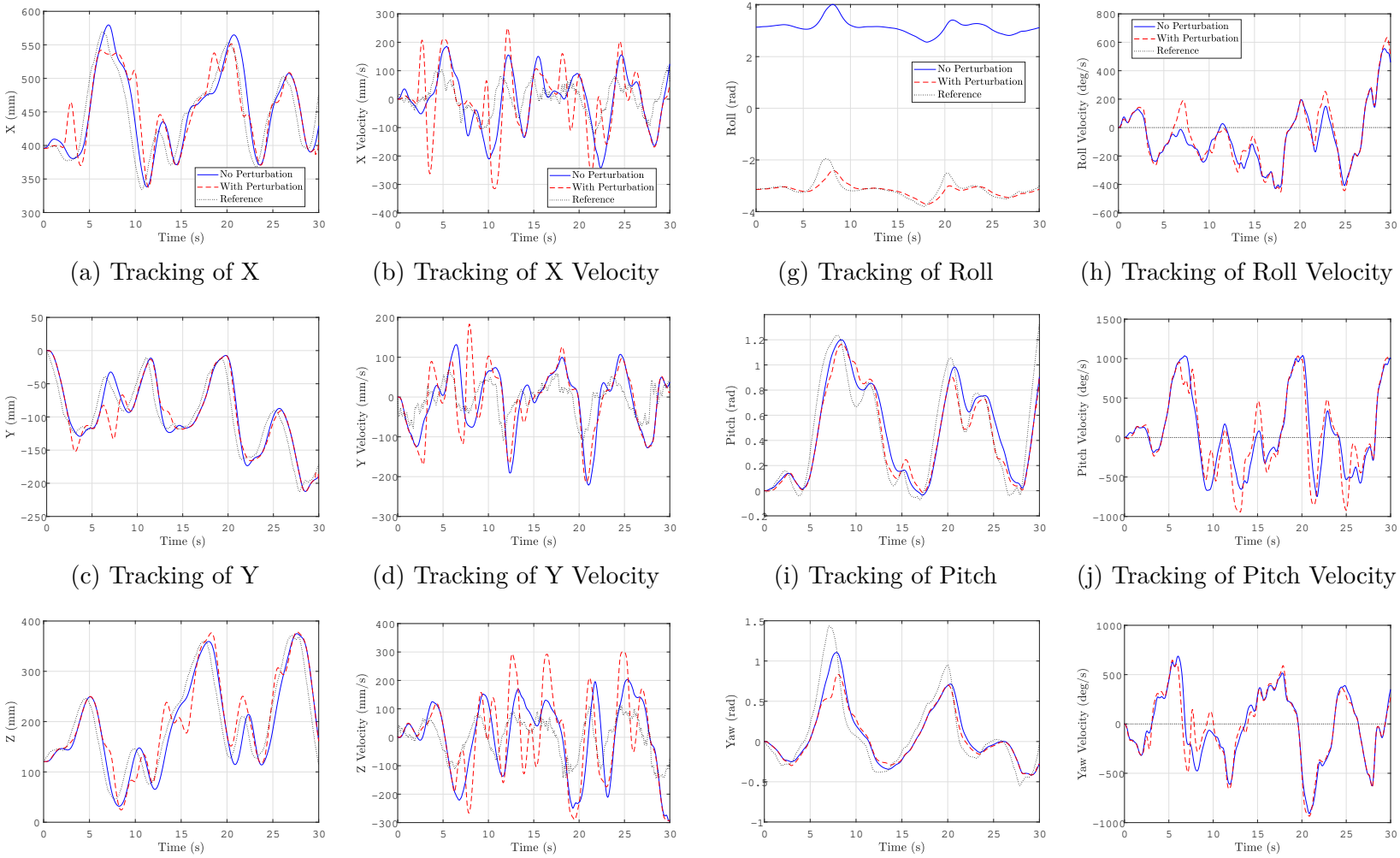
Kinova API Kinova provides an API to communicate and command the robotic manipulators. The API is built in C++. Examples of different controls, such as admittance, angular, or cartesian control, are constructed using API functions. The internal communication protocol for controlling the actuators is RS485, but it's also possible to interact with the actuators using an RS485 interface or USB-RS485 communications module. Kinova provides a detailed guide on constructing the RS485 message format.

Communication Methods and Control System : The digital processing unit (DSP) in the base of the arm serves as the "brain" of MICO2, processing received data and transmitting commands to the actuators. It's also responsible for gravity compensation, mode switching management, and advanced algorithms ensuring the arm's safety and task execution.

Communication with the arm can be established through RS485, joystick, or API-MICO2 methods. The Kinova API and SDK are compatible with both Windows and Linux. A sequence of actions, including loading necessary libraries and initializing functions, must be taken to connect to the arm.

The control system frequency using the API varies between 100 and 500 Hz, depending on the communication method. High-level communication via USB ranges from 100-500 Hz, while low-level communication directly with the actuators is 500 Hz. The advantages of using the low-level API with RS485 are clear, but this work followed a high-level approach with a refresh rate of 100 Hz, due to the availability and accessibility of the High-level API.

APPENDIX C EXPERIMENT 4: TRAJECTORY TRACKING CARTESIAN IMPEDANCE CONTROL

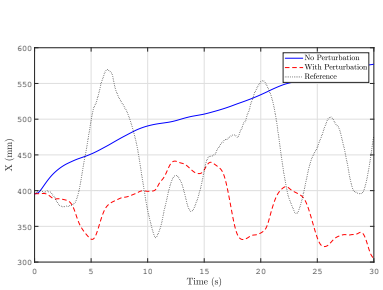


(a) Tracking of X (b) Tracking of X Velocity (g) Tracking of Roll (h) Tracking of Roll Velocity

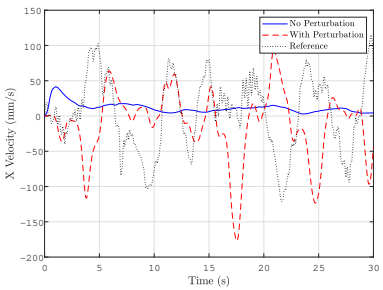
(c) Tracking of Y (d) Tracking of Y Velocity (i) Tracking of Pitch (j) Tracking of Pitch Velocity

(e) Tracking of Z (f) Tracking of Z Velocity (k) Tracking of Yaw (l) Tracking of Yaw Velocity

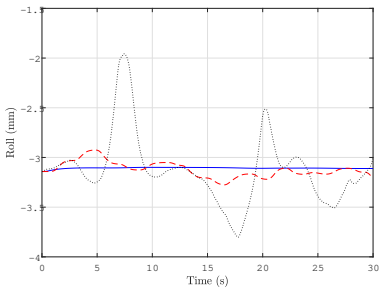
Figure C.1 Tracking performance using the Cartesian impedance controller with the Z_{high} parameters. Position and velocity tracking for translation degrees of freedom (DOFs) in the X, Y, and Z directions, and orientation and angular velocity tracking for rotational DOFs (Roll, Pitch, Yaw). The results demonstrate the controller's efficacy in maintaining the desired trajectory within the high impedance setting, with and without perturbations



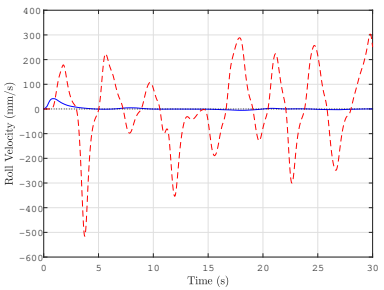
(a) X Position



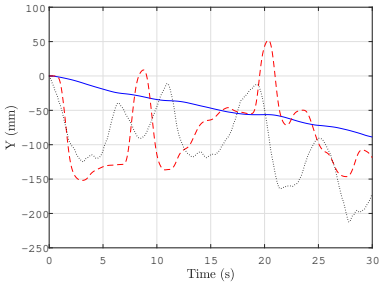
(b) X Velocity



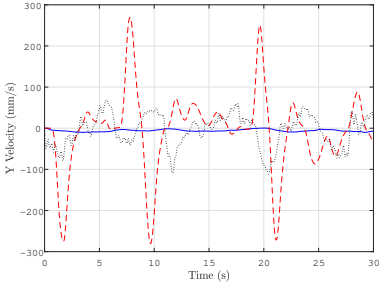
(a) Roll Position



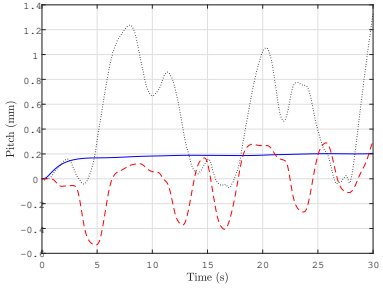
(b) Roll Velocity



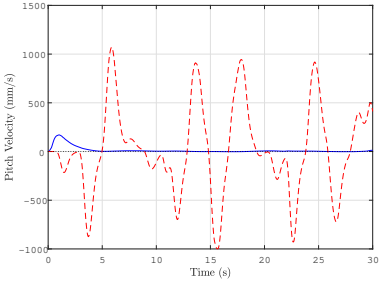
(c) Y Position



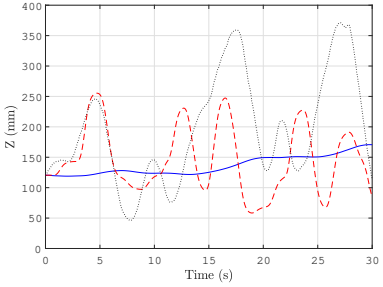
(d) Y Velocity



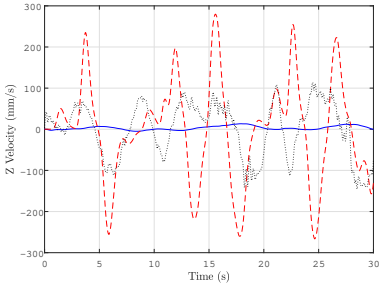
(c) Pitch Position



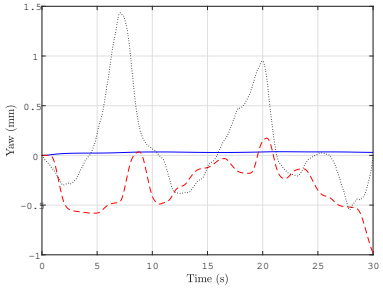
(d) Pitch Velocity



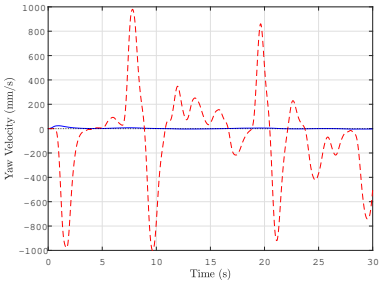
(e) Z Position



(f) Z Velocity

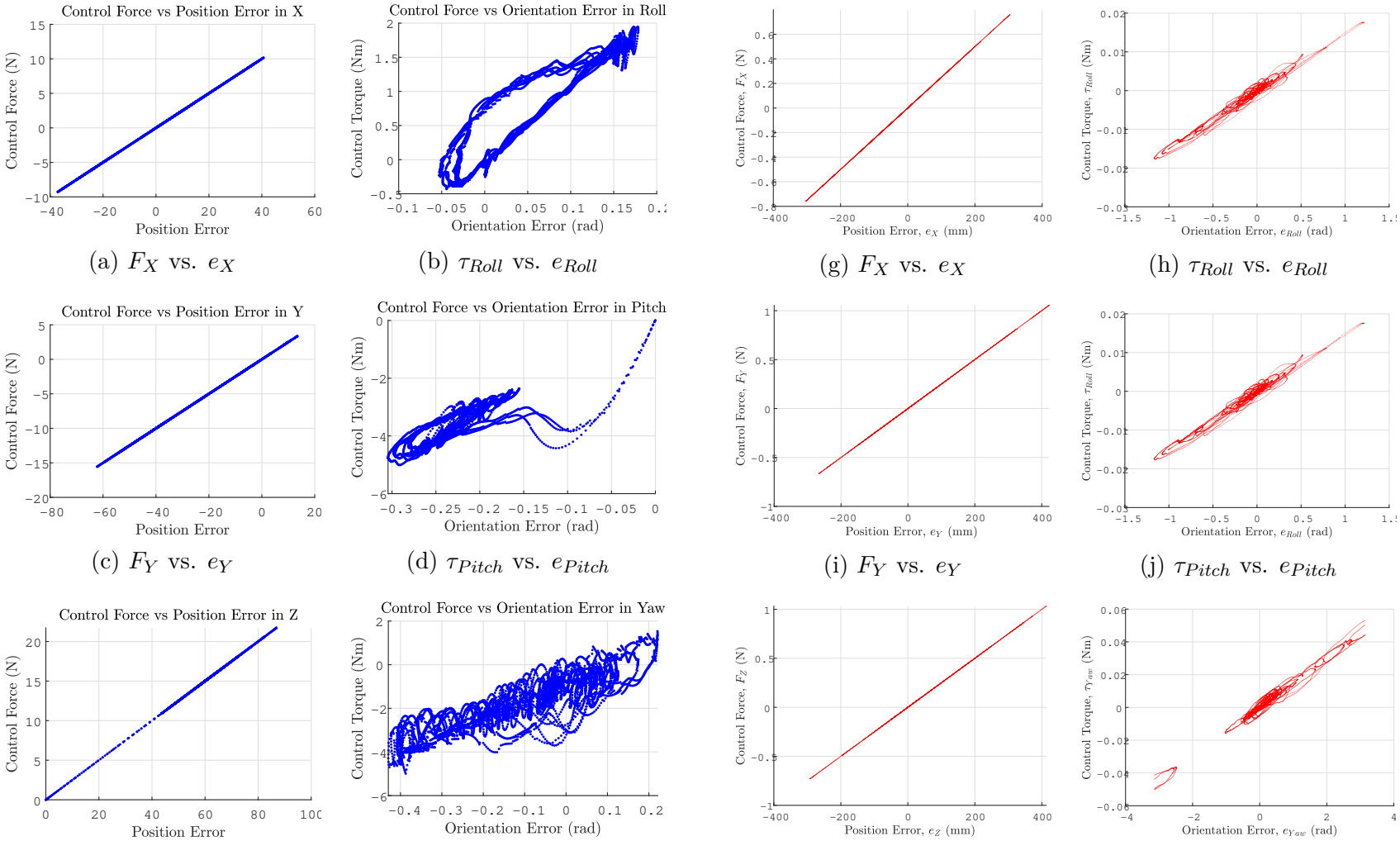


(e) Yaw Position



(f) Yaw Velocity

Figure C.2 Tracking performance using the Cartesian impedance controller with the Z_{low} parameters. **(a)-(f)** Position and velocity tracking for translation degrees of freedom (DOFs) in the X, Y, and Z directions. **(g)-(l)** Orientation and angular velocity tracking for rotational DOFs (Roll, Pitch, Yaw). The results demonstrate the controller's inability in maintaining the desired trajectory within the low impedance setting,



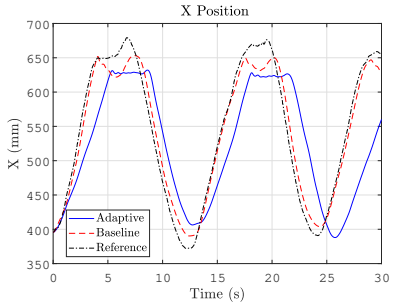
(a) F_X vs. e_X (b) τ_{Roll} vs. e_{Roll} (g) F_X vs. e_X (h) τ_{Roll} vs. e_{Roll}

(c) F_Y vs. e_Y (d) τ_{Pitch} vs. e_{Pitch} (i) F_Y vs. e_Y (j) τ_{Pitch} vs. e_{Pitch}

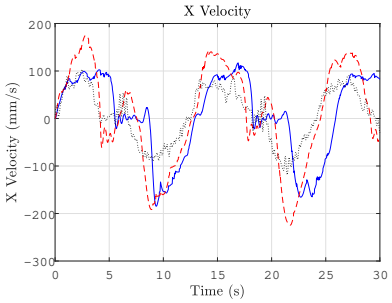
(e) F_Z vs. e_Z (f) τ_{Yaw} vs. e_{Yaw} (k) F_Z vs. e_Z (l) τ_{Yaw} vs. e_{Yaw}

Figure C.3 Relationship between position error and control force from the impedance controller. **(a)-(f)** Display the results with high impedance values (Z_{high}). **(g)-(l)** Show the results with low impedance values (Z_{low})

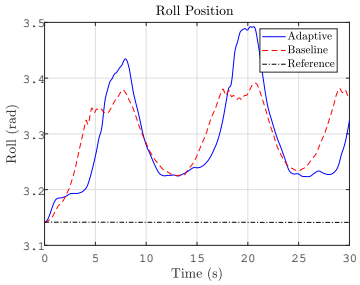
APPENDIX D EXPERIMENT 5 : ADAPTIVE CARTESIAN IMPEDANCE CONTROL



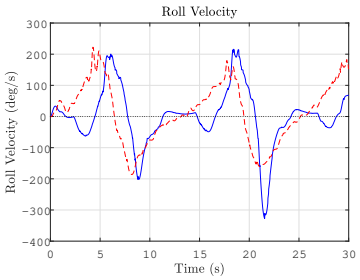
(a) Tracking of X



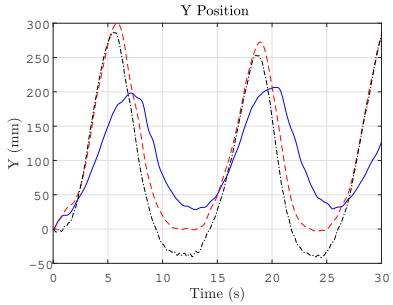
(b) Tracking of X



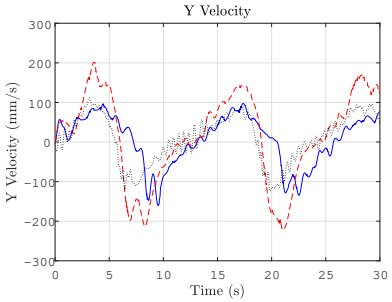
(g) Tracking of Roll



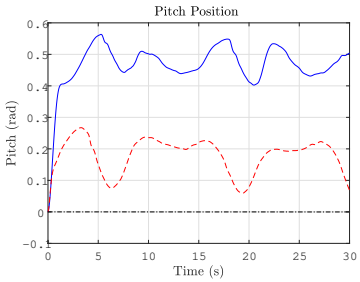
(h) Tracking of Roll Velocity



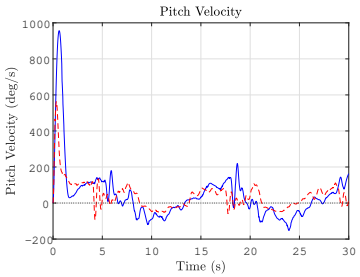
(c) Tracking of Y



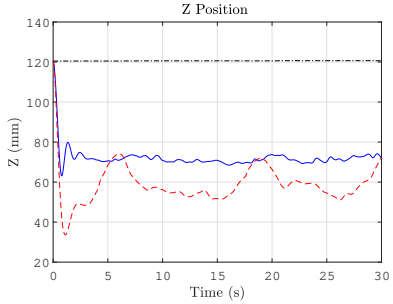
(d) Tracking of Y



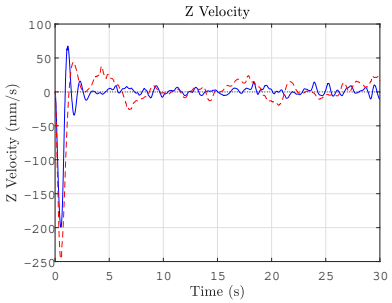
(i) Tracking of Pitch



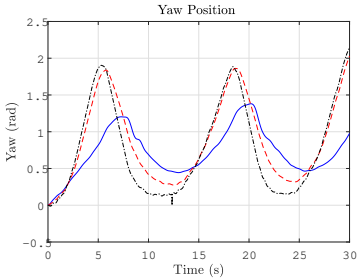
(j) Tracking of Pitch Velocity



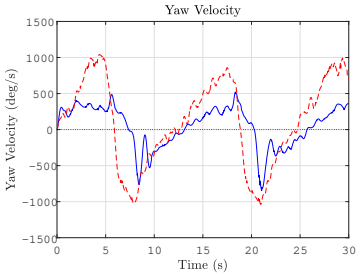
(e) Tracking of Z



(f) Tracking of Z

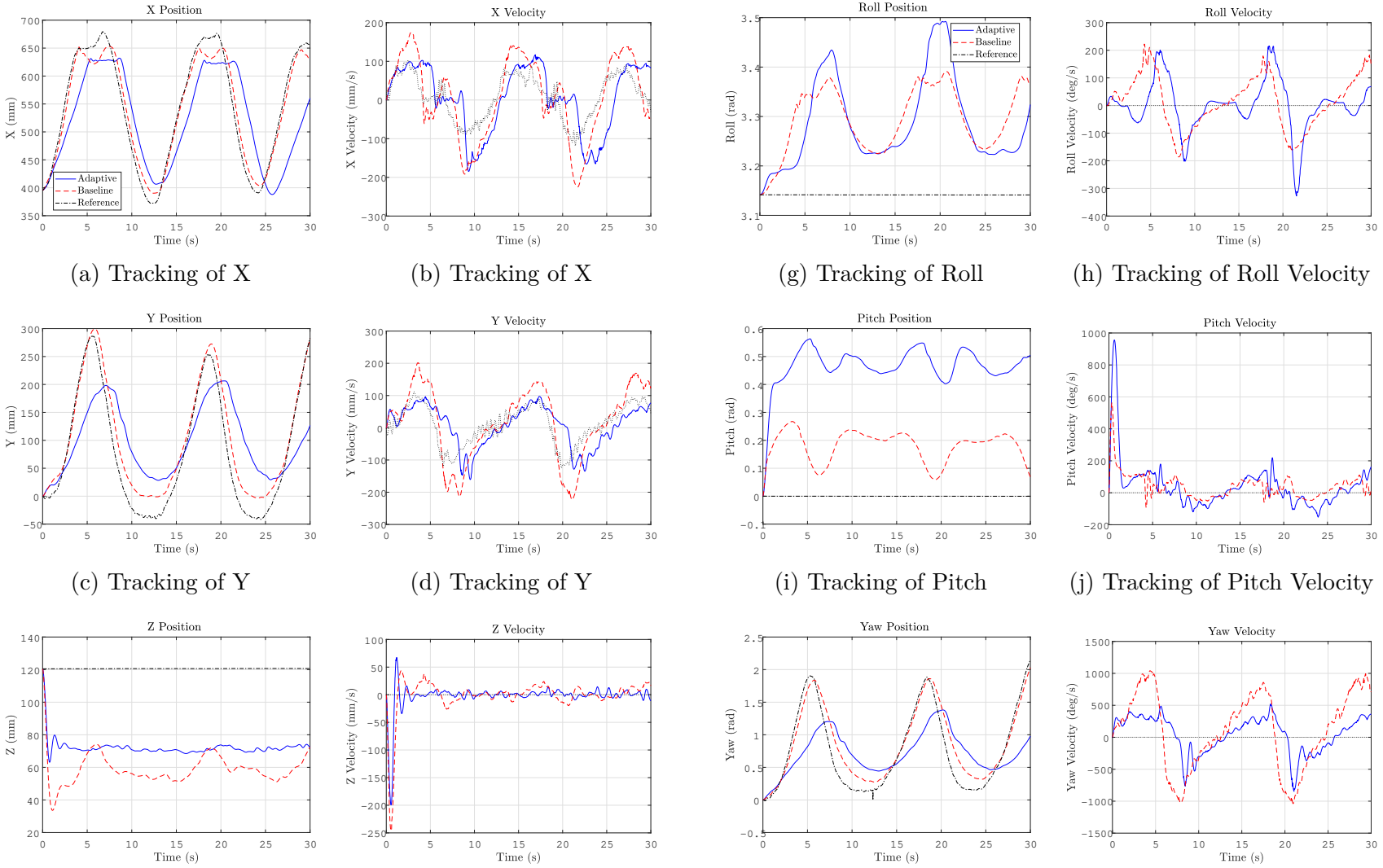


(k) Tracking of Yaw



(l) Tracking of Yaw Velocity

Figure D.1 Tracking performance using the Cartesian impedance controller, with adaptive parameters and passive parameters. **(a)-(f)** Position and velocity tracking for translation degrees of freedom (DOFs) in the X, Y, and Z directions. **(g)-(l)** Orientation and angular velocity tracking for rotational DOFs (Roll, Pitch, Yaw). The results demonstrate the controller's efficacy in maintaining the desired trajectory using the adaptive Impedance Control for passive rehabilitation exercises

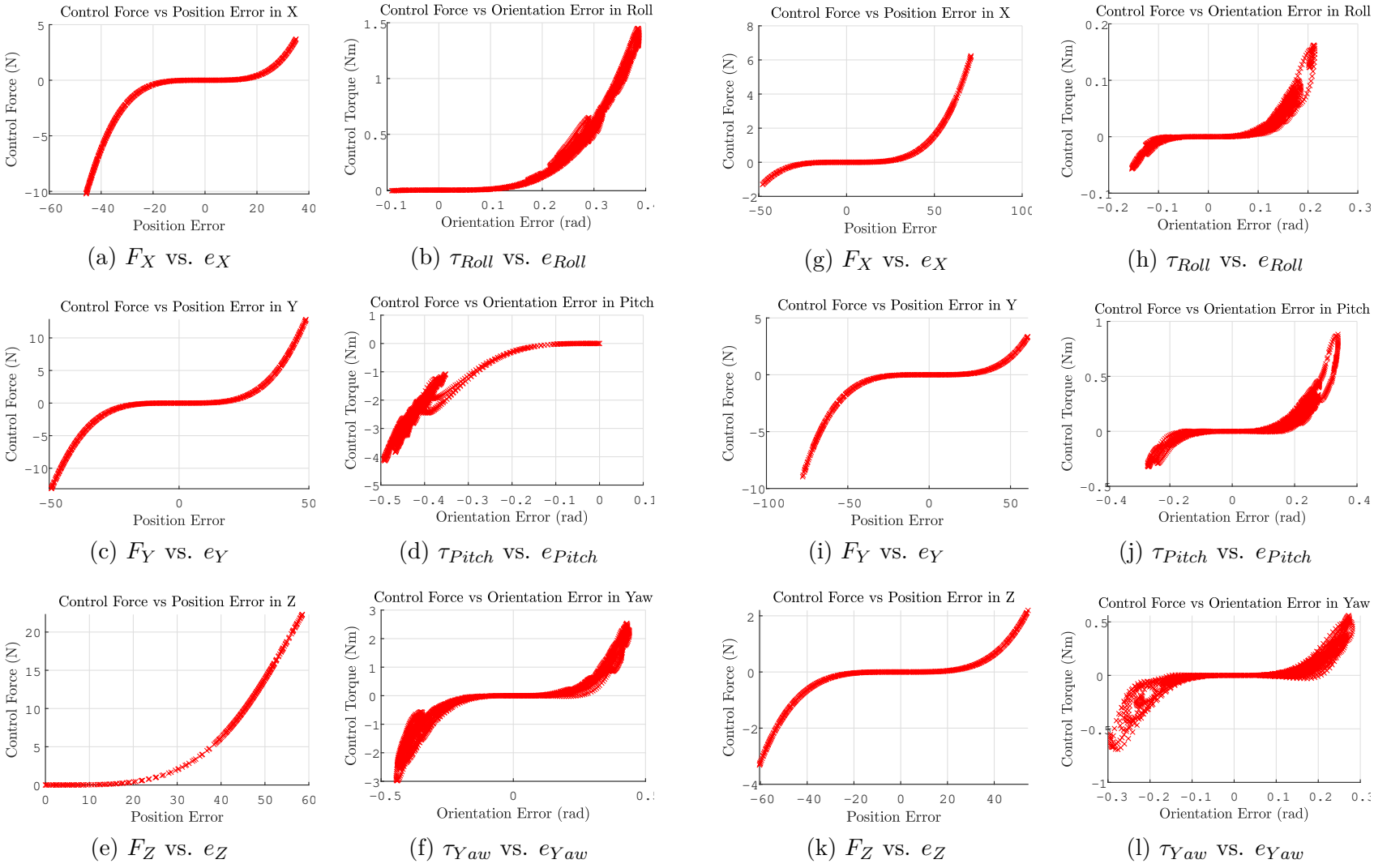


(a) Tracking of X (b) Tracking of X (g) Tracking of Roll (h) Tracking of Roll Velocity

(c) Tracking of Y (d) Tracking of Y (i) Tracking of Pitch (j) Tracking of Pitch Velocity

(e) Tracking of Z (f) Tracking of Z (k) Tracking of Yaw (l) Tracking of Yaw Velocity

Figure D.2 Tracking performance using the Cartesian impedance controller, with adaptive parameters and passive parameters. **(a)-(f)** Position and velocity tracking for translation degrees of freedom (DOFs) in the X, Y, and Z directions. **(g)-(l)** Orientation and angular velocity tracking for rotational DOFs (Roll, Pitch, Yaw). The results demonstrate the controller's efficacy in maintaining the desired trajectory using the adaptive Impedance Control for passive rehabilitation exercises

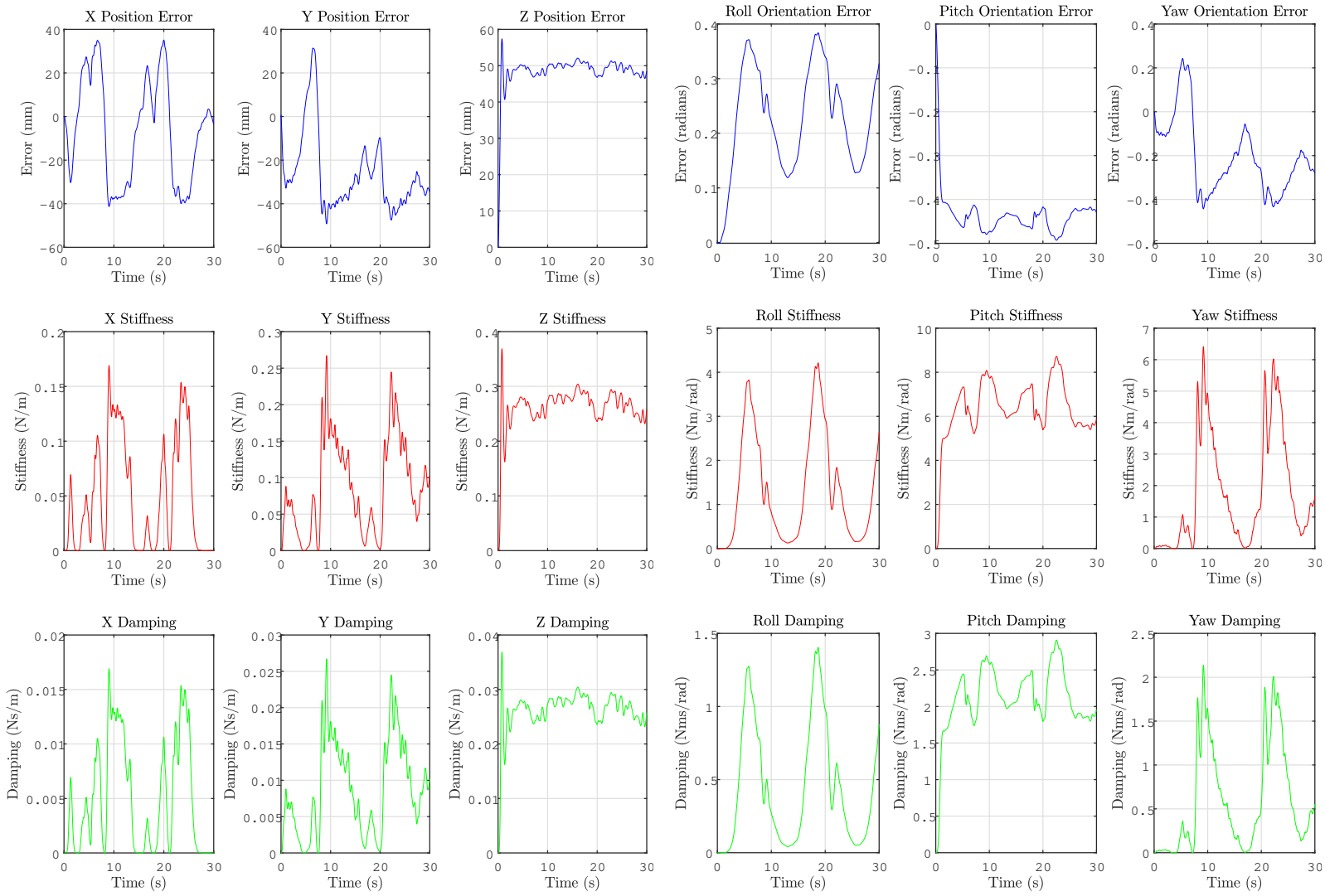


(a) F_X vs. e_X (b) τ_{Roll} vs. e_{Roll} (g) F_X vs. e_X (h) τ_{Roll} vs. e_{Roll}

(c) F_Y vs. e_Y (d) τ_{Pitch} vs. e_{Pitch} (i) F_Y vs. e_Y (j) τ_{Pitch} vs. e_{Pitch}

(e) F_Z vs. e_Z (f) τ_{Yaw} vs. e_{Yaw} (k) F_Z vs. e_Z (l) τ_{Yaw} vs. e_{Yaw}

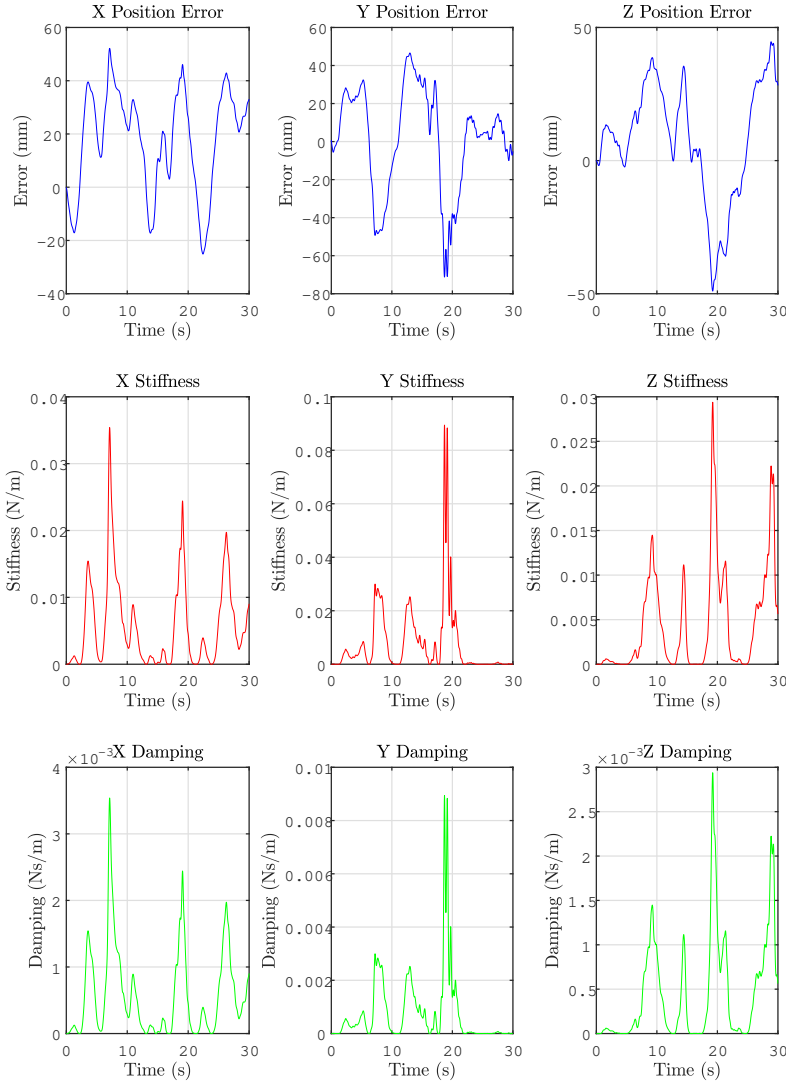
Figure D.3 Relationship between position error and control force from the impedance controller. **(a)-(f)** Display the results for the passive shoulder flexion exercise. **(g)-(l)** Show the results for the Can Off Shelf exercise



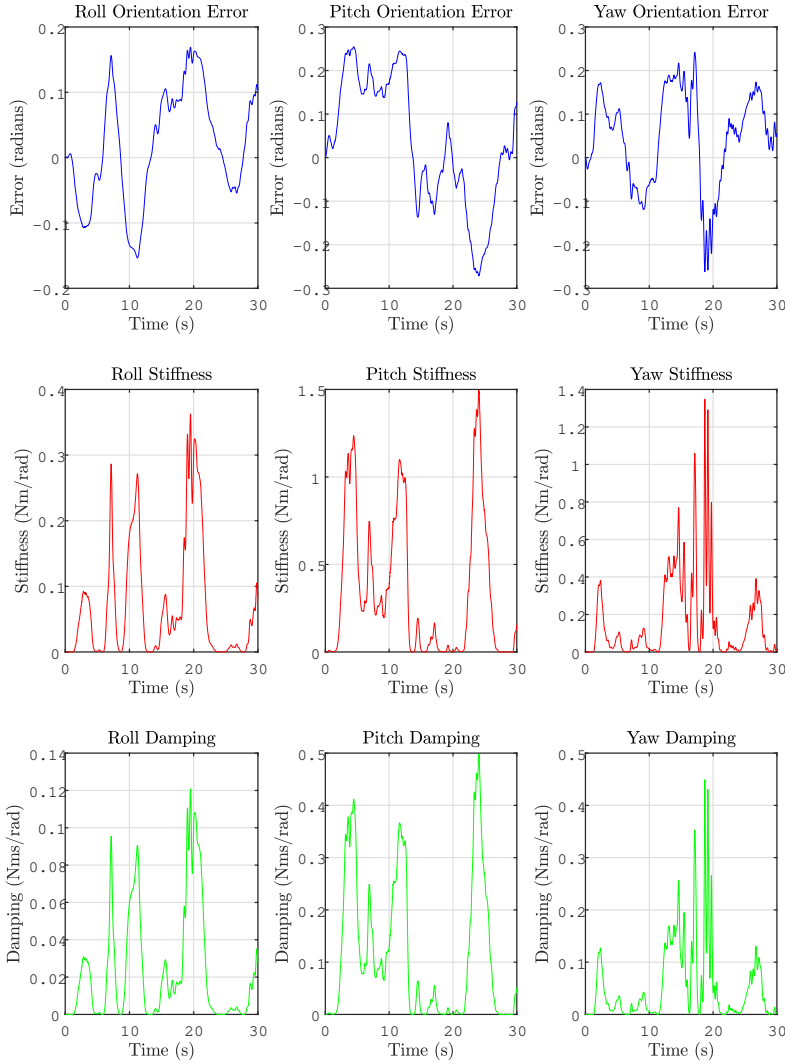
(a) Translation impedance parameters during the shoulder flexion exercise

(b) Orientation impedance parameters during the shoulder flexion exercise

Figure D.4 Variable Impedance Control Parameters : this figure shows the variation of the impedance parameters throughout the shoulder passive exercise



(a) Translation impedance parameters during *Can Off Shelf* exercise



(b) Orientation impedance parameters during *Can Off Shelf* exercise

Figure D.5 Variable Impedance Control Parameters : this figure shows the variation of the impedance parameters throughout the *Can Off Shelf* exercise