Learning-based Upper Limb Robotic Rehabilitation

by

David BEDOLLA MARTINEZ

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
IN PARTIAL FULFILLMENT FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
Ph.D.

MONTREAL, MARCH 14, 2024

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Maarouf Saad, Thesis supervisor
Department of electrical engineering at École de technologie supérieure

Mr. Cristóbal Ochoa Luna, Thesis Co-Supervisor
Tecnologico de Monterrey, School of Engineering and Sciences

Mr. Tony Wong, President of the Board of Examiners
Department of Systems Engineering at École de technologie supérieure

Mr. Ambrish Chandra, Member of the Jury
Department of electrical engineering at École de technologie supérieure

Mr. Mohammad Habibur Rahman, External Member
Department of Mechanical/Biomedical Engineering, University of Wisconsin-Milwaukee

Mr. Samir Ouaret, External Independent Examiner
Department of Electric Engineering, Collège Ahunstic

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON FEBRUARY 22, 2024

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

I would like to extend my heartfelt gratitude to every person and entity who has been instrumental in the successful completion of this project.

First and foremost, I am deeply thankful to my mentor and advisor, Dr. Maarouf Saad, whose unwavering support, patience, and guidance have been the cornerstone of my academic journey. Dr. Maarouf Saad not only provided valuable insights and academic expertise but also served as an inspiration with his unwavering commitment to my growth as a researcher. Words cannot fully describe how fortunate I feel to have been under his guidance, and I can say without a doubt that he is not only an excellent academic but also a remarkable person.

I also wish to extend my appreciation to Dr. Cristobal Ochoa Luna, my co-supervisor, who helped me understand the intricate architecture of the ETS-MARSE. His expertise and contributions have significantly enriched my work. I must also acknowledge the contributions of Dr. Mohammed Habibur Rahman, whose academic support and valuable insights were instrumental in developing the research presented here, and Dr. Yassine Kali for his invaluable assistance in the control theory aspect.

The constructive input and suggestions from my committee members, including Dr. Tony Wong, Dr. Ambrish Chandra, Dr. Mohammed Habibur Rahman, and Dr. Samir Ouaret, have been invaluable in shaping this research. Their willingness to dedicate their time and expertise to my project is greatly appreciated.

I would like to acknowledge the financial support provided by the National Council of Humanities, Sciences, and Technologies (CONAHCYT) from Mexico, formerly known as CONACYT, which played a pivotal role in sustaining my research efforts. The scholarship, numbered 739833, awarded by CONAHCYT, was crucial in making this project possible.

I extend my gratitude to my colleagues, including Chahira Arrami, Hazar Boughanmi, Abderaouf Termoul, Mahmoud Abdulkareem Yousef Abdallah, Al-Abduljaleel Saif Sinan Adnan, Victor Amador, for their assistance during tests, their camaraderie, shared experiences, and the

exchange of ideas. Their companionship and collaboration have made this research journey more rewarding.

My family, particularly my parents and siblings, and most notably my mother, Paula Martinez Lopez, have been a constant source of encouragement and support throughout this journey. Their unwavering belief in my abilities and their understanding during challenging times has been a driving force behind my achievements. Also thanks to my spiritual guide Angel Esteban Hernandez Sarabia.

Lastly, I want to thank my wife, Elizabeth Cortes Suarez, who has stood by my side, offering both emotional support and unwavering patience during the ups and downs of this academic endeavor. I must also express my gratitude to my two daughters, Daeli Nicole and Amanda; your love and encouragement have been my anchor.

This project would not have been feasible without the collective support I have received. I am profoundly grateful for their contributions and I eagerly look forward to staying connected as I continue my journey.

Réhabilitation robotique des membres supérieurs basée sur l'apprentissage

David BEDOLLA MARTINEZ

RÉSUMÉ

Dans cette thèse de doctorat, plusieurs avancées sont présentées pour améliorer les capacités d'un robot exosquelette à sept degrés de liberté (7 DDL) conçu pour la rééducation des membres supérieurs. Tout d'abord, une solution cinématique inverse de type humain, basée sur une technique d'apprentissage automatique, est introduite. Cette solution ouvre la voie à la génération de postures naturelles et confortables des membres supérieurs. Contrairement aux méthodes conventionnelles, cette approche offre un moyen pratique et efficace pour trouver des solutions cinématiques inverses appropriées pour la rééducation des membres supérieurs via de robot exosquelettes redondants. L'avantage central réside dans ses capacités en temps réel.

De plus, un contrôleur robuste basé sur l'apprentissage est développé pour traiter les incertitudes et les perturbations inhérentes à l'interaction patient-robot. Cela inclut la gestion des forces inconnues et l'assurance du respect des contraintes d'entrée et d'état prédéfinies. Ainsi, le contrôleur offre la sécurité fonctionnelle et des performances optimales lors des exercices de rééducation.

La dernière avancée dans cette thèse est la conception d'un système de rééducation miroir. Ce système vise à améliorer les compétences motrices des individus atteints d'hémiplégie par la stimulation de la neuroplasticité cérébrale. L'approche implique la fourniture de rétroaction visuelle et de stimulation proprioceptive tandis que les deux bras se déplacent de manière symétrique et simultanée. Cette méthodologie représente un grand pas en avant dans le domaine de la rééducation, exploitant le pouvoir de la neuroplasticité pour favoriser la récupération.

De plus, ces avancées s'adressent à la fois aux modes de rééducation passifs et actifs, permettant une approche flexible et personnalisée de la récupération. En mode passif, le patient peut se détendre tandis que l'exosquelette guide le bras le long d'une trajectoire prédéfinie. En mode actif, le porteur gagne de l'indépendance pour initier des mouvements et accomplir des tâches souhaitées sans assistance externe, lui permettant de prendre en main son processus de rééducation. Afin de valider l'efficacité et l'applicabilité en temps réel de ces avancées, une série d'expérimentations a été réalisée. Les résultats de ces expériences ont été documentés, soumis et/ou publiés dans plusieurs revues, contribuant à la communauté scientifique et au domaine de la robotique de rééducation. Les avancées présentées ici ouvrent de nouvelles perspectives pour améliorer la qualité de vie et les perspectives de récupération des personnes ayant besoin de rééducation, posant les bases pour un avenir prometteur dans le domaine de la rééducation assistée par des exosquelettes robotiques.

Mots-clés: exosquelette robotique, rééducation du membre supérieur

Learning-based Upper Limb Robotic Rehabilitation

David BEDOLLA MARTINEZ

ABSTRACT

In this doctoral thesis, several advancements are presented to enhance the capabilities of a seven DoF robotic exoskeleton designed for upper-limb rehabilitation. First and foremost, a human-like inverse kinematic solution, underpinned by a machine-learning technique, is introduced. This solution paves the way for generating natural and comfortable upper-limb postures. Unlike conventional methods, this approach provides a practical and efficient means of finding appropriate inverse kinematic solutions for upper-limb rehabilitation via redundant robotic exoskeletons. The core advantage lies in its real-time capabilities.

Additionally, a robust learning-based controller is developed to tackle uncertainties and disturbances that are inherent in the patient-robot interaction. This includes addressing unknown forces and ensuring compliance with predefined input and state constraints. By doing so, the controller offers functional safety and optimal performance during rehabilitation exercises.

The last development in this thesis is the design of a mirror rehabilitation system. This system is devised with the aim of enhancing the motor skills of individuals with hemiplegia through brain neuroplasticity stimulation. The approach involves providing visual feedback and proprioceptive stimulation while both arms move symmetrically and simultaneously. This methodology represents a significant leap forward in rehabilitation, harnessing the power of neuroplasticity to drive recovery.

Furthermore, these advancements cater to both passive and active rehabilitation modes, allowing for a flexible and tailored approach to recovery. In the passive mode, the patient can relax while the exoskeleton guides the arm through a predefined trajectory. In the active mode, the wearer gains the independence to initiate movements and complete desired tasks without external assistance, empowering them to take charge of their rehabilitation journey. To validate the efficacy and real-world applicability of these advancements, a series of real-time experiments have been conducted. The results of these experiments were documented, submitted and/or published in several journals, contributing to the scientific community and the field of rehabilitation robotics. The advancements showcased herein open new horizons for enhancing the quality of life and recovery prospects for individuals in need of rehabilitation, setting the stage for a promising future in the field of robotic exoskeleton-assisted rehabilitation.

Keywords: robotic exoskeleton, upper-limb rehabilitation

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

LIST OF ALGORITHMS

Page

LIST OF ABBREVIATIONS

ETS	École de Technologie Supérieure
MARSE	Motion Assistive Robotic-exoskeleton for Superior Extremity
GREPCI	Groupe de Recherche en Électronique de Puissance et Commande Industrielle
DoF	Degree of Freedom
DH	Denavit-Hartenberg parameters
PD	Proportional-Derivative
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
CTC	Computed Torque Control
TDE	Time Delay Estimation
SMC	Sliding Mode Control
RMS	Root-Mean-Square
FPGA	Field Programmable Gate Array
EMG	Electromyographic
HIK	Human Inverse Kinematics
WMI	Wearer Motion Intention
MPC	Model Predictive Control
GP	Gaussian Process
ISM	Integral Sliding Mode
ERL	Exponential Reaching Law

LIST OF SYMBOLS AND UNITS OF MEASUREMENTS

m	meter
cm	centimeter
rad	radian
Nm	Newton meter
s or sec	second
μs	microsecond
$^\circ$ or deg	degree
N	Newton
kg	Kilogram
g	gram

SYMBOLS

XYZ	Cartesian coordinate system main axes
α_{i-1}	is the angle between Z_{i-1} and Z_i around X_{i-1}
a_{i-1}	is the distance from Z_{i-1} to Z_i along X_{i-1}
θ_i	is the angle between X_{i-1} and X_i around Z_i
d_i	is the distance from X_{i-1} to X_i along Z_i
${}_i^{i+1}T$	Homogeneous transformation matrix
τ	torque

INTRODUCTION

The pace of technological advancement has been rapid, progressing from the first industrial revolution in the 16th century to the era of Industry 4.0 (Oztemel & Gursev, 2020), bringing forth numerous technological challenges. Among these challenges, the development of robotics systems stands out as a crucial area (Gao *et al.*, 2020). Robotics finds applications in various fields, including flexible robots for medical purposes (Zhang & Lu, 2020), augmented reality applications (Ong *et al.*, 2020), autonomous systems in military applications (Torossian *et al.*, 2020), search and rescue operations during natural disasters (Ekambaram *et al.*, 2020), and manufacturing industries (Bhatt *et al.*, 2020).

Regarding robotic exoskeletons, they find applications in power amplification (Lee *et al.*, 2014) as well as rehabilitation purposes providing physical training to the biomechanical structure of human limbs (Brahmi *et al.*, 2018). Its primary objective is to assist individuals in their daily activities (Barrios-Muriel *et al.*, 2020). Utilizing exoskeletons for medical rehabilitation presents an alternative to traditional therapy methods and has proven to enhance the quality of life for patients by enabling long-term therapy (Brahmi *et al.*, 2016). This is particularly relevant considering the insufficient number of health centers (Teasell *et al.*, 2009) and professionals available to meet the growing rehabilitation demand (Islam *et al.*, 2020).

In robotic rehabilitation, passive and active modes are used to exercise the limb. Passive mode often serves as the initial stage of rehabilitation. In this mode, the robot follows a predefined trajectory and it is typically employed in the early weeks following an accident, such as a stroke, sports-related injury, trauma, spinal cord injury, or due to aging. The subsequent stage is the active mode which allows the patient to voluntarily initiate movements and accomplish tasks at will.

This thesis builds upon prior research conducted by (Rahman, 2012), (Ochoa-Luna, 2016), and (Brahmi, 2019) to develop a 7 Degrees-of-Freedom (DoF) exoskeleton robot, named

ETS-MARSE robot (Motion Assistive Robotic-Exoskeleton for Superior Extremity), for upper-limb rehabilitation, in the Groupe de Recherche en Électronique de Puissance et Commande Industrielle (GREPCI) at the École de Technologie Supérieure (ETS).

This thesis aims to introduce several developments for upper-limb rehabilitation, including an intelligent inverse kinematic solution, a learning-based robust control, and a mirror rehabilitation system for hemiplegic patients. These developments involve the measurement of the wearer's motion intention (WMI) using Electromyography (EMG) and Inertial Measurement Units (IMU) sensors. Furthermore, the proposed developments are implemented in real-time. The primary contribution of this work lies in enhancing the capabilities of the ETS-MARSE exoskeleton robot to assist individuals with physical disabilities in their rehabilitation process, thereby improving their quality of life.

This thesis is organized as follows: Chapter 1 describes and justifies the research problem, presents the state-of-the-art, and set the objectives and the original methodology for this work. Chapter 2 presents the theoretical background. Chapters 3, 4, 5, and 6 are the resultant papers that report the contributions made to the existing literature.

Chapter 3 presents a novel approach to obtaining natural human-type inverse kinematics solutions using a machine-learning technique. Additionally, a robust predictive feedback controller is enhanced with an integral sliding mode controller to reject uncertainties and disturbances while optimally performing rehabilitation exercises. Experimental results verify the effectiveness of the proposed approach.

Chapter 4 presents an improved version of the predictive feedback controller developed in Chapter 3, where an improved prediction model based on a machine-learning technique is incorporated to exploit the ETS-MARSE's physical capabilities optimally. This paper presents

the first hardware implementation of the Learning-Based Model Predictive Control on a highly nonlinear system such as the ETS-MARSE.

Chapter 5 presents a mirror rehabilitation system designed for hemiplegic wearers where the healthy arm generates a desired trajectory while the exoskeleton robot assists the impaired arm by tracking the mirrored trajectory. Additionally, a fixed-time super-twisting-like algorithm control with prescribed performance is proposed to account for uncertainties and disturbances in the system. The proposed approaches were implemented in real-time and the experimental results confirm their applicability.

Chapter 6 presents a standardized design framework for exoskeleton robots that facilitates the comparison between kinematic designs. Additionally, a novel inverse kinematic method is proposed by defining the natural elbow frame as additional information to obtain a human-type solution.

CHAPTER 1

PROBLEM STATEMENT

Advances in fast response protocols for stroke have increased the number of survivors, leading to an unprecedented burden on physiotherapists and hospitals (Johnson *et al.*, 2019). In order to address the needs of individuals with upper-limb impairments affecting shoulder, elbow, forearm, and wrist motions, this research aims to enhance the capabilities of an exoskeleton robot to provide both passive and active rehabilitation modes. To achieve this objective, three main components are required to enable collaborative interaction between the wearer and the robot. In this thesis, we redefine the components or schemes defined in prior research as follows:

The first component is a high-level scheme that generates a suitable desired joint-space trajectory, ensuring that the wearer's postures are natural for the upper limb. The challenge in this scheme arises from the redundancy nature of robotic exoskeletons with more than six degrees of freedom (DoF), as is the case here. Given a task-space target, there are infinitely many possible configurations that satisfy the required end-effector position and orientation. The method used to select the most natural solutions for a given upper-limb movement (e.g., diagonal reaching, pick and place, or elbow flexion/extension) should operate in real-time and provide unique solutions.

The second component is a low-level scheme that ensures robustness and optimal performance while tracking a desired trajectory and handling uncertainties and disturbances. This scheme faces the primary challenge of achieving optimal performance while dealing with unknown dynamics, including unmodeled dynamics (e.g., Coulomb friction), model uncertainties (e.g., unknown wearer physical characteristics), and external disturbances (e.g., unknown wearer efforts). These unknown dynamics can degrade performance and even lead to instability. Optimal performance involves factors such as fast convergence, accurate tracking, reduced control effort, and closed-loop stability. Additionally, it is crucial to consider physical constraints during rehabilitation tasks to ensure safety and operational functionality, including input constraints to prevent excessive control activity that could damage the motors, as well as trajectory and

limit constraints (position and velocity limits) to avoid harming the links and transmission mechanisms.

The third component is a task-level scheme that focuses on measuring the desired task-space trajectories based on measurements of the wearer's motor intention (WMI), utilizing techniques such as EMG and IMU signal processing. The challenge here is to measure the WMI in real-time with minimal delay between the intended movement and the actual movement of the robot. Additionally, the generated desired task-space trajectory should exhibit smoothness.

To address these challenges, in the first place this thesis proposes a novel intelligent approach to obtaining unique and natural inverse kinematic solutions for impaired upper limbs. Furthermore, a learning-based Model Predictive Controller is developed to track the trajectories generated by the previous contribution, enabling optimal performance in passive rehabilitation and upper-limb exercises. Moreover, a mirror rehabilitation system is designed for hemiplegic patients, where the healthy arm generates the desired trajectory while the impaired arm is assisted by an exoskeleton robot. This setup allows for symmetrical and simultaneous movement between both arms, facilitating the improvement of motor skills in the impaired arm.

1.1 Literature review

This section reviews the state of the art on the previously mentioned schemes for robotic exoskeletons.

1.1.1 High-level scheme

In the last decades, the human-like motion study has received attention in robotics (Billard *et al.*, 2006), for example, to give natural movement to Humanoid robots (Almasri & Ouezdou, 2008), (Bin *et al.*, 2011), and (Xu *et al.*, 2010). Several vision-based methods to measure and replicate human motion have been applied (Alibeigi *et al.*, 2017), (Date *et al.*, 2004), and (Poppe, 2007).

Particularly, in upper-limb rehabilitation, the human inverse kinematic problem requires unique anthropomorphic solutions that bring comfort and natural movement to the wearer. However, due to the redundancy of the upper-limb that includes the shoulder, elbow, and wrist (Benati *et al.*, 1980) there are an infinite number of solutions available. Therefore, the inverse kinematics of redundant robotic manipulators is a complex problem with mathematical solutions that are in general very difficult to derive. The redundancy provides several advantages such as a larger workspace, singularity space reduction, a high manipulability index, and dexterity capabilities (Rahman *et al.*, 2015) while working on task-space and joint-space, enhancing wearer comfort and improving physical interaction (Gunasekara *et al.*, 2012).

To solve the inverse kinematic problem, numerical iterative methods using the Jacobian and its pseudo-inverse have been proposed (Craig, 2005), (Kelly *et al.*, 2005) and (Klein & Huang, 1983), which are not suitable for real-time implementation for redundant robots. Human inverse kinematic modeling based on recorded data methods (Ficuciello *et al.*, 2014) with problems of model generalization (unable to generate new solutions). Optimization-based human inverse kinematics use different cost functions, for example, potential energy (Zhao *et al.*, 2014), joint displacement discomfort (Jung *et al.*, 1994) and (Jung *et al.*, 1995) with implementation in real-time (Tang *et al.*, 1998) but the importance of each cost function is still not well defined (Campos & Calado, 2009). Some approaches derive closed-form solutions using analytical and geometric procedures combined (Bin *et al.*, 2011) and (Soltani-Zarrin *et al.*, 2017), the analytical method gives multiple solutions and it is used when the desired point is within the workspace. Otherwise, the geometric solution is used. These combined geometric and analytical solutions (Brahmi *et al.*, 2017b), (Kim *et al.*, 2011) and (Tolani *et al.*, 2000) require to define the swivel angle (Tolani & Badler, 1996) to minimize the required energy (Almasri & Ouezdou, 2008) solving redundancy, achieving comfortable human solutions.

1.1.2 Low-level scheme

Linear approaches such as PI, PD, and PID controllers have been the most popular controllers in the industry due to their easiness of implementation. However, on nonlinear systems, these

linear approaches only have good performance around a fixed operation point and redesign should be done to adapt to new operation points and circumstances.

To work in a larger operation region, linearization techniques such as feedback linearization (Slotine & Li, 1991) have been proposed. This technique needs an accurate dynamic model to compensate for the non-linearities, but constantly changing operational circumstances make unfeasible its use. Usually, CTC is combined with adaptive techniques (Han *et al.*, 2020). To estimate the unknown dynamics online Time Delay Estimation (TDE) (Youcef-Toumi & Ito, 1990) and Backstepping (Marquez, 2003) had been proposed for high accuracy in the presence of uncertainties and disturbances. However, TDE suffers from the time delay error (TDR). To deal with unknown dynamics, the Sliding Mode Control (SMC) approach has been proposed for robustness.

However standard SMC suffers from a not defined finite-time convergence, lack of high-accuracy tracking, and chattering that can damage the motors. A combination of different approaches has been developed to obtain the benefits of these techniques. For finite-time convergence, Exponential Reaching Law Sliding Mode control (ERL-SMC) has been proposed (Fallaha *et al.*, 2011). For finite-time convergence, an adaptive approach Terminal Sliding Mode Control (TSMC) with TDE was proposed (Brahmi *et al.*, 2020). The adaptive approach Jacobian Transpose Terminal Sliding Mode Control with Time Delay Estimation (JSTDE) (Brahmi *et al.*, 2017a) was proposed for finite-time convergence while working in task space. To reduce the chattering problem and for high accuracy tracking trajectory in the presence of uncertainties and unknown disturbances, an adaptive Super Twisting Algorithm Control combined with TDE had been proposed (Kali *et al.*, 2018). To reduce chattering problems a Second-Order Sliding Mode Control (SOSMC) was proposed (Levant, 2007), but the second derivative of the sliding surface produces instability with small disturbances. To solve this problem, the SOSMC was combined with the TDE approach (Brahmi *et al.*, 2018).

Applications, where the control policy is learned, are Composite learning control (Xu *et al.*, 2020), reinforcement learning (Perrusquía *et al.*, 2020) only used with repetitive tasks, Neural

Control combined with SMC has been implemented (Jebri *et al.*, 2020) with problems of extreme control activity and chattering, Parametric iterative learning control (Xu & Jian-Xin, 2013) with problems of how to satisfy constraints to prevent damaging the plant, how to overcome uncertainties and how to choose the initial conditions for convergence.

The control techniques described above lack an internal safety verification process. For that reason, to get optimal control signals while ensuring constraint satisfaction and reliable operation, a Model Predictive Control (MPC) approach (Rawlings *et al.*, 2017) can be used. MPC is a model-based optimization approach for feedback systems. The theory of MPC is well established, some robust MPC approaches are Min-Max (Scokaert & Mayne, 1998) with problems of expensive computation, using tubes (Langson *et al.*, 2004) and one-step (Park *et al.*, 1999) with small attraction region. To ensure robustness and to satisfy constraints through the prediction of future behavior, an accurate model is needed for MPC. For robotic manipulators, the dynamical model's complexity can grow drastically with few DoFs and will be a problem while solving numerical optimization in real-time. Another approach based on machine learning methods can build accurate models from data.

Besides nominal machine learning models e.g. feedforward neural networks, probabilistic methods can assign an uncertainty value to each predicted state, making it easier to deal with insufficient data and/or to decide between high performance or conservative control (Ostafew *et al.*, 2016a) also known as cautious or dual control (Mesbah, 2018). Particularly, the Gaussian process stochastic model behaves well with high-dimensional data, missing data, attenuation of noise contribution, and small training data. The only disadvantage is that the model grows exponentially with the training data length (Nguyen-Tuong & Peters, 2011).

To build an accurate model, a hybrid model has been used to do predictions. This hybrid model is composed of a mathematical model and a learned disturbance model (Ostafew, 2010). This approach is called Robust Learning-based Model predictive control. The disturbance model can anticipate disturbances and allows the use of simple mathematical models without exact knowledge of plant dynamics (Ostafew *et al.*, 2016a). Learning-based Model predictive control

is applied to repetitive tasks, gathering data, and learning from trial to trial (Ostafew *et al.*, 2016a). This technique was applied to a mobile robot (Hewing *et al.*, 2020a), in autonomous racing (Kabzan *et al.*, 2019), and to quadrotors (Pereida & Schoellig, 2018) to enhance the performance and safety of operation. The dynamics of these robotic systems are much simpler than the 7 DoF exoskeleton dynamic model. In this thesis, a real-time implementation of the robust learning-based model predictive control for a 7 DoF robotic exoskeleton is presented to improve the control performance while achieving reliable rehabilitation tasks.

1.1.3 Task-level scheme

The exoskeleton's movement can be guided using biological, non-biological signals, or a combination of both (Gunasekara *et al.*, 2012). These signals can be used in exoskeleton robots with the idea of activating a predefined movement (passive rehabilitation), to help correct the posture or to avoid resistive forces with WMI for assistive or active rehabilitation. The most used biological signal is electromyography (EMG). A surface or intramuscular (invasive) electrode is used when a patient can perform some movements. Pattern-based EMG recognition and classification had been applied using fuzzy logic and artificial neural network (ANN) (Chan *et al.*, 2000) and (Lei, 2019), support vector machines (Bitzer & Van Der Smagt, 2006), k-nearest neighborhood (Khairuddin *et al.*, 2019), hidden Markov models (Meng *et al.*, 2010) and Gaussian regression model (Long *et al.*, 2018). Also, WMI estimation based on muscle model had been developed: Hill muscle model with genetic algorithm (Cavallaro & Rosen, 2006) and Bilinear models (Matsubara & Morimoto, 2013). The main problem with the mentioned techniques is the training process which depends on the user characteristics, task regime, and electrode placement (Chan *et al.*, 2000). Some of these are bulky, and unfeasible for real-time implementation (Lei, 2019). And practical implementation is complex, for example, in (Treussart *et al.*, 2020) a two-minute EMG recording was used to train four classes (relaxed, flexion, extension, and contraction) using eight electrodes for a single DoF.

For non-biological signals, the force sensor (Anam & Al-Jumaily, 2012) was used to deal with the variability of EMG signals from different users, low signal activity due to muscle disorders,

and changing measurements due to fatigue. And for IMU (Inertial Measurements Units) the feasibility of only upper-arm and forearm joint estimation using IMU (Inertial Measurement Units) and EMG sensors were studied for an 8 DoF exoskeleton (Blana *et al.*, 2016). Both the force sensor and IMU are tipically combined with admittance (Huang *et al.*, 2018) or impedance control (Zhang & Lu, 2020).

In this sense, few robotic systems are found in the literature for single and multi-joint mirror rehabilitation applications. In the single joint rehabilitation case, a force sensor is used for wrist flexion and extension movement (Kim & Kim, 2017). In (Yang *et al.*, 2021b) a combination of surface EMG and force sensor is used to control the force in the impaired elbow for flexion and extension movements. Similar works controlling a single joint can be found in the literature using superficial EMG for wrist (Zhang *et al.*, 2016), and elbow (Hajian *et al.*, 2021) (Zhang *et al.*, 2019a). For multi-joint mirror rehabilitation, a mirror rehabilitation system for upper-limb (Wang & Fu, 2011) was presented using only IMU sensors, later the system underwent an upgrade by integrating a motion prediction module based on surface EMG signals (Ren *et al.*, 2019) for an 11 DoF exoskeleton.

It is worth noting that the previous multi-joint systems lack a wrist orientation estimator from EMG signals. To that end, different EMG classifiers techniques have been proposed with good results such as Support Vector Machine (SVM) and Linear Discriminant Analysis (LDA) (Khushaba *et al.*, 2016), optimized artificial neural networks (Lima *et al.*, 2018) and Fuzzy c-means clustering (Momen *et al.*, 2007). For further information on EMG pattern recognition for wrist orientation, a recent review can be found in (Parajuli *et al.*, 2019).

1.1.4 Motivation

Social inclusion of upper-limb impaired individuals is possible by enhancing their motor skills ability through exoskeleton-based physical therapy. To that end, a human inverse kinematics solution with unique solutions and natural postures for the wearer is mandatory. Also, the specifications and requirements of the exercise task must be performed as directed by the health

professional. A robust control strategy is necessary for the desired performance even in the presence of uncertainties and disturbances while satisfying operation constraints. The real-world implementation process is not straightforward. The exoskeleton is a system with multiple inputs and multiple outputs (MIMO) subject to human-robot interaction forces and different wearer physiological characteristics. Most of the control strategies used on exoskeletons lack a constraint satisfaction system within the controller. Usually, the state and input constraints are handled after the control law is computed. Considering these issues, a robust model predictive control approach will be implemented to satisfy constraints while achieving optimal performance. Additionally, a sensing system able to measure the WMI to provide active rehabilitation is necessary.

1.1.5 Objectives

The main objective of this research is to help impaired people to enhance their motor capabilities in daily life activities after a stroke or injury. The specific objectives are as follows:

1. Find a Human inverse kinematics solution, avoiding singularities, and ensuring natural configurations based on analytical and machine learning methods.
2. Develop a learning-based MPC to ensure robustness, optimal performance, and reliable passive rehabilitation.
3. Develop a mirror rehabilitation system through the measurements of the wearer's motion intention using EMG and IMU sensors.
4. Implementation and experimentation of the proposed developments on the ETS-MARSE exoskeleton robot in real-time.

1.2 Methodology

To achieve the objectives mentioned above, this research is structured into four distinct stages, each depicted in different colors in Fig. 1.1. The first stage, indicated in orange, involves the development of a learning-based solution for human inverse kinematics. This solution addresses kinematic redundancy and produces natural postures for the wearer.

The second stage, depicted in green, encompasses the development and integration of robust learning-based model predictive control into the real-time system. The data collected from the trials will be utilized to train the Gaussian process disturbance model offline, leading to iterative performance enhancements.

The third stage, shown in blue, focuses on the sensing system responsible for signal processing from EMG and IMU sensors. These sensors are instrumental in measuring the wearer's motion intention while maintaining a low computational cost.

Finally, the fourth stage entails the real-time design, implementation, and experimentation to demonstrate the effectiveness of the proposed approaches.

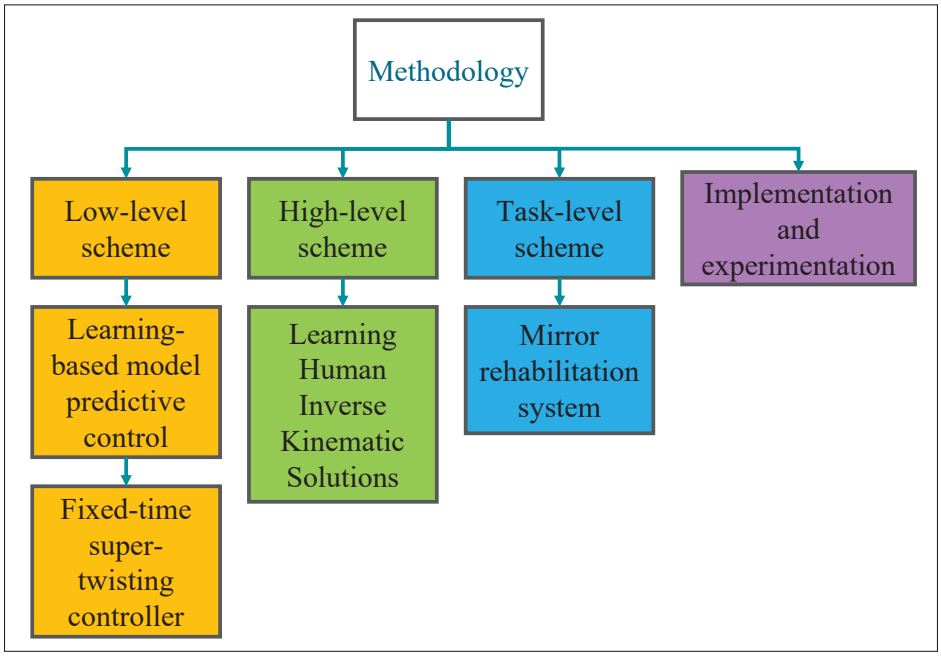

Figure 1.1 Methodology

1.2.1 Develop a human-type inverse kinematic intelligent solution

For the first stage, to execute any rehabilitation program, it is necessary to work with specific trajectories specified in Cartesian coordinates. These must be transformed into joint coordinates to enable smooth, singularity-free movements. Given the complexity of deriving complete

analytical solutions, implementing numerical methods for real-time control of seven Degrees of Freedom (DoF) robots is often unfeasible. Additionally, there is a lack of generalization in models trained with human-recorded data.

The identified challenges in this context include obtaining comfortable configurations and resolving the redundancy issue, where there are infinite solutions to a single problem. The initial phase of this research involves analyzing and deriving a geometric and analytical solution for the human inverse kinematics, addressing kinematic redundancy and achieving natural configurations. In this thesis, real-time implementation of human inverse kinematics using a combination of machine learning techniques and analytical methods was explored, with a focus on providing comfort and natural movement for the robot-wearer system. This approach is designed to deliver precise and accurate solutions while avoiding singularities.

Considering that the last three joints determine the orientation while joints one to four define the position, it is possible to develop analytical and geometrical closed-form solutions by decoupling the problem into two stages. A comparative analysis will be conducted, comparing the learning-based method, the minimum discomfort index, and the closest solution to previous ones.

1.2.2 Develop a learning-based MPC

The second stage in this thesis is the development and implementation of the robust learning-based model predictive control on the ETS-MARSE robotic exoskeleton. Taking advantage of the repetitive nature of physical therapy, a learning-based model predictive control will be implemented for optimal performance and reliable operation on rehabilitation tasks. To ensure robustness in the model predictive implementation, the model accuracy is the most important issue, which is solved using a hybrid model composed of a nominal mathematical model (beforehand) and a learned model (model from data trials). The learned model is based on the Gaussian processes model, a stochastic learning technique that enables the assignment of an amount of uncertainty to the computed output. This property allows measuring the

amount of safeness, in a specific robot configuration (Ostafew, 2010). The learned model is trained using the differences between the measured state and the estimated state (using the nominal mathematical model), this difference is called the disturbance model. Some interesting techniques that can be combined with LB-MPC are:

1. Super-twisting algorithm
2. Integral sliding mode
3. Exponential reaching law SMC

1.2.3 Develop a mirror rehabilitation system

The third stage presents an innovative mirror rehabilitation system designed for individuals recovering from hemiplegia after a stroke. In this system, the unaffected arm is responsible for generating a desired trajectory, and this trajectory is tracked using a Motion Capture (MoCap) system. Simultaneously, an upper-limb robotic exoskeleton assists the impaired arm in mimicking this trajectory, resulting in the attainment of natural and comfortable arm configurations. An outstanding feature of this system is the inclusion of a wrist orientation estimator, which is based on surface EMG signals from the muscles of the forearm.

To evaluate its practicality and effectiveness, experimental tests are conducted using a 7 Degrees of Freedom (DoF) exoskeleton robot, affirming its real-time efficiency and robust performance. In summary, this stage makes a significant contribution to the field by introducing a mirror rehabilitation system that not only focuses on achieving natural and comfortable postures, especially for the elbow and forearm but also integrates a wrist orientation estimator based on EMG signals from the forearm muscles.

1.2.4 Implementation and experimentation

Implementation of the proposed improvements experimentally concludes each previous stage.

1.3 Originality of the research contribution

The contributions and results of this research are shown in the following list.

1. A novel learning-based approach for human inverse kinematics that obtains natural and comfortable solutions while significantly reducing online computation burden.
2. A novel learning-based controller that incorporates a robust model predictive controller with an integral sliding mode controller. The Gaussian process technique was proposed to model measured uncertainty to enhance the prediction accuracy.
3. An innovative mirror rehabilitation system is designed to enhance the motor skills of individuals with hemiplegia through brain neuroplasticity stimulation.
4. A novel fixed-time super-twisting-like algorithm with prescribed performance is proposed to track a challenging trajectory while multiple joints are active simultaneously.
5. Experimental contribution developing passive and active rehabilitation capabilities using a seven DoF robotic manipulator.

CHAPTER 2

THEORETICAL BACKGROUND

2.1 Forward kinematics

2.1.1 Homogeneous transformation

The homogeneous transformation concept is very important for experts and beginners to develop and analyze robotic manipulators. Mathematically, a homogeneous transformation is defined as a 4×4 matrix representing position and orientation in the Cartesian space as follows:

$$T = \begin{bmatrix} R & P \\ 0_{1\times3} & 1 \end{bmatrix} \tag{2.1}$$

Where $R \in \mathbb{R}^{3\times3}$ is an orientation matrix and $P \in \mathbb{R}^3$ is a position vector. The term $0_{1\times3}$ represents an 1×3 zero matrix. Given a numerical homogeneous transformation matrix T, a visual representation of the frame (see Fig. 2.1) can be obtained by the following Matlab code:

```
figure
T=eye(4); %Homogeneous transformation
d=0.05;
plot3([T(1,4),T(1,4)+d*T(1,2)],[T(2,4),T(2,4)+d*T(2,2)],[T
   (3,4),T(3,4)+d*T(3,2)],'g','LineWidth',2); %y axis
hold on,xlabel('x'),ylabel('y'),zlabel('z')
plot3([T(1,4),T(1,4)+d*T(1,1)],[T(2,4),T(2,4)+d*T(2,1)],[T
   (3,4),T(3,4)+d*T(3,1)],'r','LineWidth',2); %x axis
plot3([T(1,4),T(1,4)+d*T(1,3)],[T(2,4),T(2,4)+d*T(2,3)],[T
   (3,4),T(3,4)+d*T(3,3)],'b','LineWidth',2); %z axis
axis square
```

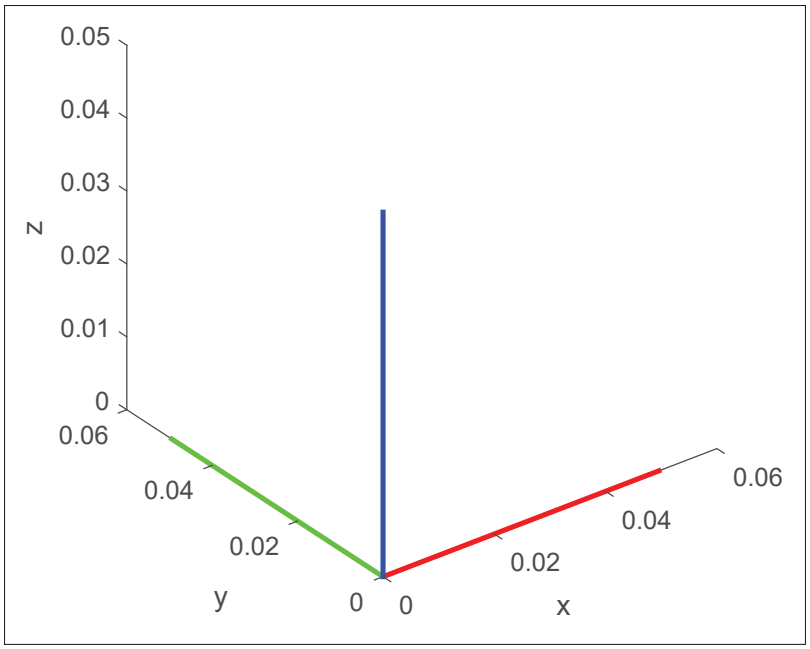

Figure 2.1 Homogeneous transformation as a frame (red for x-axis, green for y-axis and blue for z-axis)

The frame shown in Fig. 2.1 can be rotated or translated at will using the basic homogeneous transformation matrices $T_x(\cdot)$, $T_y(\cdot)$, and $T_z(\cdot)$ as rotation matrices and $T_D(\cdot_x, \cdot_y, \cdot_z)$ as translation matrix, around and along the xyz axes respectively.

$$
\begin{aligned}
T_x(\cdot) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c(\cdot) & -s(\cdot) & 0 \\ 0 & s(\cdot) & c(\cdot) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_y(\cdot) = \begin{bmatrix} c(\cdot) & 0 & s(\cdot) & 0 \\ 0 & 1 & 0 & 0 \\ -s(\cdot) & 0 & c(\cdot) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
T_z(\cdot) &= \begin{bmatrix} c(\cdot) & -s(\cdot) & 0 & 0 \\ s(\cdot) & c(\cdot) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_D(\cdot_x, \cdot_y, \cdot_z) = \begin{bmatrix} 1 & 0 & 0 & \cdot_x \\ 0 & 1 & 0 & \cdot_y \\ 0 & 0 & 1 & \cdot_z \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{2.2}
$$

The terms $s(\cdot)$ and $c(\cdot)$ denote the $sin(\cdot)$ and $cos(\cdot)$ functions respectively.

2.1.2 Forward and inverse Euler angles

Euler angles are employed to represent orientation in Cartesian space. While various conventions exist for Euler angles, this section will specifically describe two of them. The first variation is determined by the order of operations, which follows the sequence ZYX, as follows:

$$T_z(\gamma_d)T_y(\beta_d)T_x(\alpha_d) = \begin{bmatrix} c(\beta_d)c(\gamma_d) & c(\gamma_d)s(\alpha_d)s(\beta_d) - c(\alpha_d)s(\gamma_d) & s(\alpha_d)s(\gamma_d) + c(\alpha_d)c(\gamma_d)s(\beta_d) & 0 \\ c(\beta_d)s(\gamma_d) & c(\alpha_d)c(\gamma_d) + s(\alpha_d)s(\beta_d)s(\gamma_d) & c(\alpha_d)s(\beta_d)s(\gamma_d) - c(\gamma_d)s(\alpha_d) & 0 \\ -s(\beta_d) & c(\beta_d)s(\alpha_d) & c(\alpha_d)c(\beta_d) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

In this context, the symbols γ_d, β_d, and α_d represent the desired Euler angles used to construct a desired orientation matrix. Additionally, the Euler angles following the order ZYX can be computed for any desired orientation matrix denoted as R_d, defined by:

$$R_d = \begin{bmatrix} r_{11d} & r_{12d} & r_{13d} \\ r_{21d} & r_{22d} & r_{23d} \\ r_{31d} & r_{32d} & r_{33d} \end{bmatrix} \tag{2.4}$$

Here, r_{ijd} represents the ij element of the desired orientation matrix. The Euler angles can be computed by solving Equation (2.5) using the atan2 function, as shown in Equation (2.6).

$$T_z(\gamma_d)T_y(\beta_d)T_x(\alpha_d) = \begin{bmatrix} R_d & 0_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \tag{2.5}$$

$$\begin{aligned} \gamma_d &= a\text{tan2}(r_{21d}, r_{11d}) \\ \beta_d &= a\text{tan2}(-r_{31d}s(\gamma_d), r_{21d}) \\ \alpha_d &= a\text{tan2}(r_{32d}, r_{33d}) \end{aligned} \tag{2.6}$$

It's important to note that the atan2 function effectively manages situations involving division by zero. The second variation, applied in the context of the ETS-MARSE, follows the order XYZ,

as shown below:

$$T_x(\alpha_d)T_y(\beta_d)T_z(\gamma_d) = \begin{bmatrix} c(\beta_d)c(\gamma_d) & -c(\beta_d)s(\gamma_d) & s(\beta_d) & 0 \\ c(\alpha_d)s(\gamma_d) + c(\gamma_d)s(\alpha_d)s(\beta_d) & c(\alpha_d)c(\gamma_d) - s(\alpha_d)s(\beta_d)s(\gamma_d) & -c(\beta_d)s(\alpha_d) & 0 \\ s(\alpha_d)s(\gamma_d) - c(\alpha_d)c(\gamma_d)s(\beta_d) & c(\gamma_d)s(\alpha_d) + c(\alpha_d)s(\beta_d)s(\gamma_d) & c(\alpha_d)c(\beta_d) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.7}$$

Similarly to the ZYX Euler angles order, when dealing with a given desired orientation matrix, the Euler angles that adhere to the order XYZ can be determined by solving Equation (2.8) with the assistance of the atan2 function, as illustrated in Equation (2.9).

$$T_x(\alpha_d)T_y(\beta_d)T_z(\gamma_d) = \begin{bmatrix} R_d & 0_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \tag{2.8}$$

$$\begin{aligned} \alpha_d &= a\text{tan2}(-r_{23d}, r_{33d}) \\ \beta_d &= a\text{tan2}(r_{13d}c(\alpha_d), r_{33d}) \\ \gamma_d &= a\text{tan2}(-r_{12d}, r_{11d}) \end{aligned} \tag{2.9}$$

Furthermore, the desired orientation can also be expressed using a unit Quaternion denoted as $\hat{\mathbf{q}} = [\hat{q}_1, \hat{q}_2, \hat{q}_3, \hat{q}_4]^T$. This Quaternion can be converted into a homogeneous transformation matrix, as outlined in (Kuipers, 1999), as follows:

$$\begin{bmatrix} 1 - 2\hat{q}_3^2 - 2\hat{q}_4^2 & 2\hat{q}_2\hat{q}_3 - 2\hat{q}_1\hat{q}_4 & 2\hat{q}_2\hat{q}_4 + 2\hat{q}_1\hat{q}_3 & 0 \\ 2\hat{q}_2\hat{q}_3 + 2\hat{q}_1\hat{q}_4 & 1 - 2\hat{q}_2^2 - 2\hat{q}_4^2 & 2\hat{q}_3\hat{q}_4 - 2\hat{q}_1\hat{q}_2 & 0 \\ 2\hat{q}_2\hat{q}_4 - 2\hat{q}_1\hat{q}_3 & 2\hat{q}_3\hat{q}_4 + 2\hat{q}_1\hat{q}_2 & 1 - 2\hat{q}_2^2 - 2\hat{q}_3^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R_d & 0_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \tag{2.10}$$

Although the Euler angles are intuitive, by using quaternions smoother rotations between two orientations are obtained, additionally, the Euler angles can suffer when two axes align leading to singularities. And Quaternions are computationally efficient. In many applications, quaternions are preferred for representing and manipulating 3D rotations, especially in computer

graphics, robotics, and simulations, due to their advantages in terms of interpolation, avoiding singularities, and efficiency. However, Euler angles can still be useful in situations where human interpretability or simplicity is more important, as long as their limitations are carefully managed.

2.1.3 Denavit-Hartenberg conventions

There are multiple methods for characterizing a kinematic chain through sequential rotations and translations with few parameters. Within the academic literature, two predominant conventions have emerged: the modified Denavit-Hartenberg (mDH) convention and the standard Denavit-Hartenberg (DH) convention (Craig, 2005). These conventions primarily differ in the sequence of operations and the method used to select frames for parameter computation. For the standard DH, the homogeneous transformation between two successive frames is given by:

$$ {}^{i-1}_{i}T = T_z(\theta_i)T_D(0,0,d_i)T_D(a_i,0,0)T_x(\alpha_i) \tag{2.11} $$

where

- θ_i is the angle between X_{i-1} and X_i around Z_{i-1}
- d_i is the distance from X_{i-1} to X_i along Z_{i-1}
- a_i is the distance from Z_{i-1} to Z_i along X_i
- α_i is the angle between Z_{i-1} and Z_i around X_i

where X_*, Y_* and Z_* are the axes for frames i and $i-1$ as shown in Fig. 2.2. And Equation (2.11) results in:

$$ {}^{i-1}_{i}T = \begin{bmatrix} c(\theta_i) & -c(\alpha_i)s(\theta_i) & s(\alpha_i)s(\theta_i) & a_i c(\theta_i) \\ s(\theta_i) & c(\alpha_i)c(\theta_i) & -s(\alpha_i)c(\theta_i) & a_i s(\theta_i) \\ 0 & s(\alpha_i) & c(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.12} $$

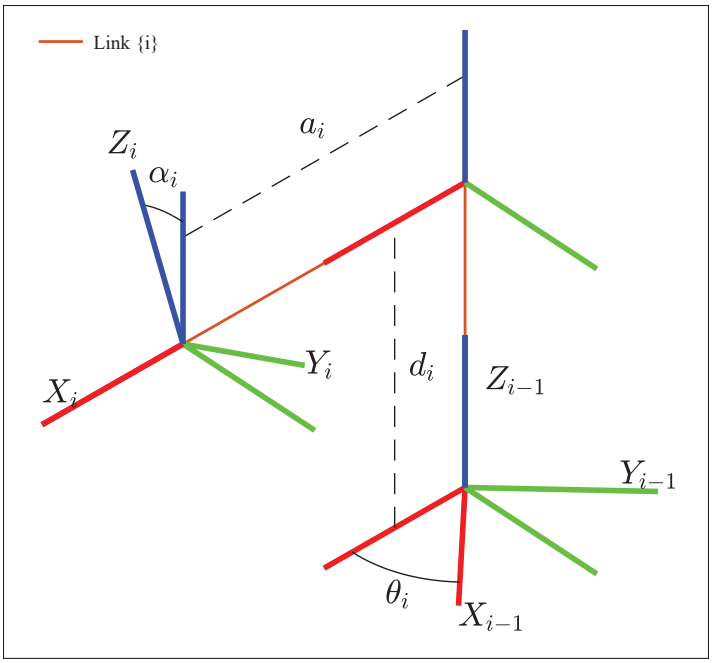

Figure 2.2 Standard DH parameters with i and $i-1$ frames

Since the design of ETS-MARSE is grounded in the mDH convention, we will consistently apply this convention in the subsequent sections. Therefore, for the mDH convention, the homogeneous transformation between two consecutive frames can be expressed as follows:

$$ {}^{i-1}_{i}T = T_x(\alpha_{i-1})T_D(a_{i-1},0,0)T_z(\theta_i)T_D(0,0,d_i) \tag{2.13} $$

where

- α_{i-1} is the angle between Z_{i-1} and Z_i around X_{i-1}
- a_{i-1} is the distance from Z_{i-1} to Z_i along X_{i-1}
- θ_i is the angle between X_{i-1} and X_i around Z_i
- d_i is the distance from X_{i-1} to X_i along Z_i

where the i and $i-1$ frames can be observed in Fig. 2.3. By expanding Equation (2.13) the

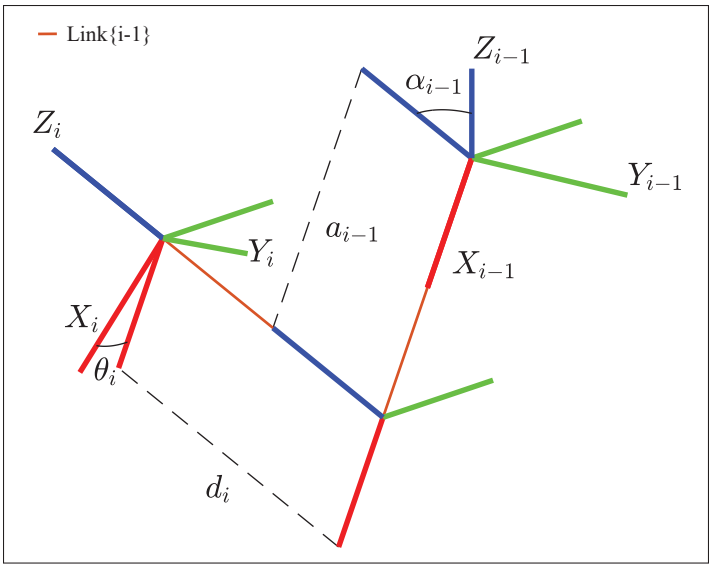

Figure 2.3 Modified DH parameters with i and $i-1$ frames

following result is obtained.

$$
{}^{i-1}_{i}T = \begin{bmatrix} c(\theta_i) & -s(\theta_i) & 0 & a_{i-1} \\ c(\alpha_{i-1})s(\theta_i) & c(\alpha_{i-1})c(\theta_i) & -s(\alpha_{i-1}) & -d_i s(\alpha_{i-1}) \\ s(\alpha_{i-1})s(\theta_i) & s(\alpha_{i-1})c(\theta_i) & c(\alpha_{i-1}) & d_i c(\alpha_{i-1}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.14}
$$

Readers can confirm that the following equations are equivalent to Equation (2.13) and lead to the same result as Equation (2.14). This equivalence arises from the specific nature of the matrices involved in this multiplication.

$$
\begin{aligned}
{}^{i-1}_{i}T &= T_x(\alpha_{i-1})T_D(a_{i-1},0,0)T_D(0,0,d_i)T_z(\theta_i) \\
&= T_D(a_{i-1},0,0)T_x(\alpha_{i-1})T_D(0,0,d_i)T_z(\theta_i) \\
&= T_D(a_{i-1},0,0)T_x(\alpha_{i-1})T_z(\theta_i)T_D(0,0,d_i)
\end{aligned} \tag{2.15}
$$

In order to characterize the kinematic chain of ETS-MARSE, we employ a set of Modified Denavit-Hartenberg (mDH) parameters, as illustrated in Table 2.1. The mDH parameters include

the joints: SIER (Shoulder Internal/External Rotation), SFE (Shoulder Flexion/Extension), EFE (Elbow Flexion/Extension), FPS (Forearm Pronation/Supination), WRUD (Wrist Radial/Ulnar Deviation), and WFE (Wrist Flexion/Extension). The constants d_e and d_w represent the lengths

Table 2.1 Modified DH parameters Taken from Rahman (2015)

Frame	α_{i-1}	a_{i-1}	d_i	θ_i	**Joint**
0	0	0	0	0	Base frame
1	0	0	d_s	q_1	SIER
2	$-\frac{\pi}{2}$	0	0	q_2	SFE
3	$\frac{\pi}{2}$	0	d_e	q_3	SIER
4	$-\frac{\pi}{2}$	0	0	q_4	EFE
5	$\frac{\pi}{2}$	0	d_w	q_5	FPS
6	$-\frac{\pi}{2}$	0	0	$q_6 - \frac{\pi}{2}$	WRUD
7	$-\frac{\pi}{2}$	0	0	q_7	WFE

of the upper arm and forearm, respectively. The constant d_s denotes the distance between the base frame and the shoulder frame. By utilizing the mDH parameters provided in Table 2.1, the

individual homogeneous transformations for each joint results in:

$$
\begin{aligned}
{}^{0}_{1}T &= \begin{bmatrix} c(q_1) & -s(q_1) & 0 & 0 \\ s(q_1) & c(q_1) & 0 & 0 \\ 0 & 0 & 1 & d_s \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{1}_{2}T = \begin{bmatrix} c(q_2) & -s(q_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s(q_2) & -c(q_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^{2}_{3}T &= \begin{bmatrix} c(q_3) & -s(q_3) & 0 & 0 \\ 0 & 0 & -1 & -d_e \\ s(q_3) & c(q_3) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{3}_{4}T = \begin{bmatrix} c(q_4) & -s(q_4) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s(q_4) & -c(q_4) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^{4}_{5}T &= \begin{bmatrix} c(q_5) & -s(q_5) & 0 & 0 \\ 0 & 0 & -1 & -d_w \\ s(q_5) & c(q_5) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{5}_{6}T = \begin{bmatrix} s(q_6) & c(q_6) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ c(q_6) & -s(q_6) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^{6}_{7}T &= \begin{bmatrix} c(q_7) & -s(q_7) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s(q_7) & -c(q_7) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{2.16}
$$

The homogeneous transformation from the i-th frame to the k-th frame is determined through a sequence of matrix multiplications, as illustrated below:

$$
{}^{i}_{k}T = {}^{i}_{i+1}T \cdots {}^{k-1}_{k}T \tag{2.17}
$$

The forward kinematics results in Equation (2.18), where the terms s_q, e_q, and w_q represent the positions of the robot's shoulder, elbow, and wrist, respectively, given a set of joint angle values.

$$\begin{bmatrix} s_q \\ 1 \end{bmatrix} = {}^0_1T \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} e_q \\ 1 \end{bmatrix} = {}^0_4T \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} w_q \\ 1 \end{bmatrix} = {}^0_7T \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{2.18}$$

where

$$s_q = \begin{bmatrix} 0 & 0 & d_s \end{bmatrix}^T \tag{2.19}$$

$$e_q = \begin{bmatrix} d_e c_1 s_2 & d_e s_1 s_2 & d_s + d_e c_2 \end{bmatrix}^T \tag{2.20}$$

$$w_q = \begin{bmatrix} -d_w(s_4(-c_1c_2c_3 + s_1s_3) - c_1s_2c_4) + d_e c_1 s_2 \\ d_w(s_4(c_1s_3 + s_1c_2c_3) + s_1s_2c_4) + d_e s_1 s_2 \\ d_s + d_w(c_2c_4 - s_2c_3s_4) + d_e c_2 \end{bmatrix} \tag{2.21}$$

The frames ${}^0_1T$, ${}^0_2T, \ldots,$ ${}^0_7T$, obtained using Equation (2.17) for ETS-MARSE, can be visualized in Matlab as shown in Fig. 2.4. To achieve this, modify the code used to generate Fig. 2.1, with the joint angles set to $q_4 = \pi/2$ and the rest of the joints set to zero. It's worth noting that plotting these frames in Matlab can provide valuable learning opportunities for beginners. While plotting keep in mind that the z-axis and y-axis appear reversed due to the orientation of the base frame.

2.2 Inverse kinematics

The inverse kinematics problem involves determining the joint angles within a kinematic chain that lead to a desired frame, encompassing the desired position and orientation, as depicted below:

$$T_d = \begin{bmatrix} R_d & w_d \\ 0_{1\times3} & 1 \end{bmatrix} \tag{2.22}$$

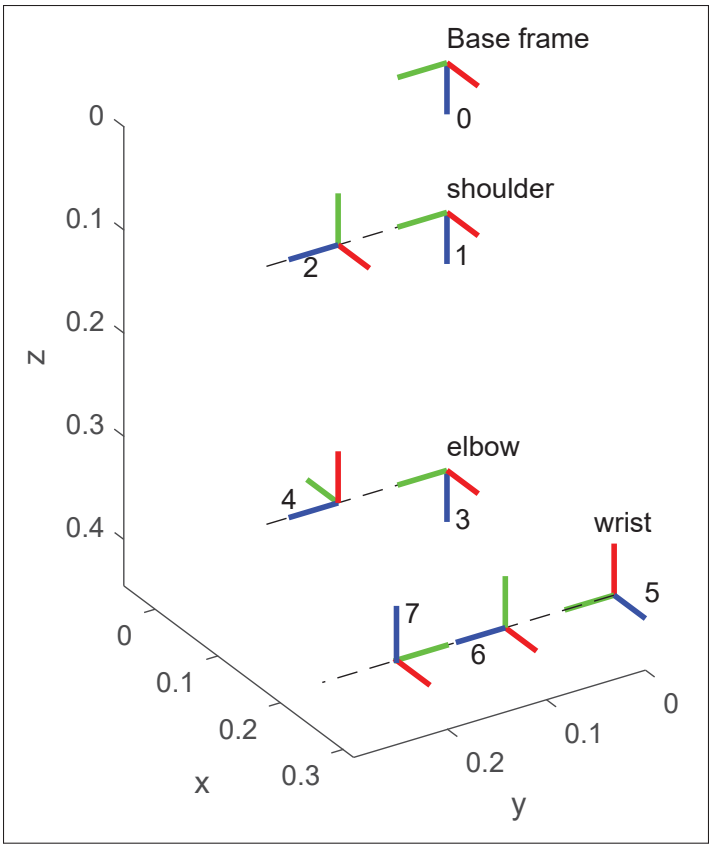

Figure 2.4 ETS-MARSE frames (red for x-axis, green for y-axis and blue for z-axis)

where $w_d = [w_{xd},\ w_{yd},\ w_{zd}]^T$, and R_d defined in Equation (2.4) represent the desired position and orientation, respectively. The ETS-MARSE is a redundant robot with 7 degrees of freedom, enabling it to reach the same end-effector position and orientation using different joint configurations. By separating the inverse kinematics into inverse position and orientation, this problem can be effectively solved in two stages. The first stage involves finding the first four joint values to attain the desired end-effector position, typically accomplished using the swivel angle approach. The second stage focuses on determining the last three joint values to achieve the desired end-effector orientation.

2.2.1 Swivel angle approach

The swivel angle approach was developed to handle the challenge of parameterizing the infinite solutions when mapping four joints to a 3D position in Cartesian space, as described in (Tolani & Badler, 1996). The core concept behind the swivel angle approach involves the use of a rotating frame with axes labeled as x_{sw}, y_{sw}, and z_{sw} (as shown in Fig. 2.5) to establish a suitable elbow position in an anthropomorphic manner, determined by a swivel angle ϕ. The rotating

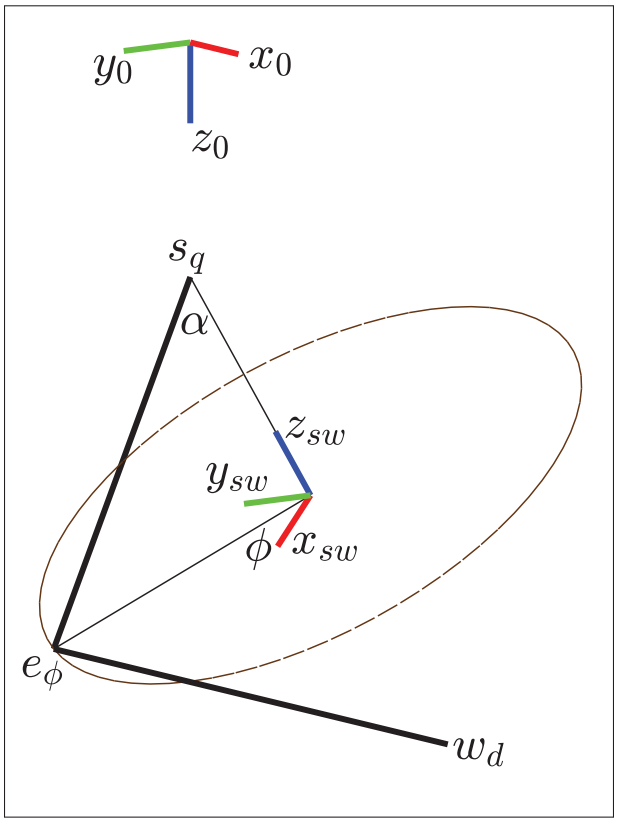

Figure 2.5 Swivel angle frame

frame is computed based on the robot's shoulder position s_q and the desired wrist position w_d. It's important to emphasize that the shoulder position s_q remains fixed, as illustrated in Equation (2.19). Consequently, the configuration of the rotating frame is solely dependent on the desired end-effector position w_d. The vector z_{sw}, which serves as the rotation axis, can be computed as

follows:

$$z_{sw} = \frac{s_q - w_d}{||s_q - w_d||} \tag{2.23}$$

The vector x_{sw} is defined as:

$$x_{sw} = \frac{-z_a + (z_a \cdot z_{sw})z_{sw}}{|| - z_a + (z_a \cdot z_{sw})z_{sw}||} \tag{2.24}$$

Here, the "$\cdot$" symbol represents the dot product, and $z_a = [0,\ 0,\ -1]^T$ is an arbitrary vector employed to derive the x_{sw} vector and to define the zero position of the swivel angle. To finalize the construction of the rotating frame, the y_{sw} vector is determined using the cross product, as shown below:

$$y_{sw} = z_{sw} \times x_{sw} \tag{2.25}$$

Please note when vectors z_{sw} and x_{sw} are aligned the cross product will result in an invalid rotation frame. Then, after obtaining the rotating frame, it is necessary to determine its center c_{sw} and the angle α. The angle α can be found using the following equation:

$$\alpha = a\cos\left(\frac{d_w^2 - d_e^2 - ||w_d - s_q||^2}{-2d_e||w_d - s_q||}\right) \tag{2.26}$$

and the center of the rotating frame is computed as:

$$c_{sw} = -c(\alpha)d_e z_{sw} + s_q \tag{2.27}$$

Hence, the elbow position in terms of the swivel angle ϕ can be calculated as follows:

$$e_\phi = c_{sw} + d_e s(\alpha)(c(\phi)x_{sw} + s(\phi)y_{sw}) \tag{2.28}$$

Note that e_ϕ can mathematically reach any position in the circular representation displayed in Fig. 2.5. In the same figure, a positive choice for the swivel angle ϕ has been made, leading to a natural placement of the right elbow on the right side of the body. However, it's worth noting that depending on the method used to calculate z_{sw}, achieving the aforementioned natural elbow

configuration may require a negative value for ϕ. For example, a negative ϕ is required if the vectors z_{sw} and c_{sw} are instead computed as:

$$\begin{aligned} z_{sw} &= \frac{w_d - s_q}{||w_d - s_q||} \\ c_{sw} &= c(\alpha) d_e z_{sw} + s_q \end{aligned} \tag{2.29}$$

Readers can discern the distinctions between the equations used to compute z_{sw} and c_{sw} by making a visual comparison between Fig. 2.5 and Fig. 2.6.

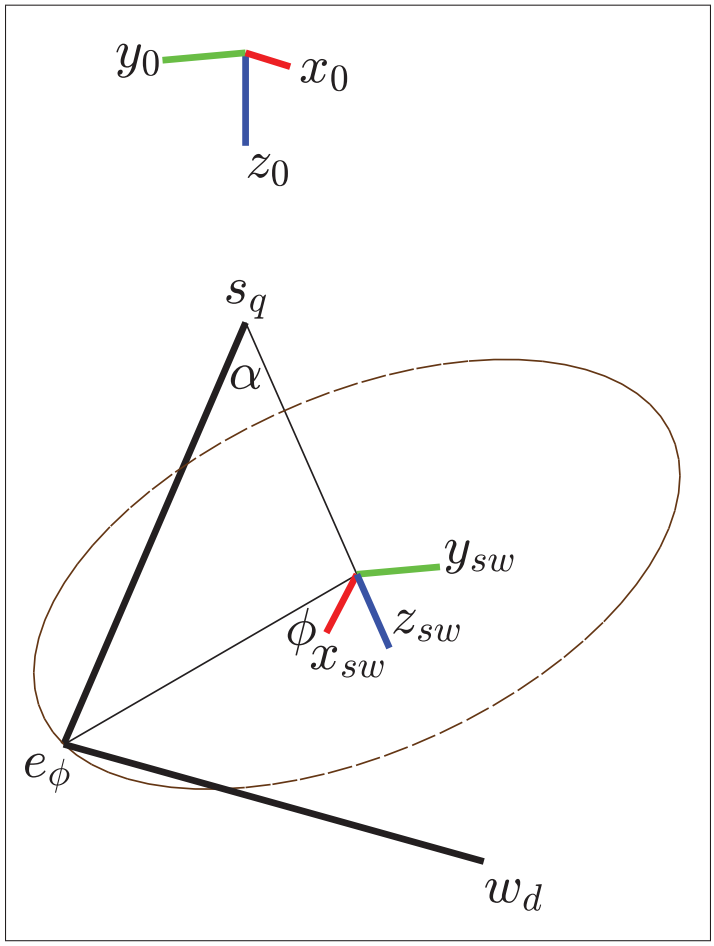

Figure 2.6 Swivel angle frame (negative angle)

2.2.2 Swivel angle estimation

Selecting an appropriate swivel angle is a challenging task, and it can vary depending on the specific situation. The human brain intuitively selects a swivel angle, and it's often difficult to determine if there's a unique swivel angle for every desired wrist position and orientation. Consequently, various techniques have been proposed to estimate a suitable swivel angle. For example, the Newton-Raphson method (Craig, 2005), aims to find the shortest path between two given targets while minimizing energy. Alternatively, some methods optimize mathematical functions, such as discomfort indices (Zhao *et al.*, 2014). More recently, machine learning techniques have been applied for this purpose (Lauretti *et al.*, 2018). A comparison with measured data was conducted (Bedolla-Martinez *et al.*, 2023) and presented in Chapter 3 of this thesis, and some results are presented in Fig. 2.7.

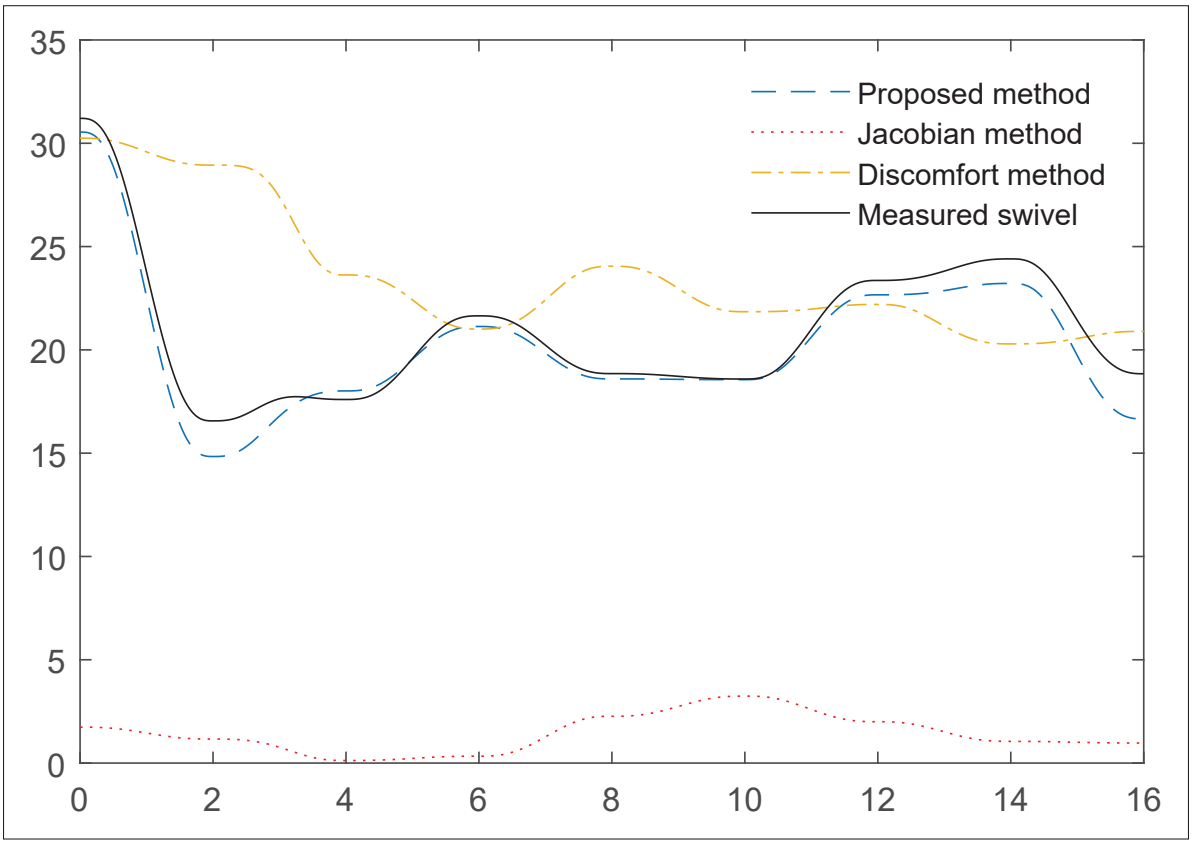

Figure 2.7 Swivel angle estimation

2.2.3 Inverse position

To compute the first four joint values of ETS-MARSE in order to attain a specific position, you need both the desired wrist position w_d and elbow position $e_d = [e_{xd},\ e_{yd},\ e_{zd}]^T$. It's worth mentioning that e_d can be derived from e_ϕ given in (2.28) or directly measured using a motion capture system. Consequently, the values for joints 1 and 2 are determined as follows:

$$\begin{aligned} q_1 &= atan2(e_{yd}, e_{xd}) \\ q_2 &= atan2(\pm\sqrt{e_{xd}^2 + e_{yd}^2}, e_{zd} - d_s) \end{aligned} \tag{2.30}$$

The value for joint 3 can also be determined analytically. By substituting $w_d = [w_{xd},\ w_{yd},\ w_{zd}]^T$ into Equation (2.21) and then multiplying both sides by $({}^0_1T\,{}^1_2T)^{-1}$, the following equation is obtained:

$$\begin{bmatrix} d_w c(q_3)s(q_4) \\ -d_e - d_w c(q_4) \\ d_w s(q_3)s(q_4) \\ 1 \end{bmatrix} = \begin{bmatrix} w_{xd}c(q_1)c(q_2) + w_{yd}s(q_1)c(q_2) - w_{zd}s(q_2) + d_s s(q_2) \\ -w_{xd}c(q_1)s(q_2) - w_{yd}s(q_1)s(q_2) - w_{zd}c(q_2) + d_s c(q_2) \\ -w_{xd}s(q_1) + w_{yd}c(q_1) \\ 1 \end{bmatrix} \tag{2.31}$$

Then

$$q_3 = atan2(-w_{xd}s(q_1) + w_{yd}c(q_1), w_{xd}c(q_1)c(q_2) + w_{yd}s(q_1)c(q_2) - w_{zd}s(q_2) + d_s s(q_2)) \tag{2.32}$$

The value of joint 4 does not depend on the swivel angle and it is computed by:

$$q_4 = \pi \pm a\cos\left(\frac{d_w^2 + d_e^2 - ||w_d - s_q||^2}{2d_e d_w}\right) \tag{2.33}$$

2.2.4 Inverse orientation

Using the desired orientation matrix R_d from Equation 2.4, the joint angles for the last three joints can be computed. It's important to highlight that the matrix ${}^4_7T$ encompasses the orientation of

the end-effector resulting in:

$$ {}^{4}_{7}T = {}^{4}_{5}T\,{}^{5}_{6}T\,{}^{6}_{7}T = \begin{bmatrix} s_5 s_7 + c_5 s_6 c_7 & s_5 c_7 - c_5 s_6 s_7 & c_5 c_6 & 0 \\ -c_6 c_7 & c_6 s_7 & s_6 & -d_w \\ -c_5 s_7 + s_5 s_6 c_7 & -c_5 c_7 - s_5 s_6 s_7 & s_5 c_6 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.34} $$

Using the matrices R_d and ${}^{4}_{7}T$ the following relation is built

$$ {}^{4}_{7}T = {}^{0}_{4}T^{-1} \begin{bmatrix} R_d & 0_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \tag{2.35} $$

Developing the right-hand part of (2.35), one obtains:

$$ {}^{4}_{7}T = \begin{bmatrix} \underline{o}_{11} & \underline{o}_{12} & \underline{o}_{13} & \underline{o}_{14} \\ \underline{o}_{21} & \underline{o}_{22} & \underline{o}_{23} & \underline{o}_{24} \\ \underline{o}_{31} & \underline{o}_{32} & \underline{o}_{33} & \underline{o}_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.36} $$

Considering (2.34) and (2.36), the solution for q_6 is:

$$ q_6 = atan2\left(\underline{o}_{23}, c_6\right) \tag{2.37} $$

where $c_6 = \pm\sqrt{\underline{o}_{13}^2 + \underline{o}_{33}^2}$. The solution for q_5 is:

$$ q_5 = atan2\left(\underline{o}_{33}, \underline{o}_{13}\right) \tag{2.38} $$

The solution of q_7 is:

$$ q_7 = atan2\left(\underline{o}_{22}, -\underline{o}_{21}\right) \tag{2.39} $$

2.3 Trajectory generation

Two primary categories of trajectories exist Joint-space and Task-space trajectories. This section will elucidate both Joint-space and Task-space trajectories (Biagiotti & Melchiorri, 2008).

2.3.1 Joint-space polynomial trajectories

While polynomial trajectories are utilized for both joint-space and task-space, this section focuses on joint-space polynomial trajectories, with the implementation for task-space being relatively straightforward. Two types of polynomial trajectories are commonly used: third-order and fifth-order. The third-order trajectory enables the specification of initial position and velocity (q_{ini}, $\dot{q}_{ini}$), final position and velocity (q_{fin}, $\dot{q}_{fin}$), and initial and final time (t_{ini}, t_{fin}).

On the other hand, the fifth-order trajectory offers the same parameters as the third-order type and, in addition, allows for the definition of initial and final accelerations ($\ddot{q}_{ini}$ and $\ddot{q}_{fin}$), resulting in smoother trajectories (see the comparison on Fig. 2.8). The polynomial used for the third-order trajectory is as follows:

$$\begin{aligned} q(t) &= \epsilon_1 + \epsilon_2 t + \epsilon_3 t^2 + \epsilon_4 t^3 \\ \dot{q}(t) &= \epsilon_2 + 2\epsilon_3 t + 3\epsilon_4 t^2 \\ \ddot{q}(t) &= 2\epsilon_3 + 6\epsilon_4 t \end{aligned} \tag{2.40}$$

where ϵ_i are coefficients that depend on the previously described initial and final conditions.

$$\begin{aligned} \epsilon_1 &= q_{ini} \\ \epsilon_2 &= \dot{q}_{ini} \\ \epsilon_3 &= \frac{-3(q_{ini} - q_{fin}) - (2\dot{q}_{ini} + \dot{q}_{fin})(t_{fin} - t_{ini})}{(t_{fin} - t_{ini})^2} \\ \epsilon_4 &= \frac{2(q_{ini} - q_{fin}) + (\dot{q}_{ini} + \dot{q}_{fin})(t_{fin} - t_{ini})}{(t_{fin} - t_{ini})^3} \end{aligned} \tag{2.41}$$

On the other hand, the fifth-order polynomial is defined as:

$$\begin{aligned} q(t) &= \epsilon_1 + \epsilon_2 t + \epsilon_3 t^2 + \epsilon_4 t^3 + \epsilon_5 t^4 + \epsilon_6 t^5 \\ \dot{q}(t) &= \epsilon_2 + 2\epsilon_3 t + 3\epsilon_4 t^2 + 4\epsilon_5 t^3 + 5\epsilon_6 t^4 \\ \ddot{q}(t) &= 2\epsilon_3 + 6\epsilon_4 t + 12\epsilon_5 t^2 + 20\epsilon_6 t^3 \end{aligned} \tag{2.42}$$

where the coefficients are in function of the desired initial and final conditions.

$$\begin{aligned} \epsilon_1 &= q_{ini} \\ \epsilon_2 &= \dot{q}_{ini} \\ \epsilon_3 &= \frac{1}{2}\ddot{q}_{ini} \\ \epsilon_4 &= \frac{1}{2(t_{fin} - t_{ini})^3}\left[20(q_{fin} - q_{ini}) - (8\dot{q}_{fin} + 12\dot{q}_{ini})(t_{fin} - t_{ini}) - (3\ddot{q}_{fin} - \ddot{q}_{ini})(t_{fin} - t_{ini})^2\right] \\ \epsilon_5 &= \frac{1}{2(t_{fin} - t_{ini})^4}\left[30(q_{ini} - q_{fin}) + (14\dot{q}_{fin} + 16\dot{q}_{ini})(t_{fin} - t_{inif}) + (3\ddot{q}_{fin} - 2\ddot{q}_{ini})(t_{fin} - t_{ini})^2\right] \\ \epsilon_6 &= \frac{1}{2(t_{fin} - t_{ini})^5}\left[12(q_{fin} - q_{ini}) - 6(\dot{q}_{fin} + \dot{q}_{ini})(t_{fin} - t_{inif}) - (\ddot{q}_{fin} - \ddot{q}_{ini})(t_{fin} - t_{ini})^2\right] \end{aligned} \tag{2.43}$$

A comparison of the third-order and fifth-order trajectories is presented in Fig. 2.8 where a smoother acceleration is achieved by the fifth-order trajectory type. The trajectory is composed of two segments, for Segment 1, the initial position is $q_{ini} = 0$, the final position is $q_{fin} = 2\pi$, and the initial and final times are $t_{ini} = 0\ sec$ and $t_{fin} = 2\ sec$, respectively. Initial and final velocities and accelerations are constrained to be zero for this segment. In Segment 2, the initial conditions are equal to the final conditions of the previous segment. The final position is $q_{fin} = 0$, and the final time is $t_{fin} = 4\ sec$, with final velocities and accelerations similarly held at zero.

2.3.2 Task-space trajectories

A task-space trajectory includes a series of end-effector positions and orientations. Consider a parametric function that exhibits an infinity-shaped trajectory, as illustrated in Fig. 2.9. With

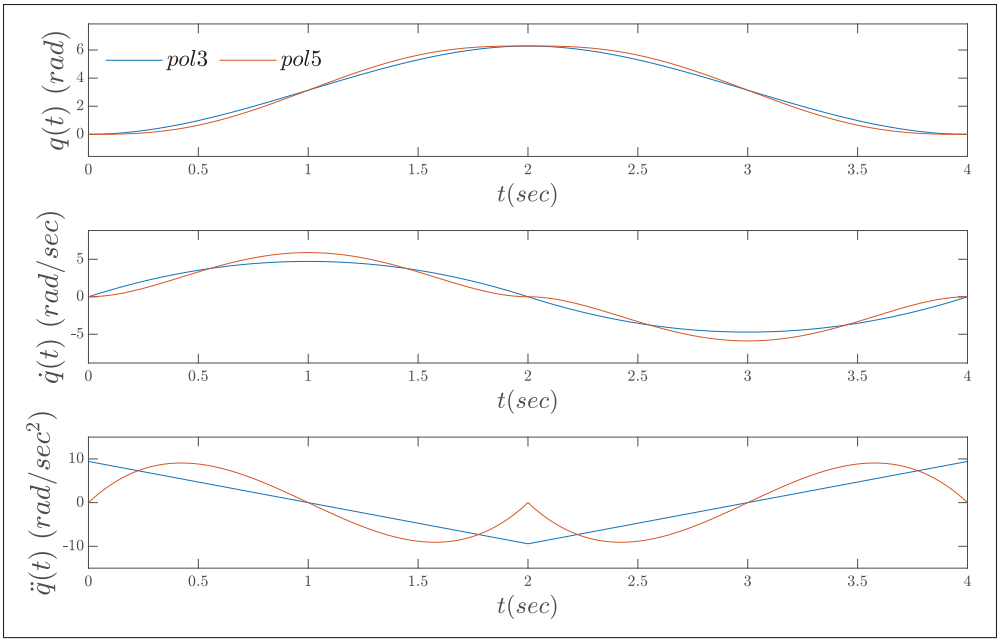

Figure 2.8 Third-order and fifth-order trajectories comparison

T_{sim} as the trajectory time, ϕ as a fixed swivel angle for simplicity. The Euler angles α_d, β_d, and γ_d define the desired orientation matrix (fixed for simplicity). The desired end-effector position defined by the parametric function $w_d = [w_{xd}, w_{yd}, w_{zd}]$ from time 0 to T_{sim}. The corresponding joint-space solution is presented in Fig. 2.10.

$$
\begin{aligned}
&T_{sim} = 8\ sec,\ t = 0 \rightarrow T_{sim},\ \phi = 45^\circ \\
&\alpha_d = 180^\circ,\ \beta_d = 5^\circ,\ \gamma_d = 5^\circ \\
&w_{xd} = 0.267 + \frac{1}{8}c(\pi/5)(1 - c(2\pi t/T_{sim})),\ w_{yd} = \frac{1}{8}s(4\pi t/T_{sim}) \\
&w_{zd} = 0.25 + \frac{1}{8}s(\pi/5)(1 + c(2\pi t/T_{sim}))
\end{aligned}
\tag{2.44}
$$

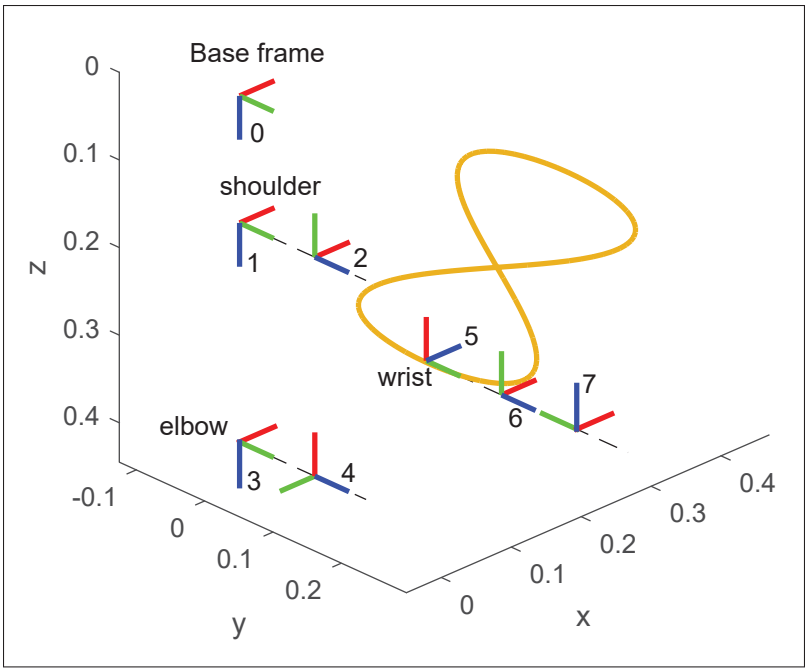

Figure 2.9 Infinity shape trajectory

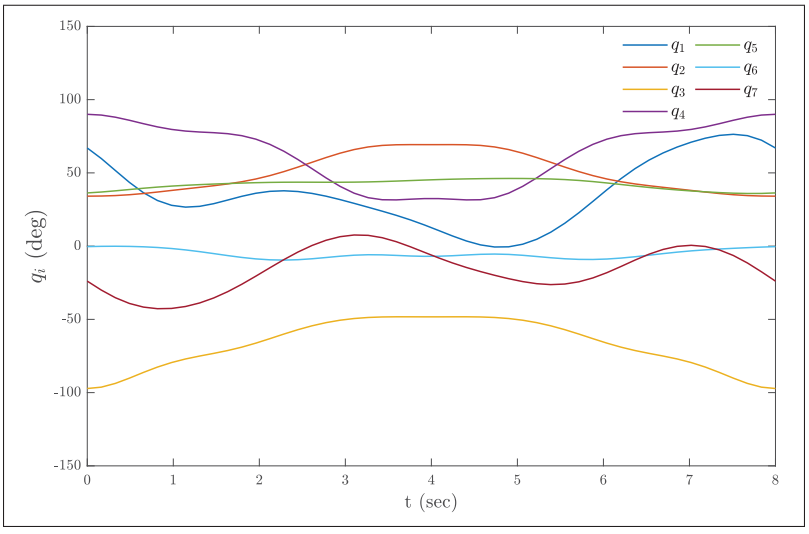

Figure 2.10 Joint-space trajectory

2.4 Feedback controller design

In the typical process of designing a feedback controller, the initial steps involve a feedback linearization procedure aimed at compensating for known system dynamics. Subsequently, when dealing with this uncertain linear system, the development of a virtual control strategy becomes essential to address any remaining unknown dynamics. Many feedback controllers necessitate the use of an equivalent control, which characterizes the system's behavior during the steady-state phase, and a reaching law, which governs the system's behavior during the transient phase.

2.4.1 Feedback linearization

The dynamics of the studied exoskeleton are given in the following standard form:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau + \tau_u \tag{2.45}$$

where $\theta \in \mathbb{R}^n$, $\dot{\theta} \in \mathbb{R}^n$ and $\ddot{\theta} \in \mathbb{R}^n$ are the joint angular position, velocity, and acceleration respectively, $M(\theta) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $C(\theta, \dot{\theta}) \in \mathbb{R}^{n\times n}$ is the Coriolis and centrifugal matrix, $G(\theta) \in \mathbb{R}^n$ is the gravitational force vector, τ is the torque applied by the motors and τ_u is the unknown dynamics e.g. external disturbances, model uncertainties and human-robot interaction forces. Notice that n is the number of DoF and is equal to 7 in our case.

By using the known dynamics of (2.45), a nominal control law is obtained as follows:

$$\tau = M(\theta)\nu + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) \tag{2.46}$$

where ν is a virtual control input that will be designed in the subsequent part. Substituting (2.46) into (2.45) yields:

$$\ddot{\theta} = \nu + w_u \tag{2.47}$$

where $w_u = M^{-1}(\theta)\tau_u$.

2.4.2 Virtual control

The virtual control serves the purpose of guiding the system from the transient phase to the steady-state phase, taking into account the tracking error. It's essential for the uncertain dynamics denoted as w_u to be effectively influenced and dominated by the virtual control ν throughout the control process to ensure convergence to a small region or ultimately to zero at a specific time. Virtual control typically encompasses equivalent and reaching control laws.

$$\nu = \nu_{eq} + \nu_{rea} \tag{2.48}$$

Let's consider an SMC approach with the following sliding surface:

$$s = \dot{e} + f(e) \tag{2.49}$$

where $e = \theta - \theta_d$, with θ_d as the desired reference. And $f(e)$ is in function of the error e. Taking the derivative of the sliding surface results in:

$$\dot{s} = \ddot{e} + \dot{f}(e)\dot{e} \tag{2.50}$$

Then substituting the nominal parts of (2.47) and (2.48) in the Equation (2.50) yields:

$$\begin{aligned} \dot{s} &= \ddot{\theta} - \ddot{\theta}_d + \dot{f}(e)\dot{e} \\ \dot{s} &= \nu - \ddot{\theta}_d + \dot{f}(e)\dot{e} \\ \dot{s} &= \nu_{eq} + \nu_{rea} - \ddot{\theta}_d + \dot{f}(e)\dot{e} \\ \dot{s} - \nu_{rea} &= \nu_{eq} - \ddot{\theta}_d + \dot{f}(e)\dot{e} \end{aligned} \tag{2.51}$$

By setting $\dot{s} = \nu_{rea}$ in the previous equation, the equivalent control can be computed as:

$$\nu_{eq} = \ddot{\theta}_d - \dot{f}(e)\dot{e} \tag{2.52}$$

It's worth emphasizing that, in most instances, the equivalent control primarily influences the system's dynamics during the steady-state phase, while the reaching law is selected to govern the behavior during the reaching phase. Consequently, when deciding on the reaching law ν_{rea}, it's crucial to take into account the maximum value of the unknown dynamic w_u.

2.4.3 Error dynamics

Once the equivalent control is properly defined by selecting an appropriate $f(e)$ function and a reaching law ν_{rea}, the system's error dynamics can be determined by substituting the virtual control ν from Equation (2.48) into the derivative of the sliding surface defined in Equation (2.50).

$$\begin{aligned} \dot{s} &= \ddot{e} + \dot{f}(e)\dot{e} \\ \dot{s} &= \nu_{rea} + \ddot{\theta}_d - \dot{f}(e)\dot{e} - \ddot{\theta}_d + \dot{f}(e)\dot{e} + w_u \\ \dot{s} &= \nu_{rea} + w_u \end{aligned} \tag{2.53}$$

The stability analysis hinges on the choice of the reaching law ν_{rea}, and in most cases, it is sufficient to assume that $||w_u|| < ||\nu_{rea}||$ for all time. However, certain methods like the super-twisting or backstepping approach, as described in (Kali *et al.*, 2018), demand a distinct mathematical procedure for stability analysis.

2.5 Prescribed performance control

Prescribed performance control is a methodology that allows explicitly specifying and achieving the desired control performance in a systematic and structured manner, making it a valuable tool in various control applications. The specifications can include requirements for tracking a reference signal, achieving a specific settling time, and minimizing overshoot.

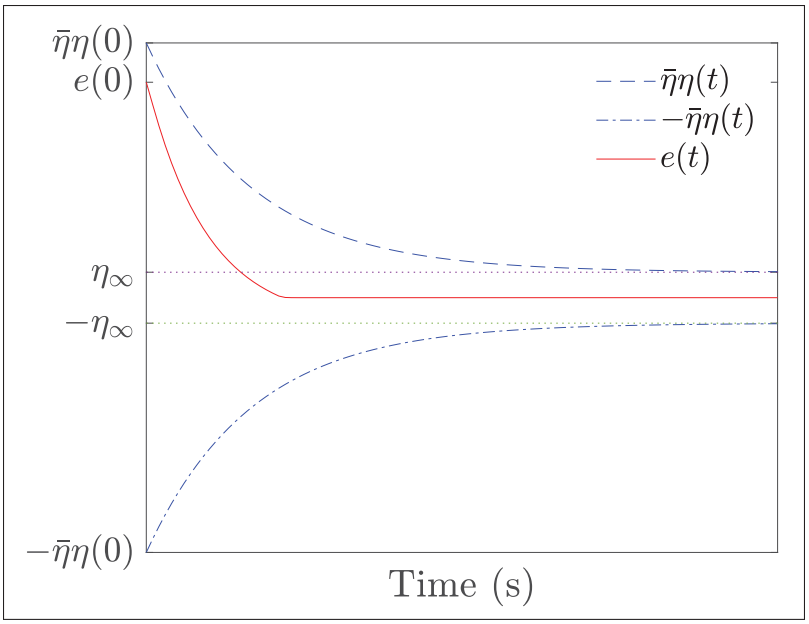

Figure 2.11 Constraints defined by the prescribed performance function

2.5.1 Prescribed performance development

One of the commonly used prescribed performance functions is the "exponential" function $\eta(t)$ (Bu, 2023) with its derivatives defined as follows:

$$\begin{aligned} \eta(t) &= \eta_\infty + (\eta_0 - \eta_\infty)\exp(-lt) \\ \dot{\eta}(t) &= -l(\eta_0 - \eta_\infty)\exp(-lt) \\ \ddot{\eta}(t) &= l^2(\eta_0 - \eta_\infty)\exp(-lt) \end{aligned} \tag{2.54}$$

where $\eta(t)$ goes from η_0 to η_∞ in $t = 0 \rightarrow \infty$ as shown in Fig. 2.11. Let us define the i^{th} constrained error as follows:

$$-\bar{\eta}\eta(t) < e_i = \theta_i - \theta_{id} < \bar{\eta}\eta(t) \tag{2.55}$$

where θ_{id} is the i^{th} desired reference. With $\bar{\eta}$ as constraint modifier. The constrained error is transformed into an unconstrained one $z = [z_1, .., z_n]^T$ with first and second derivatives defined as:

$$\begin{aligned} z_i &= \frac{1}{2}\ln\left(\frac{\frac{e_i}{\eta(t)} + \bar{\eta}}{\bar{\eta} - \frac{e_i}{\eta(t)}}\right) \\ \dot{z}_i &= r_i H_{1i} \\ \ddot{z}_i &= \dot{r}_i H_{1i} + r_i \left(\ddot{e}_i + H_{2i}\right) \end{aligned} \tag{2.56}$$

where H_{1i} and H_{2i} are defined below:

$$\begin{aligned} H_{1i} &= \dot{e}_i - \frac{e_i \dot{\eta}(t)}{\eta(t)} \\ H_{2i} &= \frac{e_i \dot{\eta}(t)^2}{\eta(t)^2} - \frac{e_i \ddot{\eta}(t)}{\eta(t)} - \frac{\dot{e}_i \dot{\eta}(t)}{\eta(t)} \end{aligned} \tag{2.57}$$

And $r \in \mathbb{R}^{n \times n}$ and its derivative are given by:

$$\begin{aligned} r_i &= -\frac{\bar{\eta}\eta(t)}{(e_i + \bar{\eta}\eta(t))(e_i - \bar{\eta}\eta(t))} \\ \dot{r}_i &= \frac{-H_{3i}\bar{\eta}}{(e_i + \bar{\eta}\eta(t))^2 (e_i - \bar{\eta}\eta(t))^2} \end{aligned} \tag{2.58}$$

with

$$H_{3i} = \left(2\bar{\eta}\eta(t)^2 + e_i^2\right)\dot{\eta}(t) - 2\eta(t) e_i \dot{e}_i \tag{2.59}$$

2.5.2 Virtual control

Now after defining z_i, $\dot{z}_i$, and $\ddot{z}_i$, the prescribed performance approach can be integrated with another controller. To that end let us consider a sliding mode control approach with a sliding surface in function of the unconstrained error z as follows:

$$s = \dot{z} + D(z) \tag{2.60}$$

where the derivative of $D(z)$ will later serve as a part of the equivalent control. The derivative of Equation (2.60) yields:

$$\dot{s} = \ddot{z} + \dot{D}(z)\dot{z} \tag{2.61}$$

By substituting $\ddot{z}$, the nominal parts of Equations (2.47) and (2.48) in Equation (2.61) results in:

$$\begin{aligned} \dot{s} &= \dot{r}H_1 + r\left(\ddot{e} + H_2\right) + \dot{D}(z)\dot{z} \\ \dot{s} &= \dot{r}H_1 + r\left(\ddot{\theta} - \ddot{\theta}_d + H_2\right) + \dot{D}(z)\dot{z} \\ \dot{s} &= \dot{r}H_1 + r\left(\nu_{eq} + \nu_{rea} - \ddot{\theta}_d + H_2\right) + \dot{D}(z)\dot{z} \\ \dot{s} - r\nu_{rea} &= \dot{r}H_1 + r\left(\nu_{eq} - \ddot{\theta}_d + H_2\right) + \dot{D}(z)\dot{z} \end{aligned} \tag{2.62}$$

Considering that $\dot{s} = r\nu_{rea}$, the equivalent control ν_{eq} yields:

$$\nu_{eq} = \ddot{\theta}_d - H_2 - r^{-1}\left(\dot{r}H_1 + \dot{D}(z)\dot{z}\right) \tag{2.63}$$

Finally, the virtual control can be written as follows:

$$\nu = \nu_{eq} + r\nu_{rea} \tag{2.64}$$

Note that the reader can choose the reaching law ν_{rea} and the function $D(z)$ to have a strong control. The procedure to obtain the error dynamics is similar to that in Section 2.4.3. An example of the prescribed performance approach with a fixed-timed SMC is developed in Chapter 5.

2.6 Robust Model Predictive Control

Model Predictive Control (MPC) is an optimization-based approach for feedback control. Robust MPC is specifically designed to enhance the stability and performance of control systems in the presence of diverse sources of uncertainty. This optimization process aims to find a balance between constraint satisfaction and control performance at each time t (Mesbah, 2016). The

optimization problem is subject to input and state constraints.

$$\begin{aligned} \min_{\mathbf{\Pi}(t)} \quad & J(\bar{\mathbf{z}}(t), \mathbf{\Pi}(t)) \\ \text{s.t.:} \quad & \dot{\bar{z}}(t_p) = A\bar{z}(t_p) + B\nu_{mpc}(t_p) \\ & \Pi(t_p) \in \mathcal{U} \ominus K\Omega_{tube}, \quad \bar{z}(t) \in \mathcal{Z} \ominus \Omega_{tube} \\ & \bar{\xi}(t_f) \in \Omega_\epsilon \end{aligned} \tag{2.65}$$

where $\ominus$ is the Pontryagin difference. The nominal prediction model is:

$$\dot{\bar{z}}(t) = A\bar{z}(t) + B\nu_{mpc}(t) \tag{2.66}$$

where $z \in \mathbb{R}^{2n}$ represents the state vector $z = [\theta^T, \dot{\theta}^T]^T$. And the control input ν_{mpc} is obtained from the optimal control problem in Equation (2.65), the matrices $A \in \mathbb{R}^{2n\times 2n}$ and $B \in \mathbb{R}^{2n\times n}$ are defined as:

$$A = \begin{bmatrix} 0_{n\times n} & I_{n\times n} \\ 0_{n\times n} & 0_{n\times n} \end{bmatrix}, \quad B = \begin{bmatrix} 0_{n\times n} \\ I_{n\times n} \end{bmatrix} \tag{2.67}$$

The prediction time t_p goes from initial time t to the end of the prediction horizon at time t_f. And $\mathbf{\Pi}(t) = [\Pi(t), ..., \Pi(t_f)]$ is the optimized sequence of control inputs, which results in a sequence of predicted states using the nominal model $\bar{\mathbf{z}}(t) = [\bar{z}(t), ..., \bar{z}(t_f)]$. The horizon cost function is defined by:

$$J(\bar{\mathbf{z}}(t), \mathbf{\Pi}(t)) = L(\bar{z}(t_f)) + \int_t^{t_f} I(\bar{z}(t_p), \Pi(t_p))dt_p \tag{2.68}$$

where $I(\bar{z}(t_p), \Pi(t_p)) \in \mathbb{R}^+$ is the stage cost function at prediction time t_p defined as:

$$I(\bar{z}(t_p), \Pi(t_p)) = \frac{1}{2}\bar{\xi}(t_p)^T Q\bar{\xi}(t_p) + \frac{1}{2}\Pi(t_p)^T R\Pi(t_p) \tag{2.69}$$

and the terminal cost function:

$$L(\bar{z}(t_f)) = \frac{1}{2}\bar{\xi}(t_f)^T P\bar{\xi}(t_f) \tag{2.70}$$

where the predicted error at time t_p is:

$$\bar{\xi}(t_p) = \bar{z}(t_p) - z_d(t_p) \tag{2.71}$$

And $Q \in \mathbb{R}^{2n\times 2n}$ and $R \in \mathbb{R}^{n\times n}$ are positive diagonal matrices. And $P \in \mathbb{R}^{14\times 14}$ is a symmetric and positive definite matrix. The dynamic programming method was developed (Bellman, 1966) to solve the optimal control problem (2.65) but suffers from dimensional problems that reduce the application area. Then, we choose the following feedback control law:

$$\nu_{mpc}(t_p) = K_{mpc}\bar{\xi}(t_p) + \Pi(t_p) \tag{2.72}$$

where K_{mpc} is a feedback gain obtained from the solution of the following Ricatti equation:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{2.73}$$

and $\Pi(t_p)$ is the solution for the optimal control problem (2.65) using gradient descend optimization method. The theory of MPC assumes that at future time t_f, a proper P matrix obtained from Equation (2.73) allows optimization along an infinite horizon (Mayne *et al.*, 2000).

2.6.1 Terminal constraint

Let us begin by defining the input and state constraints as $\mathcal{U} = \{\nu(t)|M(\theta)^{-1}\tau_{min} \leq \nu(t) \leq M(\theta)^{-1}\tau_{max}\}$ and $\mathcal{Z} = \{z(t)|z_{min} \leq z(t) \leq z_{max}\}$ respectively. where τ_{min} and τ_{max} are the torque limits, and the state constraints are z_{min} and z_{max}. Note that these constraints play a crucial role in ensuring that the system's behavior complies with specified limits and requirements, both for the control inputs and the state variables, as it evolves over time. This contributes to the overall safety and performance of the control system.

As an additional constraint, the terminal constraint Ω_ϵ or terminal region previously introduced in Equation (2.65) serves the dual purpose of ensuring optimization feasibility and closed-loop

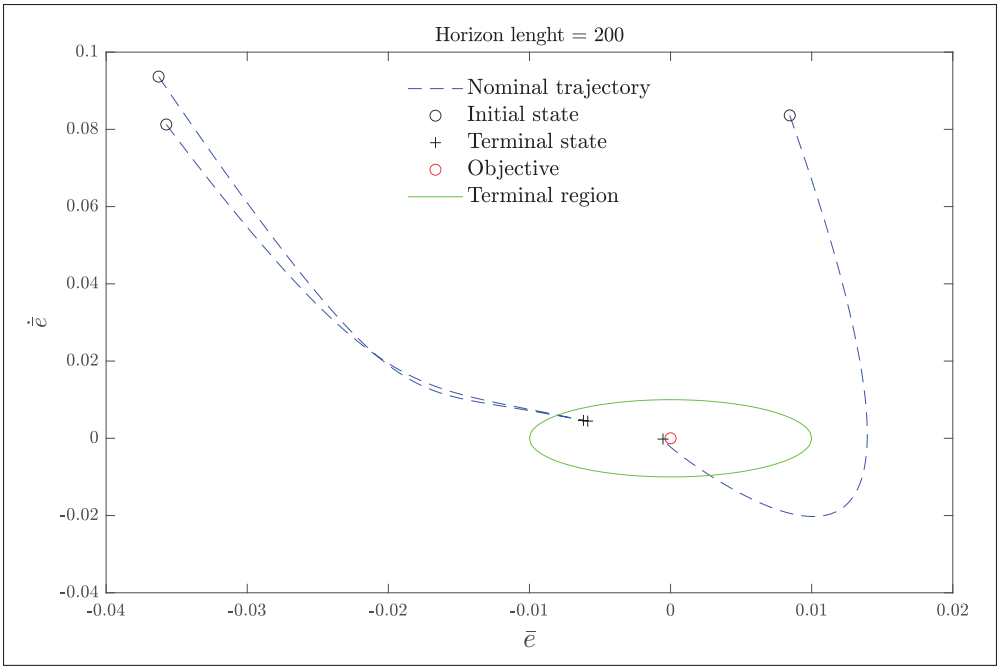

Figure 2.12 Optimization feasibility with 200 steps in the horizon length

stability. Let us use the terms initial state and terminal state referring to $\bar{\xi}(t) = [\bar{e}(t), \dot{\bar{e}}(t)]^T$ and $\bar{\xi}(t_f) = [\bar{e}(t_f), \dot{\bar{e}}(t_f)]^T$ respectively. At the time t, commencing from the initial state $\bar{\xi}(t)$, the optimization is considered feasible if the terminal state $\bar{\xi}(t_f)$ resides within the terminal region Ω_ϵ.

Thus, the terminal constraint is defined by the largest possible constant ϵ such that $\bar{u} \in K_{mpc}\Omega_\epsilon, \forall\, \bar{\xi} \in \Omega_\epsilon$, where $\Omega_\epsilon = \{\bar{\xi} \in \mathbb{R}^{2n} | \bar{\xi}^T P \bar{\xi} \leq \epsilon\}$. With $\bar{u}$ as a subset of the input constraint $\mathcal{U} \in \mathbb{R}^n$.

The feasibility of several independent optimizations with different initial states is exemplified in Figure 2.12, where all terminal states fall within the confines of the terminal region. It's worth noting that a long-horizon was employed, but this choice incurs a higher computational cost. Conversely, Figure 2.13 illustrates another scenario with a short-horizon, wherein the terminal

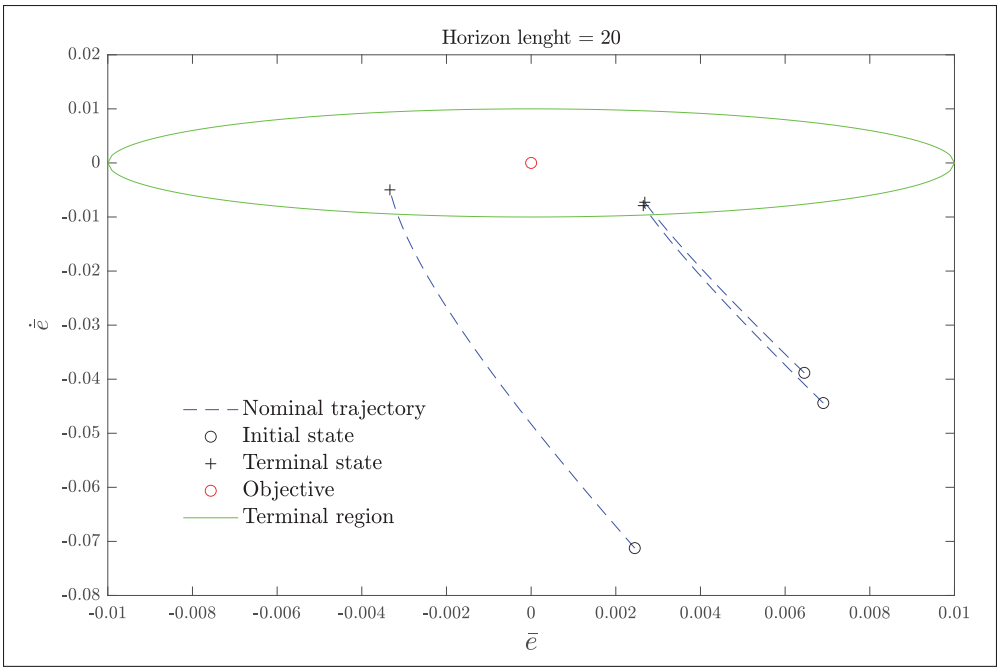

Figure 2.13 Optimization feasibility with 20 steps in the horizon length

states are inside the terminal region but situated farther from the objective compared to the long-horizon case.

The feasible region is the space within which initial states, located inside this region, have the capability to reach the terminal region. It's well established that employing longer horizons expands the area of the feasible region, as illustrated in Figure 2.14. In this figure, the red dashed line delineates the feasible region associated with a horizon length of 20, while the red solid line represents the feasible region with a horizon length of 40.

Furthermore, it has been demonstrated that when combining an SMC approach with MPC, the feasible region undergoes modifications (Incremona *et al.*, 2017), as depicted in Figure 2.15. In this figure, an expansion of the feasible region is illustrated by the red dashed line, in contrast to the feasible region without the SMC approach represented by the red solid line, both for the same horizon length.

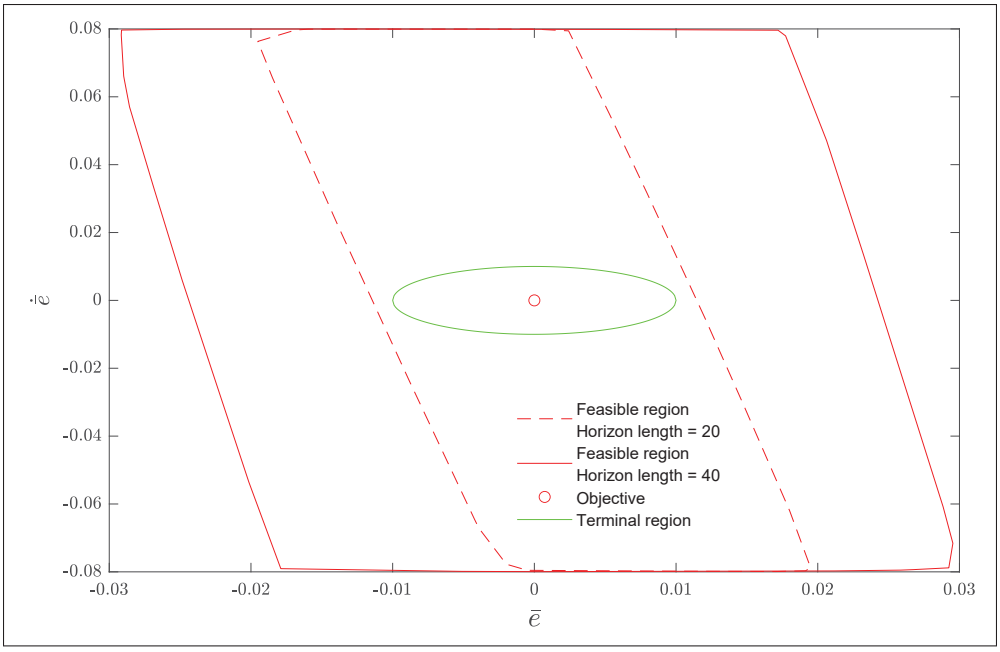

Figure 2.14 Feasible region for different horizon lengths

Thus, when a shorter horizon length is employed in combination with SMC and MPC, the resulting feasible region can be equivalent to that obtained with a longer horizon length, all while enjoying the advantage of reduced computational cost (Rubagotti *et al.*, 2010). A visual comparison is provided in Figure 2.16, where the red dashed line represents the resulting feasible region with SMC and a horizon length of 10, while the red solid line corresponds to the results obtained with a horizon length of 20 alone.

2.6.2 Tube set

Consider the uncertain linear system (2.47), that is written in matrix form as:

$$\dot{z} = Az + B\nu_{mpc} + Bw_u \tag{2.74}$$

The difference between the predicted and measured states is referred to as the measured disturbance, and it signifies the model mismatch resulting from the term Bw_u in the preceding

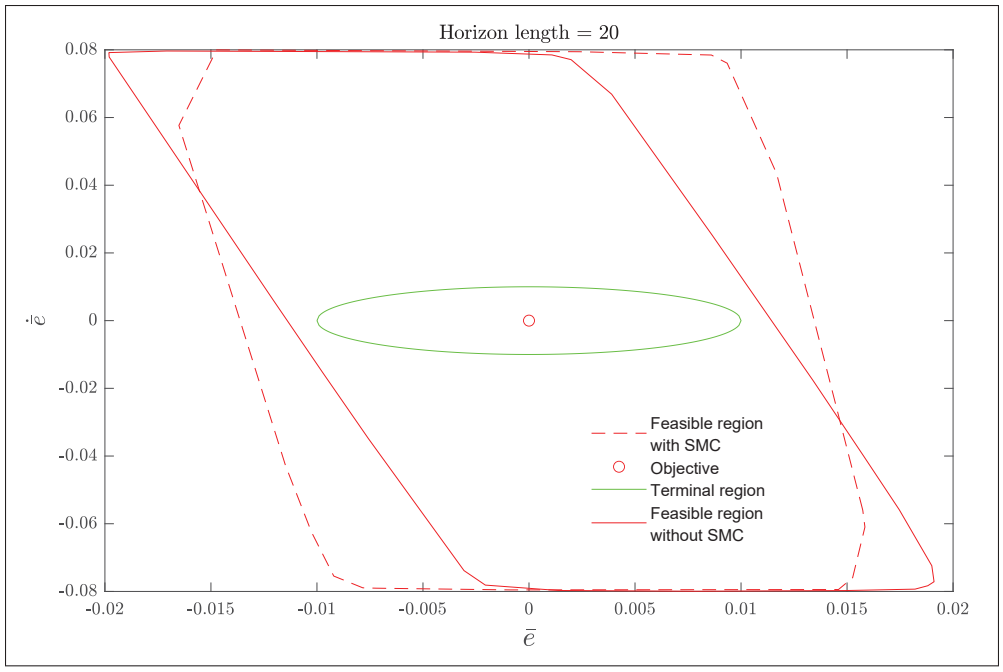

Figure 2.15 Feasible region with and without SMC

equation. With the aid of the nominal prediction model in Equation (2.66) and the uncertain linear system in Equation (2.74), the measured disturbance can be defined as follows:

$$\begin{aligned} \bar{\psi} &= z - \bar{z} \\ &= [\bar{e}_{\psi}, \dot{\bar{e}}_{\psi}]^T \end{aligned} \tag{2.75}$$

Through the measured disturbance $\bar{\psi}$, a region Ω_{tube} in the state space within which the uncertain system (Equation (2.74)) behavior is likely to lie can be modeled. The set Ω_{tube} in Equation (2.65) is commonly referred to as the "tube set". This tube set can be constructed by utilizing the measured disturbance $\bar{\psi}$ computed using measured data from a previous trial (Aswani *et al.*, 2013). The most commonly employed techniques for building the tube set include polyhedral approximations using a convex hull or set-membership approaches employing an ellipse.

Figure 2.17 illustrates an example of how the tube set is modeled using a convex hull based on the measured disturbance, applied here to the 7-DoF ETS-MARSE. Additionally, in Figure

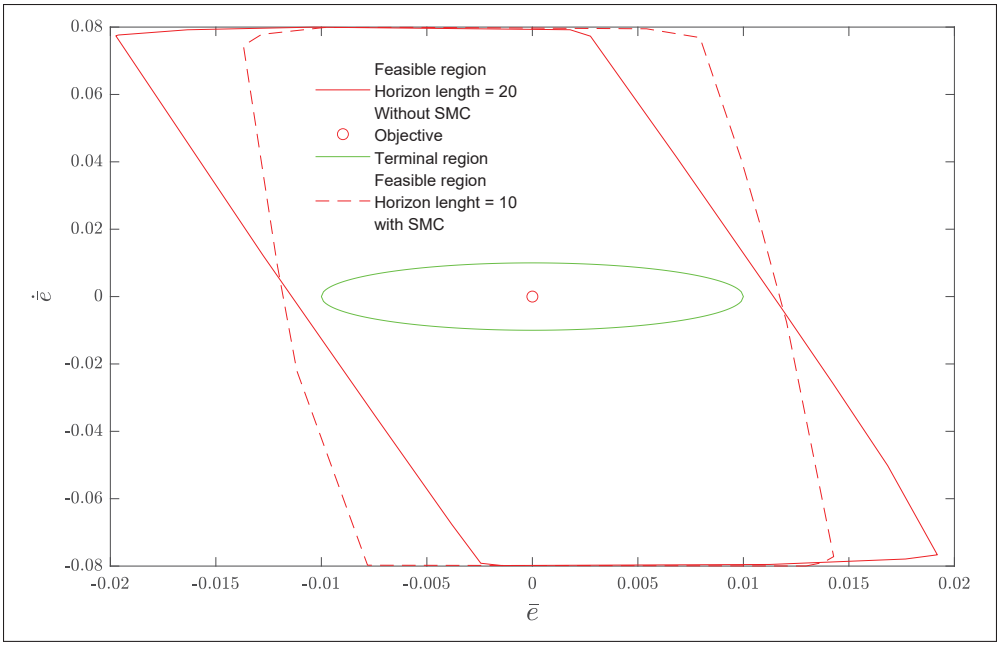

Figure 2.16 Feasible region

2.18, the tube set is depicted, and constructed from the measured disturbance using an elliptical representation. The choice between the convex hull and ellipsoid approaches depends on the specific control problem's requirements. Convex hulls are often favored for their accuracy but may come at a higher computational cost. Ellipsoids are computationally efficient and offer conservative safety guarantees but may not capture the actual shape of the uncertainty region as accurately.

The tube set is used to perform a worst-case analysis. This involves considering the most adverse conditions or disturbances within the tube set and evaluating how the system behaves under these conditions. The goal is to ensure that the system's performance and stability are guaranteed even in the presence of uncertainty (Mayne *et al.*, 2005).

During system operation, the actual state trajectory is continually monitored as shown in Fig. 2.19. The tube set is used to evaluate the uncertainty associated with the current state of the system. If the actual state remains within the tube set, it is considered consistent with the modeled uncertainty. Also, the tube set is used to tighten the constraints for example a velocity

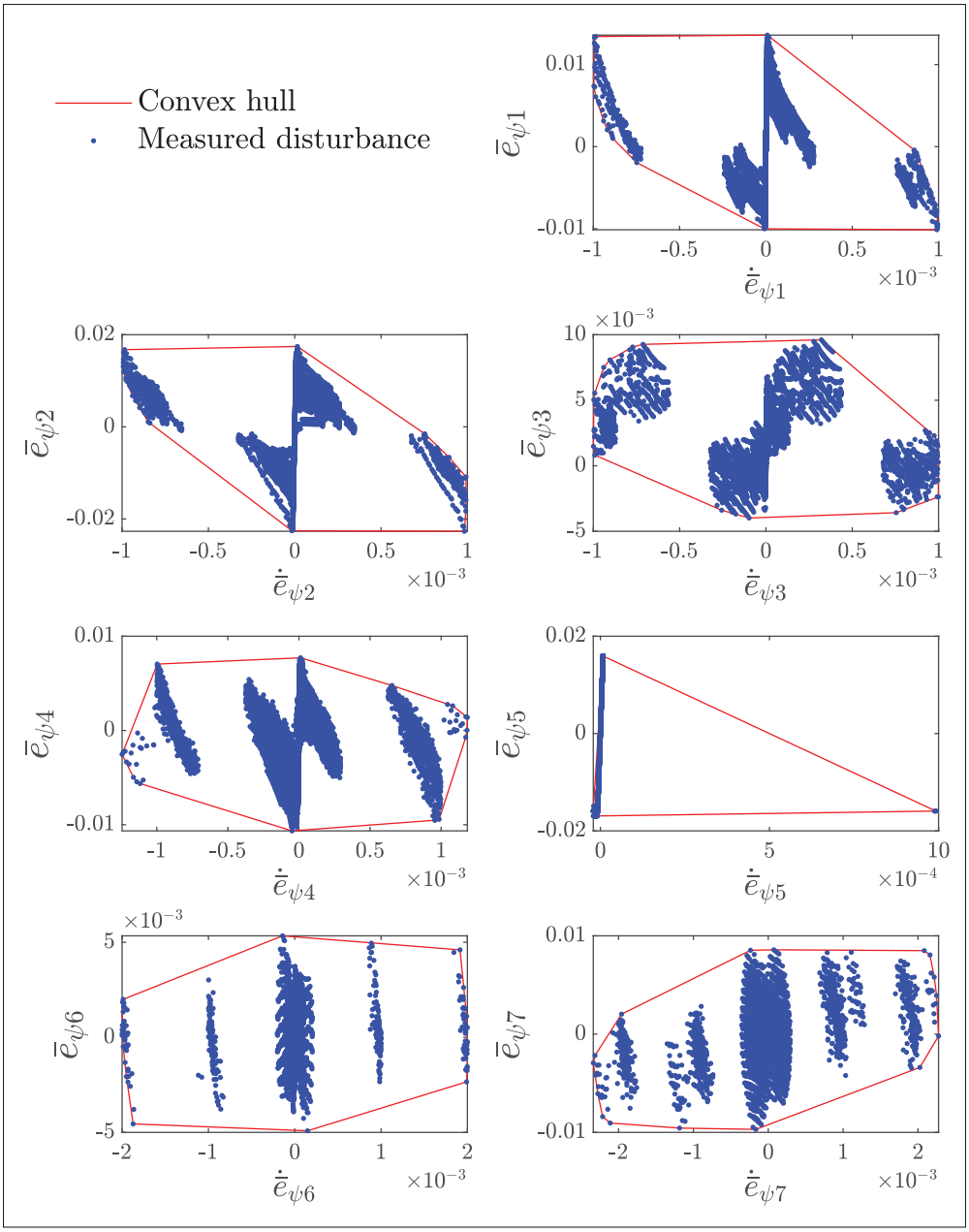

Figure 2.17 Tube set modeled by a convex hull

constraint and the tightened terminal region both shown in Fig. 2.19. By using the tube set to represent and account for uncertainty, robust control systems can provide safety and performance guarantees, ensuring that the system remains stable and meets its objectives even in the presence of uncertainty. The previous advantages are obtained while getting a computational cost similar

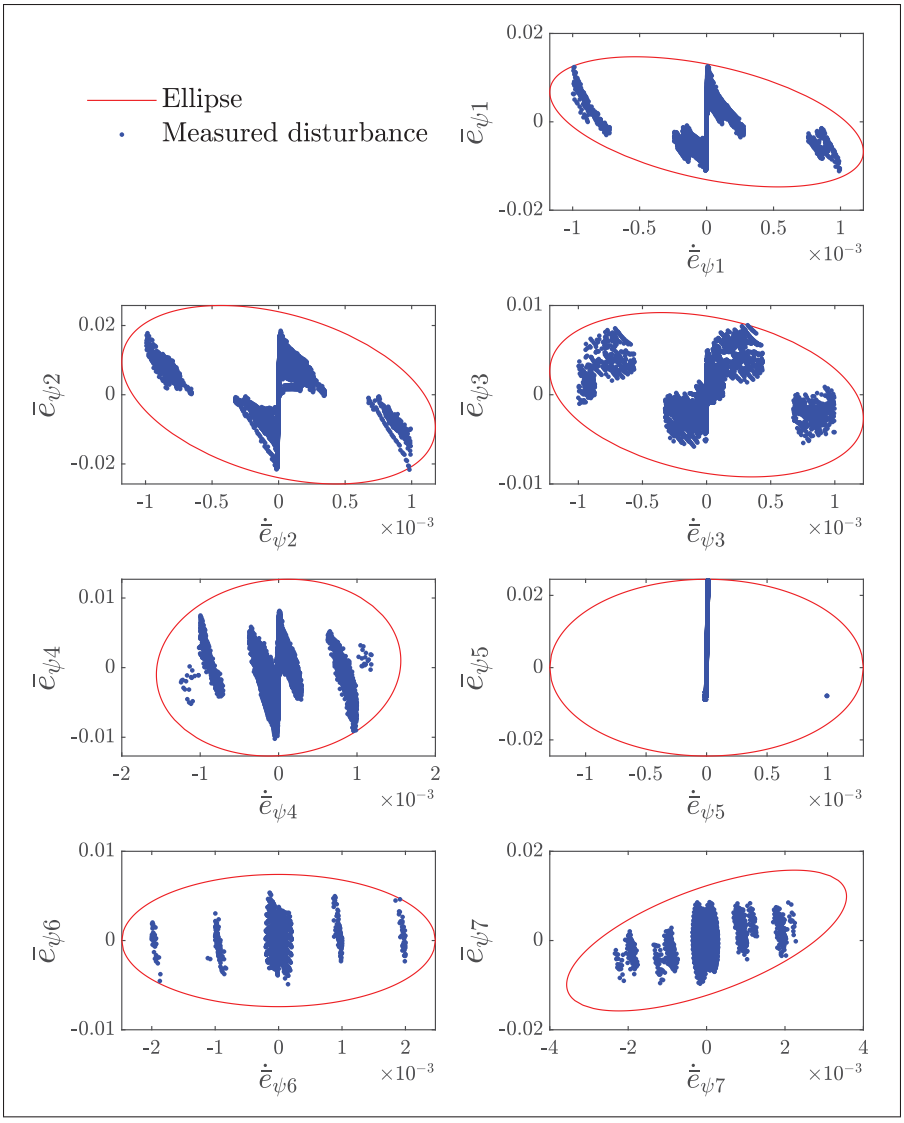

Figure 2.18 Tube set modeled by a ellipse

to the conventional MPC. An implementation of the Robust MPC with an integral sliding mode under learning-based optimization is developed in Chapter 4.

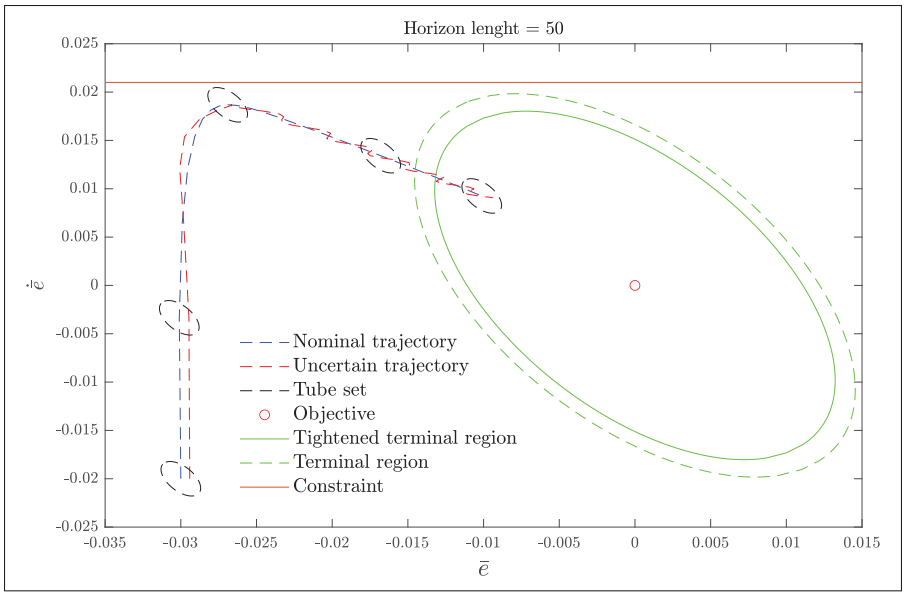

Figure 2.19 Tube approach

CHAPTER 3

LEARNING HUMAN INVERSE KINEMATICS SOLUTIONS FOR REDUNDANT ROBOTIC UPPER-LIMB REHABILITATION

David Bedolla-Martinez[1] , Yassine Kali[1] , Maarouf Saad[1] , Cristobal Ochoa-Luna[2] , Mohammad Habibur Rahman[3]

[1] Département de Génie Mécanique, École de Technologie Supérieure, 1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

[1] Tecnologico de Monterrey, School of Engineering and Sciences Ave. Eugenio Garza Sada 2501, Monterrey, N.L. México, PC: 64849

[1] Mechanical and Biomedical Engineering Department, University of Wisconsin-Milwaukee, Milwaukee, WI, USA, PC: 53211

Paper published in the *Engineering Applications of Artificial Intelligence*, August 2023. https://doi.org/10.1016/j.engappai.2023.106966

3.1 Introduction

Robotic manipulators are electromechanical systems widely used in several real-life and industrial applications. Recently, these highly nonlinear systems have been introduced in physical rehabilitation exercises, where rehabilitation exercises are defined in the task-space and then converted to joint-space for robot control. However, the redundancy in robotic exoskeletons, as is the case in this paper, brings many postures for the same task-space target (end-effector position and orientation). Thus, developing an Inverse Kinematic (IK) model for a rehabilitation robot to compute appropriate IK solutions representing natural bio-mechanical motions for real-time control of such robots is essential.

In the literature, the kinematic properties of the upper-limb have been modeled as a 7 Degrees of Freedom (DoF) redundant manipulator robot (Kim *et al.*, 2011). Although the complexity of the IK problem for redundant robots makes it impossible to obtain algebraic closed-form solutions, the redundancy of the robot increases its dexterity (Gunasekara *et al.*, 2013), which can be exploited by selecting the most natural configurations for the human arm. For general robotic manipulators, several approaches have been proposed to solve the IK problem, e.g.

numerical approaches (Jesus *et al.*, 2022) which are useful for a wide variety of kinematic configurations but not suitable for real-time applications. Learning methods such as neural networks (Duka, 2014) that model the relation between task-space targets to multiple joint angles, without proper generalization (Su *et al.*, 2018). For upper-limb exoskeletons, researchers have previously attempted to mimic the natural upper-limb motion, modeling the relation between a task-space target to joint angles and a trajectory planner parameters using neural networks from collected data (Lauretti *et al.*, 2018), where the modeling error leads to inaccuracy in the end-effector positions and orientations. Also, to mimic the natural movement, researchers attempted maximizing manipulability indices (Gams & Lenarcic, 2006), and through dynamic and kinematic multi-objective optimization (Dalla Gasperina *et al.*, 2020), which do not necessarily provide compatible solutions for the upper-limb.

To overcome the problems mentioned above, the authors in (Tolani & Badler, 1996) introduced a geometric method, the swivel angle approach, to parameterize redundancy. Contrary to the previously mentioned works, the swivel angle approach yields accurate and valid human-type solutions, but the choice of a proper swivel angle to reduce uncoordinated movements between the wearer and the robot is still an open problem (Kim *et al.*, 2012b). There is strong experimental evidence that shows kinematic regularities of arm movement even in the presence of kinematic redundancy (Wang *et al.*, 2019), (Zanchettin *et al.*, 2013). In this sense, to estimate a proper swivel angle given a task-space target, different discomfort criteria have been studied for online optimization: wrist discomfort or joint displacement criteria combined with energetic approach (Almasri & Ouezdou, 2008), total potential energy (Zhao *et al.*, 2014), and manipulability index (Kim & Rosen, 2015). However, it is known that heuristic optimization algorithms are inadequate for real-time implementation, especially when involving multiple criteria, where is difficult to choose the weights, which vary from person to person and from task to task (Campos & Calado, 2009) (e.g. Grasping or reaching tasks). Thus, in order to estimate the swivel angle by optimizing discomfort indices for different users and tasks, a different set of weights must be identified for each user-task combination. Moreover, the optimization algorithms suffer from stability and convergence problems even without the presence of Jacobian singularities (Brahmi *et al.*, 2017b).

To avoid the above-mentioned problems, in this work the Gaussian Process (GP) stochastic learning technique is proposed to model offline the relationship between the task-space targets (inputs) and the swivel angles (outputs), ensuring real-time natural IK solutions. Although the main problem of the GP technique is the computational burden, there is usually a good balance between model complexity and model accuracy which allows real-time robot control implementation, as it is not the case for computationally demanding learning methods (Su *et al.*, 2020) and (Thies *et al.*, 2020). The GP technique has been previously used to identify model disturbances in low dimensional systems (Hewing *et al.*, 2020b), but in this work, among the existing learning models, the GP is used due to its ability to behave well with small datasets (reducing model complexity for real-time application) and because the possibility to compute the output uncertainty (Ostafew *et al.*, 2016a) (relevant for future research on safe trajectory planning). Moreover, contrary to the state-of-the-art, real-time robot control is ensured and the tuning of the weights for the discomfort optimization methods is avoided saving time for the end-users (patients and therapists). To the best of our knowledge, this is the first paper to propose a stochastic learning approach for determining a proper swivel angle in real-time for robotic rehabilitation. Additionally, for the rehabilitation approach to be complete, a feedback controller for trajectory tracking is needed to maintain the stability of the system in the presence of external disturbances produced by a variety of wearers with unknown upper-limb characteristics (lengths, weights, and dexterity skills) and exercises performed in different velocities. Considering these uncertainties, the PID linear controller is not suitable for rehabilitation exercises due to the high non-linearities in the exoskeleton dynamics (Yu & Rosen, 2013). To compensate for non-linearities, computed torque control (Brackbill *et al.*, 2009), Sliding Mode (SM) control (Rahmani & Rahman, 2019), backstepping (Brahmi *et al.*, 2016) and time delay estimation (Fei *et al.*, 2017) have been proposed. However, these control techniques are not designed to operate under constrained conditions and their performance can be compromised when input and state constraints are considered (Yu *et al.*, 2021). Therefore, to ensure safe operation and optimal performance (optimality) during rehabilitation exercises, while also satisfying constraints, this paper considers a Model Predictive Control (MPC) approach (Mayne *et al.*, 2000).

For uncertain systems, the conventional MPC approach is unable to ensure stability (Grüne

et al., 2017). Thus, two robust MPC approaches have been proposed in the literature: Min-Max (Scokaert & Mayne, 1998) and Tube MPC (Langson *et al.*, 2004). It is known that the Min-Max approach is computationally expensive, while Tube MPC has a similar computation burden to the conventional MPC, but its conservativeness prioritizes stability over optimality. Despite these advances, there are still open issues in robust MPC, such as the computation burden for longer prediction horizons needed to satisfy a terminal constraint, as is the case in this paper.

The combination of MPC with SM has been previously proposed, and the SM controller brings various benefits to the MPC (Zhou *et al.*, 2001). The SM controller enables prediction horizon length shortening in MPC (Jafari Fesharaki *et al.*, 2020) reducing the computation burden. Also, the SM controller reduces the difference between the nominal prediction model and the measured states (model uncertainty) improving conservatism in MPC (Spasic *et al.*, 2016). Moreover, the ISM when combined with MPC, actively adapts to the model uncertainty (Rubagotti *et al.*, 2009) and (Incremona *et al.*, 2017). Thus, the SM and MPC combination limits the model uncertainty in rehabilitation exercises improving optimality (Bao *et al.*, 2020) and the feedback linearization combined with SM controller reduces the computation burden in the MPC, by obtaining an uncertain linear system (Rubagotti *et al.*, 2010). In this work, a hierarchical approach that makes use of feedback linearization, an ISM based on the Exponential Reaching Law (ERL) (Fallaha *et al.*, 2011) for chattering reduction, and a robust MPC (with a terminal constraint for optimality) is proposed. The feedback linearization is used here to convert the nonlinear system into an equivalent uncertain linear one, while the ISM rejects the disturbances introduced by the wearer and reduces the model uncertainty. Thus, the robust MPC addresses optimality of the resultant uncertain linear system for safe rehabilitation exercises subject to input and state constraints for the ETS-MARSE robotic exoskeleton. This combination allows the ISM to reduce the undesired chattering without losing stability.

In summary, the main contributions of this paper are:

1. Propose a learning approach based on the GP technique to compute a proper swivel angle, reducing uncoordinated movements between the robot and human arm. Providing unique real-time human-type solutions for an upper-limb redundant exoskeleton robot.

2. Provide a comparison between the Jacobian approach, the discomfort index approach, and the Gaussian process learning technique. The comparison aims to show the improvement of the proposed methodology.
3. Propose a hierarchical control approach, that enables the ISM to compensate the unknown efforts of the wearer's musculoskeletal system, while the robust MPC addresses the safe and optimal evolution of the robotic exoskeleton during rehabilitation exercises subject to input and state constraints.

This paper is structured as follows. In Section 3.2, some useful preliminaries, notations, and definitions are introduced. In Section 3.3, the ETS-MARSE kinematics are detailed, as well as the swivel angle approach, the human inverse kinematics, and the Gaussian Process technique. In Section 3.4, the motion capture system and three swivel angle estimation methods are detailed. In Section 3.5, a feedback linearization procedure is performed, then the ISM to reduce the model uncertainty is applied to the uncertain linear system, and a robust model predictive control approach is presented to address optimality, the recursive feasibility, and closed-loop stability are also proven. In section 3.6, the hardware setup, experiments details and results are shown and discussed. Finally, in section 3.7, the paper ends with the conclusions and future work.

3.2 Preliminaries

3.2.1 Notation and definitions

The terms c_i and s_i denote the $cos(\theta_i)$ and $sin(\theta_i)$ respectively. $\mathbb{R}$ and $\mathbb{R}^n$ denote the 1-dimensional and n-dimensional real numbers spaces. For the matrix $M \in \mathbb{R}^{n\times n}$, $M > 0$ and $M \geq 0$ denote positive definite and semi-definite matrix M, respectively. The matrix $I_{n\times n} \in \mathbb{R}^{n\times n}$ denotes the identity square matrix. The term $z(t)$ denotes the measured value of z at time t. The term $\bar{z}(\tau)$ denotes the predicted values of z at prediction time τ using a nominal prediction model. For a matrix A the terms $\lambda_{max}(A)$ and $\lambda_{min}(A)$ denote the maximum and minimum eigenvalues of A. $det(A)$ is the determinant of A. For the sets $\mathcal{X}$ and $\mathcal{Y}$ with x and y as its elements, the Minkowski sum is defined as $\mathcal{X} \oplus \mathcal{Y} = \{x + y | x \in \mathcal{X}, y \in \mathcal{Y}\}$ and the Pontryagin

set difference is defined as $\mathcal{X} \ominus \mathcal{Y} = \{x | x \oplus \mathcal{Y} \subseteq \mathcal{X}\}$. The function $f(\cdot)$ is of class c^2 if its first and second derivatives exist and are continuous. The function $f(\cdot)$ is a $\mathcal{K}$ function if $f(0) = 0$ and it is strictly increasing. The function $f(\cdot)$ is a $\mathcal{K}_\infty$ function if it is $\mathcal{K}$ and unbounded. The function $f(\cdot, t)$ is a $\mathcal{KL}$ if it is $\mathcal{K}_\infty$ and satisfies $lim_{t \to \infty} f(\cdot, t) = 0$.

3.3 ETS-MARSE's kinematics

The ETS-MARSE (Motion Assistive Robotic-Exoskeleton for Superior Extremity) robot (Rahman, 2012) is a seven DoF exoskeleton redundant robot designed for upper-limb rehabilitation. Due to its dexterity, the ETS-MARSE is able to perform a variety of passive, assisted, and active exercises for the impaired upper-limb. The joints 1, 2 and 3 are assigned to the shoulder, joint 4 to the elbow, and joints 5, 6 and 7 to the wrist movement. For rehabilitation purposes, the joints are constrained for safety reasons (see Fig. 3.1). The modified Denavit-Hartenberg

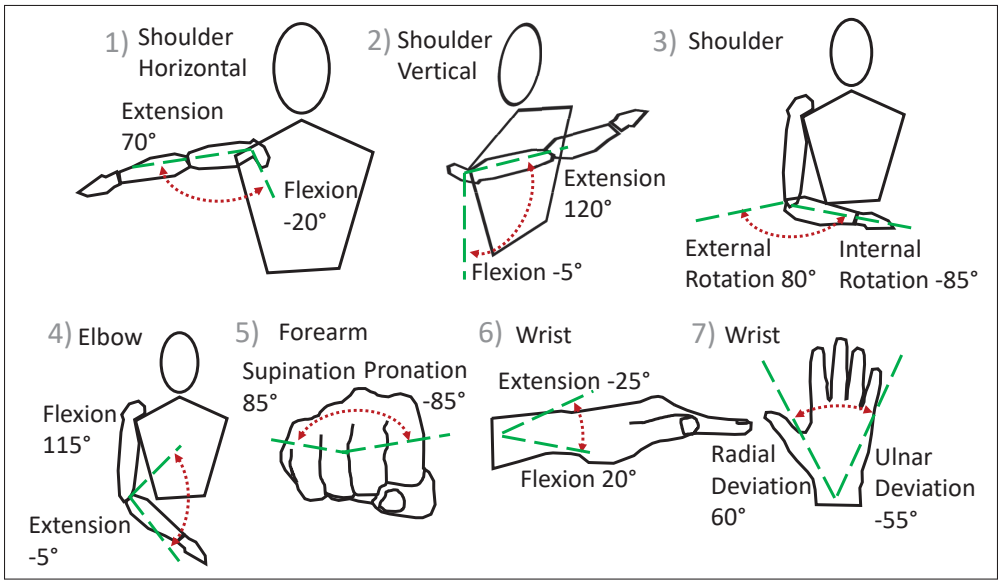

Figure 3.1 Range of motion

(mD-H) parameters (see Table 3.1) were obtained using the coordinate axes (frames) located as shown in Fig. 3.2. From the obtained mD-H parameters, we built the individual homogeneous

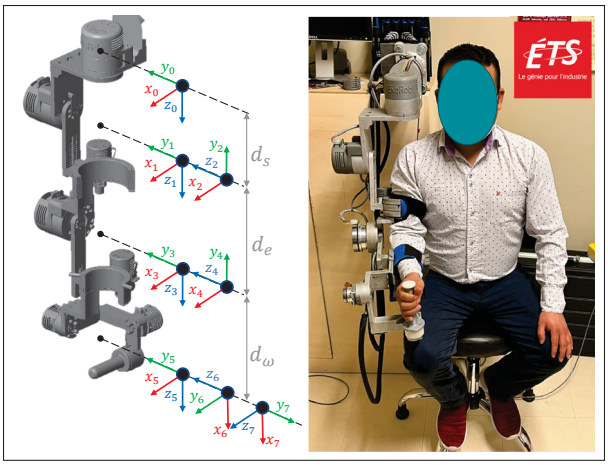

Figure 3.2 Link frame attachment

Table 3.1 Modified D-H parameters

θ_i	a_{i-1}	α_{i-1}	d_i
θ_1	0	0	d_s
θ_2	0	$-\frac{\pi}{2}$	0
θ_3	0	$\frac{\pi}{2}$	d_e
θ_4	0	$-\frac{\pi}{2}$	0
θ_5	0	$\frac{\pi}{2}$	d_w
$\theta_6 - \frac{\pi}{2}$	0	$-\frac{\pi}{2}$	0
θ_7	0	$-\frac{\pi}{2}$	0

transformation matrix between frame $i-1$ and frame i as follows:

$$ {}^{i-1}_{i}T = R_x(\alpha_{i-1})D_{xyz}(a_{i-1},0,0)R_z(\theta_i)D_{xyz}(0,0,d_i) \tag{3.1} $$

where $R_x \in \mathbb{R}^{4\times4}$, $R_z \in \mathbb{R}^{4\times4}$ and $D_{xyz} \in \mathbb{R}^{4\times4}$ are standard rotation and translation matrices (Goldstein, 1980), defined as:

$$
\begin{aligned}
R_x(\alpha_{i-1}) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_{i-1}} & -s_{\alpha_{i-1}} & 0 \\ 0 & s_{\alpha_{i-1}} & c_{\alpha_{i-1}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
R_z(\theta_i) &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
D_{xyz}(\delta_x, \delta_y, \delta_z) &= \begin{bmatrix} 1 & 0 & 0 & \delta_x \\ 0 & 1 & 0 & \delta_y \\ 0 & 0 & 1 & \delta_z \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{3.2}
$$

where δ_x, δ_y and δ_z are the Cartesian displacements. Employing the individual homogeneous transformation matrices, we built the homogeneous transformation between frame 0 and frame 7 as follows:

$$
{}^0_7T = {}^0_1T\,{}^1_2T\,{}^2_3T\,{}^3_4T\,{}^4_5T\,{}^5_6T\,{}^6_7T \tag{3.3}
$$

Obtaining the end-effector position w, as well as the elbow e and shoulder s positions, as follows:

$$
{}^0_1T \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T = \begin{bmatrix} s^T & 1 \end{bmatrix}^T \tag{3.4}
$$

$$
{}^0_1T\,{}^1_2T\,{}^2_3T \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T = \begin{bmatrix} e^T & 1 \end{bmatrix}^T \tag{3.5}
$$

$$
{}^0_1T\,{}^1_2T\,{}^2_3T\,{}^3_4T \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T = \begin{bmatrix} w^T & 1 \end{bmatrix}^T \tag{3.6}
$$

where

$$
s = \begin{bmatrix} 0 & 0 & d_s \end{bmatrix}^T \tag{3.7}
$$

$$e = \begin{bmatrix} d_e c_1 s_2 & d_e s_1 s_2 & d_s + d_e c_2 \end{bmatrix}^T \tag{3.8}$$

$$w = \begin{bmatrix} -d_w(s_4(-c_1c_2c_3 + s_1s_3) - c_1s_2c_4) + d_e c_1 s_2 \\ d_w(s_4(c_1s_3 + s_1c_2c_3) + s_1s_2c_4) + d_e s_1 s_2 \\ d_s + d_w(c_2c_4 - s_2c_3s_4) + d_e c_2 \end{bmatrix} \tag{3.9}$$

3.3.1 Swivel angle approach

The swivel angle was proposed by (Tolani & Badler, 1996) to parameterize the infinite solutions due to redundancy of kinematic models based on the human arm design. This geometric approach adds a new parameter (swivel angle) that enables the development of unique closed-form solutions. The main idea is a rotating frame ($\hat{u}$, $\hat{v}$, $\hat{\eta}$) with center at d, that is defined by the positions of a fixed shoulder, a movable elbow and a fixed wrist (see Fig. 3.3). For a given wrist

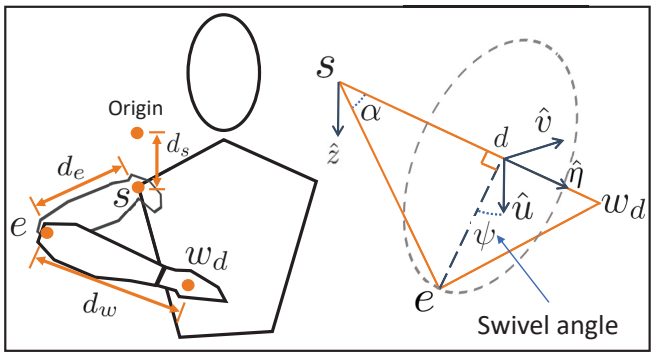

Figure 3.3 Definition of the swivel angle

desired position w_d, a unit vector that defines the vector from the shoulder s to the desired wrist position w_d can be defined by:

$$\hat{\eta} = \frac{1}{\|w_d - s\|}(w_d - s) \tag{3.10}$$

when the elbow moves around $\hat{\eta}$, it draws a circle in a normal plane to $\hat{\eta}$, the rotating frame is completed using the following cross product.

$$\hat{v} = \hat{\eta} \times \hat{u} \tag{3.11}$$

where $\hat{u}$ is a projection of $\hat{z}$.

$$\hat{u} = \frac{1}{\|\hat{z} - (\hat{z}\hat{\eta})\hat{\eta}\|}(\hat{z} - (\hat{z}\hat{\eta})\hat{\eta}) \tag{3.12}$$

And $\hat{z}$ is a unit vector that allows the elbow to reach the lowest height when the swivel angle is zero.

$$\hat{z} = [0 \quad 0 \quad 1]^T \tag{3.13}$$

The center of the rotating frame is defined as:

$$d = cos(\alpha)d_e\hat{\eta} + s \tag{3.14}$$

where α is an angle shown in Fig 3.3 and is calculated by:

$$cos(\alpha) = \frac{d_w^2 - d_e^2 - \|w_d - s\|^2}{-2d_e \|w_d - s\|} \tag{3.15}$$

The radius of the circle normal to $\hat{\eta}$ is defined by:

$$r = sin(\alpha)d_e \tag{3.16}$$

Thus the swivel angle defines the rotation about the unit vector $\hat{\eta}$. Now we can compute the position of the elbow by the swivel angle ψ:

$$e(\psi) = d + r[cos(\psi)\hat{u} + sin(\psi)\hat{v}] \tag{3.17}$$

3.3.2 Human inverse kinematics

In this subsection, the human inverse kinematics (HIK) method for the ETS-MARSE, based on the swivel angle is described. For a desired wrist position w_d and by imposing a swivel angle ψ, the elbow position is known, thus redundancy is solved (Tolani & Badler, 1996). Using Eq. (3.17) we obtain:

$$e(\psi) = \begin{bmatrix} e_x & e_y & e_z \end{bmatrix}^T \tag{3.18}$$

The value of θ_1 is computed by:

$$\theta_1 = atan2(e_y, e_x) \tag{3.19}$$

where $atan2(\cdot,\cdot)$ is the 4-quadrant inverse tangent function that manages the division by zero cases. The value of θ_2 is computed by:

$$\theta_2 = atan2\left(\pm\sqrt{e_x^2 + e_y^2}, ez - ds\right) \tag{3.20}$$

The value of θ_3 is found analytically, substituting $w_d = [w_{d,x} \quad w_{d,y} \quad w_{d,z}]^T$ by w in (3.6) and multiplying both sides by $({}^0_1T\,{}^1_2T)^{-1}$ yielding:

$$\begin{bmatrix} d_w c_3 s_4 \\ -d_e - d_w c_4 \\ d_w s_3 s_4 \\ 1 \end{bmatrix} = \begin{bmatrix} w_{d,x}c_1c_2 + w_{d,y}s_1c_2 - w_{d,z}s_2 + d_s s_2 \\ -w_{d,x}c_1s_2 - w_{d,y}s_1s_2 - w_{d,z}c_2 + d_s c_2 \\ -w_{d,x}s_1 + w_{d,y}c_1 \\ 1 \end{bmatrix} \tag{3.21}$$

Then

$$\begin{aligned} \theta_3 &= atan2(s_3, c_3) \\ s_3 &= -w_{d,x}s_1 + w_{d,y}c_1 \\ c_3 &= w_{d,x}c_1c_2 + w_{d,y}s_1c_2 - w_{d,z}s_2 + d_s s_2 \end{aligned} \tag{3.22}$$

The value of θ_4 is independent of the swivel angle value, and it is obtained by:

$$\theta_4 = \pi \pm acos\left(\frac{d_w^2 + d_e^2 - \|w_d - s\|^2}{2d_e d_w}\right) \tag{3.23}$$

3.3.3 Inverse orientation

Considering equation (3.9), it is clear that the end-effector's position is only defined by θ_1, θ_2, θ_3 and θ_4, while the orientation of the end-effector is defined by θ_5, θ_6 and θ_7. Let us define a desired orientation matrix $O_d \in \mathbb{R}^{4\times4}$ which is built using the desired Euler angles

$o_d = [\gamma_d,\ \beta_d,\ \alpha_d]^T$ with the standard rotation matrices (Goldstein, 1980).

$$O_d = R_z(\gamma_d)R_y(\beta_d)R_x(\alpha_d) \tag{3.24}$$

where $R_y(\beta_d)$ is defined as:

$$R_y(\beta_d) = \begin{bmatrix} c_{\beta_d} & 0 & s_{\beta_d} \\ 0 & 1 & 0 \\ -s_{\beta_d} & 0 & c_{\beta_d} \end{bmatrix} \tag{3.25}$$

Observe that the matrix ${}^4_7T \in \mathbb{R}^{4\times4}$ contains the orientation of the end-effector given θ_5, θ_6 and θ_7.

$${}^4_7T = {}^4_5T\,{}^5_6T\,{}^6_7T \tag{3.26}$$

The matrix ${}^4_7T$ yields:

$${}^4_7T = \begin{bmatrix} s_5s_7 + c_5s_6c_7 & s_5c_7 - c_5s_6s_7 & c_5c_6 & 0 \\ -c_6c_7 & c_6s_7 & s_6 & -d_w \\ -c_5s_7 + s_5s_6c_7 & -c_5c_7 - s_5s_6s_7 & s_5c_6 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.27}$$

Using the matrices O_d and ${}^4_7T$ the following relation is built:

$${}^4_7T = {}^0_4T^{-1}O_d \tag{3.28}$$

Developing the right-hand part of (3.28), one obtains:

$${}^4_7T = \begin{bmatrix} \underline{o}_{11} & \underline{o}_{12} & \underline{o}_{13} & \underline{o}_{14} \\ \underline{o}_{21} & \underline{o}_{22} & \underline{o}_{23} & \underline{o}_{24} \\ \underline{o}_{31} & \underline{o}_{32} & \underline{o}_{33} & \underline{o}_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.29}$$

Considering (3.27) and (3.29), the inverse orientation solution for θ_5, θ_6 and θ_7 are:

$$\begin{aligned}\theta_5 &= atan2\left(\underline{o}_{33}, \underline{o}_{13}\right)\\ \theta_6 &= atan2\left(\underline{o}_{23}, c_6\right)\\ \theta_7 &= atan2\left(\underline{o}_{22}, -\underline{o}_{21}\right)\end{aligned} \tag{3.30}$$

where $c_6 = \sqrt{\underline{o}_{13}^2 + \underline{o}_{33}^2}$, note that the function $atan2(\cdot,\cdot)$ is able to manage division by zero, therefore, the solution remains valid.

3.4 Swivel angle estimation methods

In the current section, the MoCap system is introduced and the swivel angle estimation techniques are detailed for comparison.

3.4.1 Motion capture system

The Kinect V2 was used in this work as a motion capture system, which is a 3D sensor developed by Microsoft (Microsoft, 2015). By measuring the position of the shoulder s, the elbow e, the wrist w, and the hand m (refer to Fig. 3.4), then, the swivel angle ψ and the wrist orientation o (Euler angles) are calculated using the following equations. By considering the vectors from the rotation frame ($\hat{u}$, $\hat{v}$, $\hat{\eta}$) and its center d, one can obtain:

$$\begin{bmatrix} \hat{u} & \hat{v} & \hat{\eta} & d \\ & 0_{1\times3} & & 1 \end{bmatrix} R_z(\psi) = \begin{bmatrix} \hat{u}' & \hat{v}' & \hat{\eta} & d \\ & 0_{1\times3} & & 1 \end{bmatrix} \tag{3.31}$$

where the vector $\hat{u}' \in \mathbb{R}^3$ is the vector pointing from the center of the rotation frame d (defined in equation (3.14)) to the measured elbow position e defined as:

$$\hat{u}' = \frac{1}{||e-d||}(e-d) \tag{3.32}$$

by solving (3.31) for ψ, we can compute the swivel angle. Please note that in the literature rarely this relation (swivel angle from shoulder, elbow, and wrist positions) is presented although is useful for the reader.

$$\psi = atan2(s_\psi / d_\psi, -c_\psi / d_\psi) \tag{3.33}$$

where

$$\begin{aligned}
s_\psi &= \hat{\eta}_x \hat{u}'_z \hat{u}_y - \hat{\eta}_y \hat{u}'_z \hat{u}_x - \hat{\eta}_x \hat{u}'_y \hat{u}_z \\
&+ \hat{\eta}_z \hat{u}'_y \hat{u}_x + \hat{\eta}_y \hat{u}'_x \hat{u}_z - \hat{\eta}_z \hat{u}'_x \hat{u}_y \\
c_\psi &= \hat{\eta}_x \hat{u}'_z \hat{v}_y - \hat{\eta}_y \hat{u}'_z \hat{v}_x - \hat{\eta}_x \hat{u}'_y \hat{v}_z \\
&+ \hat{\eta}_z \hat{u}'_y \hat{v}_x + \hat{\eta}_y \hat{u}'_x \hat{v}_z - \hat{\eta}_z \hat{u}'_x \hat{v}_y \\
d_\psi &= \hat{\eta}_x \hat{u}_y \hat{v}_z - \hat{\eta}_x \hat{u}_z \hat{v}_y - \hat{\eta}_y \hat{u}_x \hat{v}_z \\
&+ \hat{\eta}_y \hat{u}_z \hat{v}_x + \hat{\eta}_z \hat{u}_x \hat{v}_y - \hat{\eta}_z \hat{u}_y \hat{v}_x
\end{aligned} \tag{3.34}$$

where $\cdot_x$, $\cdot_y$ and $\cdot_z$ are the xyz elements of a given vector, respectively. We assume that the matrix built using the vectors of the rotation frame ($\hat{u}$, $\hat{v}$, $\hat{\eta}$) in (3.31) is always invertible, thus, d_ψ is never zero. The measured orientation Euler angles are computed as follows:

$$o = [atan2(\kappa_y, \kappa_x), atan2(\kappa_z, \kappa_x), pi] \tag{3.35}$$

where κ_x, κ_y and κ_z are the first, second and third element of $\kappa = \frac{m-w}{||m-w||}$ respectively.
The data is collected as follows: 1) Several subjects performed random upper-limb motions including elbow flexion/extension combined with shoulder internal/external rotation with natural limits (no specific limits), 2) Using the MoCap system, we measure and store the swivel angle (output), the wrist position and orientation (input task-space target). The swivel angle was unconstrained, which naturally varies even in simple elbow flexion/extension movements. These movements are used to capture most of the workspace for the approaches to achieve better generalization. We discarded a measurement when the inverse kinematic solution is outside the established joint limits in Fig. 3.1. Thus, from the collected data, the following sets are built:

$$\hat{\mathbf{\Psi}} = [\psi_1, ..., \psi_{4933}], \ \hat{\mathbf{H}} = [h_1, ..., h_{4933}] \tag{3.36}$$

with 4933 collected data points, ψ_i and $h_i = [w_i^T, o_i^T]^T$ are the ith measurement of the swivel angle ψ, the wrist position w, and the wrist orientation o, respectively. In the following

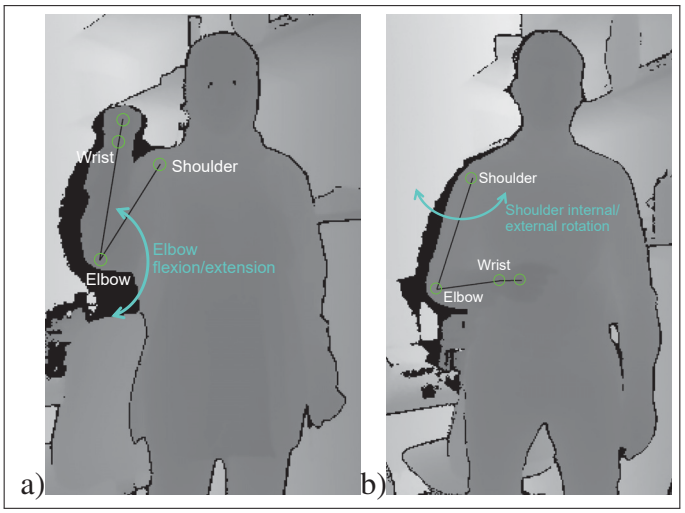

Figure 3.4 Data collection using Kinect V2, combining a) Elbow flexion/extension and b) Shoulder internal/external rotation

subsections, we describe the Jacobian approach, the discomfort index, and the GP methods to estimate a proper swivel angle given a desired task-space target:

$$h_d = [w_d^T, o_d^T]^T. \tag{3.37}$$

where w_d is the desired wrist position and o_d are the Euler angles for the wrist's desired orientation. Note that the discomfort index and the Gaussian process approaches require offline weights tuning and training procedures, respectively. To that end, from the collected data $\hat{\mathbf{\Psi}}$ and $\hat{\mathbf{H}}$ in (3.36), we have chosen 333 random measurements to build the training set, 300 measurements for the validation set, and 4300 measurements for the test set as follows:

$$\begin{aligned}
\mathbf{\Psi}_{tr} &= [\psi_1, ..., \psi_{333}], \mathbf{H}_{tr} = [h_1, ..., h_{333}] \\
\mathbf{\Psi}_{v} &= [\psi_{334}, ..., \psi_{633}], \mathbf{H}_{v} = [h_{334}, ..., h_{633}] \\
\mathbf{\Psi}_{t} &= [\psi_{634}, ..., \psi_{4933}], \mathbf{H}_{t} = [h_{634}, ..., h_{4933}]
\end{aligned} \tag{3.38}$$

Usually, the training set is larger than the validation and test sets, but in this work due to the GP's ability to work with small training datasets, we have chosen the previous set lengths to better compare the approaches.

3.4.2 Jacobian approach

The Jacobian approach is a numerical method to solve the inverse kinematics for a given task-space target. We have considered this approach as it takes the shortest path between the actual and target joint-space configurations using the Newton-Raphson method (Craig, 2005). And it is defined as:

$$\dot{\theta}_w = \alpha_w J^{\dagger}(\theta_w)\dot{e}_w \tag{3.39}$$

where $\theta_w \in \mathbb{R}^4$ are the first four joint angles of the ETS-MARSE needed to locate the end-effector, $\alpha_w \in \mathbb{R}$ is a design parameter, $\dot{e}_w \in \mathbb{R}^3$ is the derivative of the Cartesian error $e_w = w_d - w$, defined between the desired w_d and actual w end-effector positions. And $J^{\dagger} \in \mathbb{R}^{4\times3}$ is the pseudoinverse of the Jacobian defined as:

$$J^{\dagger} = J^T(JJ^T)^{-1} \tag{3.40}$$

where $J \in \mathbb{R}^{3\times4}$ is the Jacobian matrix defined as:

$$J = \begin{bmatrix} \frac{\partial w_x}{\partial \theta_1} & \cdots & \frac{\partial w_x}{\partial \theta_4} \\ \frac{\partial w_y}{\partial \theta_1} & \cdots & \frac{\partial w_y}{\partial \theta_4} \\ \frac{\partial w_z}{\partial \theta_1} & \cdots & \frac{\partial w_z}{\partial \theta_4} \end{bmatrix} \tag{3.41}$$

where w_x, w_y and w_z are the xyz elements of w in (3.9). Note that by iterative solving (3.39), the solutions of the first four joints of the ETS-MARSE are obtained online. Thus, the last three joints are solved by using equation (3.30), as shown in Fig. 3.5.

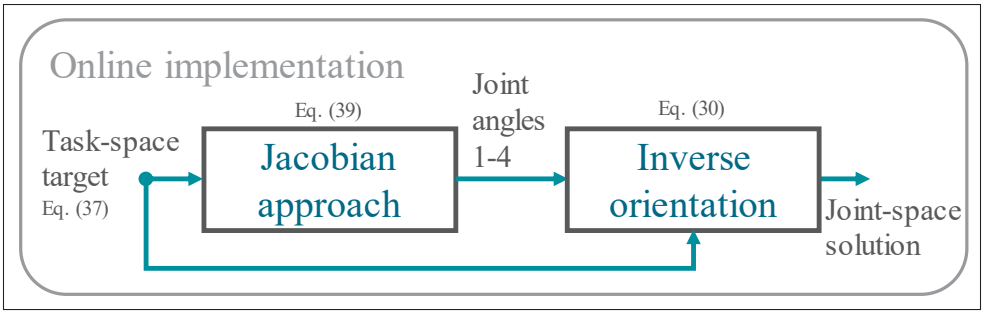

Figure 3.5 Block diagram for Jacobian approach

3.4.3 Discomfort index approach

Researchers have considered the use of mathematical equations to denote the discomfort amount given a posture of the upper-limb. By optimizing a discomfort index, unique human solutions are obtained. In this work, we have considered the following discomfort index to estimate a proper swivel angle.

$$f_{dis} = f_j + f_e + f_s \tag{3.42}$$

where the joint displacement index f_j is the most used discomfort index in the literature, proposed in (Yang *et al.*, 2004) and defined as:

$$f_j = E_1^T (\theta - \theta_m)^2 \tag{3.43}$$

where $\theta \in \mathbb{R}^n$ is the actual joint position, $\theta_m \in \mathbb{R}^n$ is the center position of the joints for a given comfortable initial posture, and $E_1 \in \mathbb{R}^n$ is a weighting constant vector that depends on the user and it can be chosen as proposed in (Yang *et al.*, 2004) or tuned from data. Note that the wrist discomfort index is included in Eq. (3.43). Moreover, the elbow's gravitational potential energy is defined as:

$$f_e = E_2 m_e g h_e \tag{3.44}$$

where $h_e = -(d_s + d_e c_2)$ is the z-component of vector e in (3.8). $E_2 \in \mathbb{R}^+$ is a weighting scalar, m_e is the nominal mass of the elbow and $g = 9.81 m/s^2$ is the gravitational constant. Lastly, the

rotational spring potential energy f_s is defined as follows (Zhao *et al.*, 2014):

$$f_s = E_3(\psi_n - \psi)^2 \tag{3.45}$$

where ψ is the actual swivel angle and ψ_n is the natural swivel angle which depends on the task. E_3 is the spring stiffness constant. Note that the potential energy of the spring is zero when the swivel angle is equal to the natural swivel angle. The weights (E_1, E_2, E_3, and ψ_n) described in this section are tuned offline to best estimate the measured data (swivel angle) as shown in Fig. 3.6 a), and the online implementation solves (3.46) (See Fig. 3.6 b)).

$$\psi_{i+1} = \psi_i + \alpha_\psi \frac{\partial f_{dis}}{\partial \psi_k} \tag{3.46}$$

where ψ_i is the swivel angle solution at optimization iteration i and α_ψ is a design parameter.

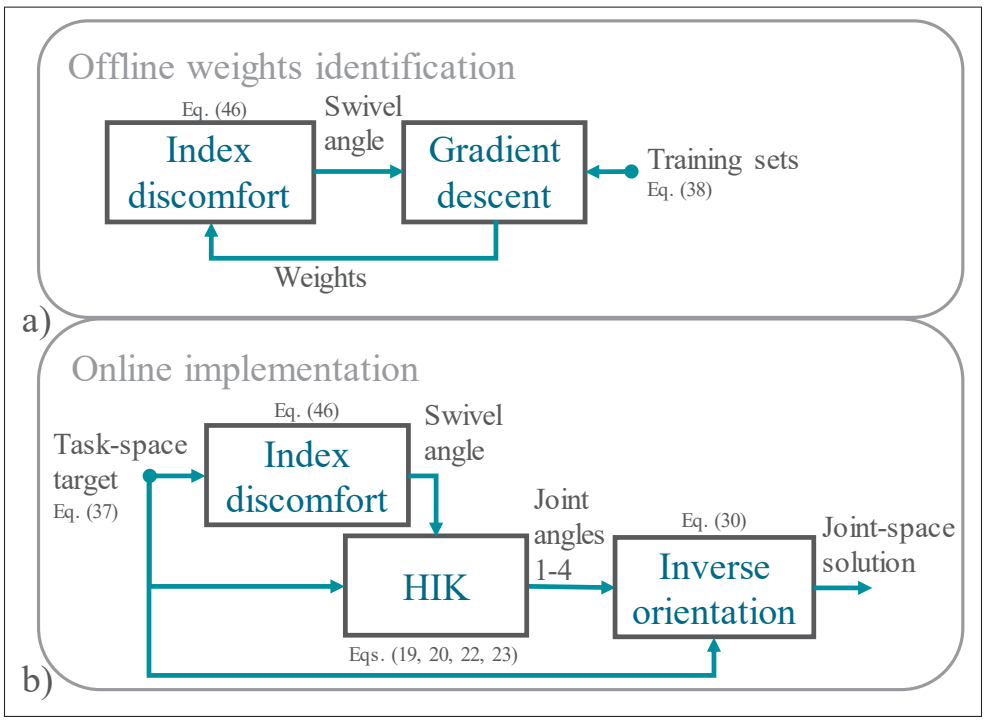

Figure 3.6 Block diagram for discomfort index approach a) offline weight tuning and b) online implementation

3.4.4 Gaussian process for learning natural solutions

In this part, p random points from the training sets $\mathbf{\Psi}_{tr}$ and $\mathbf{H}_{tr}$ (Equation (3.38)), which are defined as $\mathbf{\Psi} \in \mathbb{R}^{1\times p}$ and $\mathbf{H} \in \mathbb{R}^{6\times p}$ will be modeled using the Gaussian Process learning technique. The resultant model will generate a proper swivel angle for any target that lies inside the training set. Given an arbitrary desired position and orientation of the wrist $h_d \in \mathbf{R}^6$ (also called regressor), we can model both the measured swivel angle set $\mathbf{\Psi}$ and the modeled swivel angle ψ_d jointly as a Gaussian distribution.

$$\begin{bmatrix} \mathbf{\Psi}^T \\ \psi_d \end{bmatrix} \sim \mathcal{N}\left(0_{p+1\times 1}, \begin{bmatrix} \Upsilon & k_d(h_d)^T \\ k_d(h_d) & k_r(h_d, h_d) \end{bmatrix}\right) \tag{3.47}$$

where $\Upsilon \in \mathbb{R}^{p\times p}$ is the covariance matrix. The (i, j) element of Υ is equal to $k_r(h_i, h_j)$, $i = 1 \rightarrow p$, $j = 1 \rightarrow p$ with $h_i \in \mathbf{H}$ and $h_j \in \mathbf{H}$. For the arbitrary regressor input h_d the term $k_d(h_d) \in \mathbb{R}^{1\times p}$ is defined as:

$$k_d(h_d) = [k_r(h_d, h_1), ..., k_r(h_d, h_p)] \tag{3.48}$$

The covariance function or kernel function used is the square exponential (Williams & Rasmussen, 2006) defined as:

$$\begin{aligned} k_r(h_i, h_j) = \sigma_f^2 \exp\left(-\frac{1}{2}(h_i - h_j)^T l^{-2}(h_i - h_j)\right) \\ +\sigma_n^2 \delta_{ij} \end{aligned} \tag{3.49}$$

It is assumed that the modeled ψ_d is added with a zero mean noise and variance $\sigma_n \in \mathbb{R}^+$. The Kronecker delta $\delta_{ij} \in \{0, 1\}$ is defined as:

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & Otherwise \end{cases} \tag{3.50}$$

The hyperparameters $\sigma_f \in \mathbb{R}^+$, $l = diag([l_1, ..., l_6])$ and σ_n are obtained by maximizing the log marginal likelihood (Williams & Rasmussen, 2006) defined as:

$$log(p_r(\mathbf{\Psi}|\mathbf{H})) = -\frac{1}{2}\mathbf{\Psi}\Upsilon^{-1}\mathbf{\Psi}^T - \frac{1}{2}log(det(\Upsilon)) - \frac{p}{2}log(2\pi) \tag{3.51}$$

where $p_r(\mathbf{\Psi}|\mathbf{H})$ is the probability function of the observed output $\mathbf{\Psi}$ given the observed inputs $\mathbf{H}$ considering the tuned hyperparameters. The mean and variance of the modeled ψ_d given an arbitrary input h_d is normally distributed (Williams & Rasmussen, 2006) and can be obtained by:

$$\psi_d \sim \mathcal{N}(\psi_\mu, \psi_\Sigma) \tag{3.52}$$

where ψ_μ and ψ_Σ are the mean and variance of the modeled ψ_d respectively and are defined by:

$$\psi_\mu = k_d(h_d)\Upsilon^{-1}\mathbf{\Psi}^T \tag{3.53}$$

$$\psi_\Sigma = k_r(h_d, h_d) - k_d(h_d)\Upsilon^{-1}k_d(h_d)^T \tag{3.54}$$

Equations (3.53) and (3.54) were derived by the multivariate Gaussian linear transformation with Equation (3.47). In this work, the variance value (3.54) was not used but for future work about estimating the uncertainty in the predicted swivel angle for safe online trajectory planning. The proposed swivel angle estimation approach is shown in Fig. 3.7 for a) offline training and b) online implementation.

3.5 Robust model predictive control with integral sliding mode

To ensure safe robotic rehabilitation exercises by satisfying input and state constraints, an MPC with ISM combination is proposed as shown in Fig. 3.8. Inspired by (Incremona *et al.*, 2017), first a feedback linearization to compensate for known dynamics is used, obtaining an equivalent uncertain linear system and the ISM is used to overcome external disturbances produced by the interaction between the exoskeleton and the wearer. Then, the robust MPC approach is used to

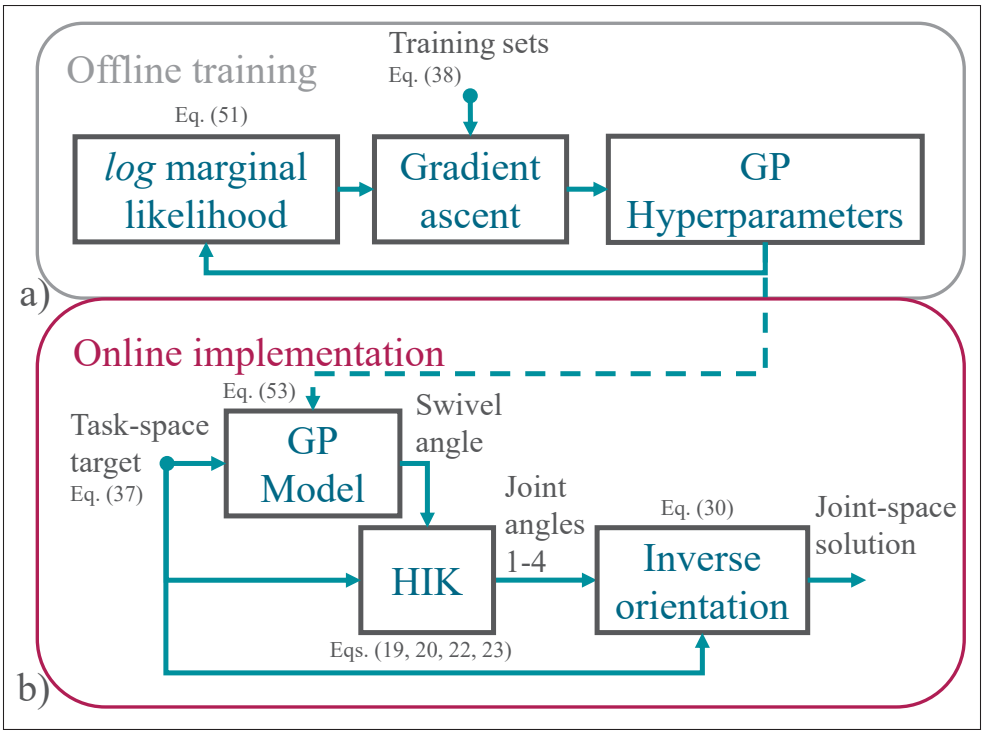

Figure 3.7 Block diagram of the proposed HIK approach for a) offline GP training and b) online real-time implementation

ensure the optimal evolution of the robot in the trajectory tracking by handling a linear system with reduced uncertainty amount (Rubagotti *et al.*, 2010).

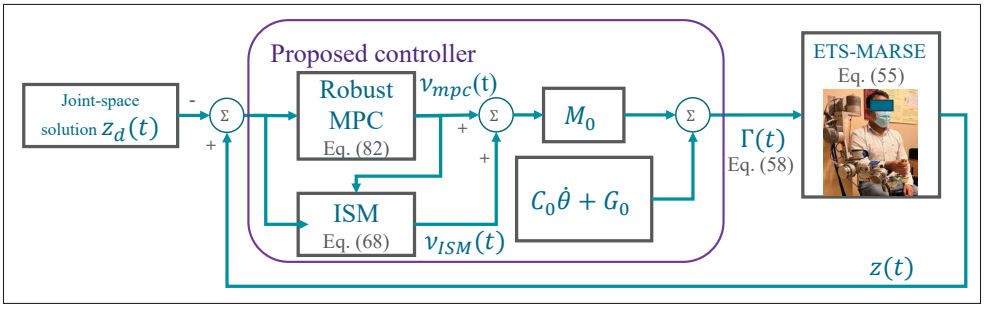

Figure 3.8 Block diagram for the proposed controller

3.5.1 Feedback linearization

The ETS-MARSE's dynamic model can be expressed by the well-known rigid body differential equation form.

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) + F(\dot{\theta}) = \Gamma(t) + \Gamma_{dis} \tag{3.55}$$

where $\theta \in \mathbb{R}^n$ is the vector containing the angular position for each joint, with $n = 7$ as the number of joints, $\dot{\theta} \in \mathbb{R}^n$ is the joint angular velocity vector, $\Gamma(t) \in \mathbb{R}^n$ is the generalized torques vector, $\Gamma_{dis} \in \mathbb{R}^n$ is the unknown disturbance forces vector, $M(\theta) \in \mathbb{R}^{n\times n}$ is the positive definite inertia matrix, $C(\theta, \dot{\theta}) \in \mathbb{R}^{n\times n}$ is the Coriolis and centrifugal forces matrix, $G(\theta) \in \mathbb{R}^n$ is the gravitational forces vector, and $F(\dot{\theta}) \in \mathbb{R}^n$ is the nonlinear friction. Let us define M_0 and M_u as the nominal and unknown parts of $M(\theta)$. Analogously, C_0, C_u, G_0 and G_u as the nominal and unknown parts of $C(\theta, \dot{\theta})$ and $G(\theta)$ respectively. Considering the friction as an unknown disturbance, the acceleration of the joints can be obtained as:

$$\ddot{\theta} = M_0^{-1}\left(\Gamma(t) - C_0\dot{\theta} - G_0\right) + \omega_0(t) \tag{3.56}$$

where $\omega_0(t) \in \mathbb{R}^n$ represents the uncertainty model and it is defined as:

$$\omega_0(t) = M_0^{-1}\left(\Gamma_{dis} - M_u\ddot{\theta} - C_u\dot{\theta} - G_u - F(\dot{\theta})\right) \tag{3.57}$$

A compensation torque obtained from the nominal parts of (3.55) gives.

$$\Gamma(t) = M_0\nu(t) + C_0\dot{\theta} + G_0 \tag{3.58}$$

where $\nu(t) \in \mathbb{R}^n$, to be designed later, is the virtual control that drives the system to the control objective. Substituting (3.58) into (3.56), one obtains the following uncertain linear system in state-space representation.

$$\dot{z}(t) = Az(t) + B\nu(t) + B\omega_0(t) \tag{3.59}$$

where $z(t) \in \mathbb{R}^{2n}$ represents the state vector $z(t) = [\theta^T, \dot{\theta}^T]^T$. The matrices $A \in \mathbb{R}^{2n \times 2n}$ and $B \in \mathbb{R}^{2n \times n}$ are defined as:

$$A = \begin{bmatrix} 0_{n\times n} & I_{n\times n} \\ 0_{n\times n} & 0_{n\times n} \end{bmatrix}, \quad B = \begin{bmatrix} 0_{n\times n} \\ I_{n\times n} \end{bmatrix} \tag{3.60}$$

where $0_{n\times n}$ is a zero matrix. The virtual control $\nu(t)$ has two parts as follows:

$$\nu(t) = \nu_{ISM}(t) + \nu_{mpc}(t) \tag{3.61}$$

where $\nu_{ISM}(t)$ is used to reduce the uncertainty amount $\omega_0(t)$ in (3.59), and $\nu_{mpc}(t)$ is the solution to the optimal control problem. Consider the following assumptions:

Assumption 1 The input and state constraint polytopes sets $\mathcal{U} \in \mathbb{R}^n$ and $\mathcal{Z} \in \mathbb{R}^{2n}$ are convex, compact and bounded, where the sets $\mathcal{U} = \{\nu(t) | {M_0}^{-1}\Gamma_{min} \leq \nu(t) \leq {M_0}^{-1}\Gamma_{max}\}$ and $\mathcal{Z} = \{z(t) | z_{min} \leq z(t) \leq z_{max}\}$.

Assumption 2 The unknown model disturbance $\omega_0(t) \in \mathcal{W}$ is locally Lipschitz, in other words, $\omega_0(t)$ is locally continuous and bounded by the constant ω_{max}.

$$0 < \|\omega_0(t)\| \leq \omega_{max} < \infty \tag{3.62}$$

3.5.2 Integral sliding mode

An ISM approach has been designed to reduce the uncertainty amount $\omega_0(t)$ in (3.59), therefore, improving optimality in the robust MPC. The integral sliding surface (Fridman *et al.*, 2014) is defined as:

$$\varsigma(t) = \Xi[e(t) - e(0) - \int_0^t (Az(t) + B\nu_{mpc}(t) - \dot{z}_d(t))]dt \tag{3.63}$$

where $\Xi \in \mathbb{R}^{n\times 2n}$ is chosen to obtain a nonsingular matrix ΞB. $\nu_{mpc}(t)$ is the output of the optimal control problem (3.76) which is a zero vector at time $t = 0$. The state error $e(t)$ at time

t is defined as:

$$e(t) = z(t) - z_d(t) \tag{3.64}$$

Considering the nominal dynamics, the time derivative of the sliding surface yields:

$$\begin{aligned}\dot{\varsigma}(t) =& \Xi(\dot{e}(t) - Az(t) - B\nu_{mpc}(t) + \dot{z}_d(t)) \\ =& \Xi(Az(t) + B(\nu_{ISM}(t) + \nu_{mpc}(t)) - \dot{z}_d(t) \\ & - Az(t) - B\nu_{mpc}(t) + \dot{z}_d(t)) \\ =& \Xi B\nu_{ISM}(t)\end{aligned} \tag{3.65}$$

Now the Sliding Mode Control Exponential Reaching Law (Fallaha *et al.*, 2011) is defined as:

$$\dot{\varsigma}(t) = -K(\varsigma(t))sign(\varsigma(t)) \tag{3.66}$$

where the ith element of $K(\varsigma(t))$ is defined for $i = 1, ..., n$:

$$K(\varsigma_i(t)) = diag\left(\frac{k_i}{\delta_i + (1-\delta_i)e^{-\alpha_i|\varsigma_i(t)|^{\rho_i}}}\right) \tag{3.67}$$

where $k_i \in \mathbb{R}^+$, $\delta_i \in [0, 1]$, $\alpha_i \in \mathbb{R}^+$ and $\rho_i \in \mathbb{R}^+$ are design parameters. Combining (3.66) and (3.65) yields the virtual control:

$$\nu_{ISM}(t) = (\Xi B)^{-1}[-K(\varsigma(t))sign(\varsigma(t))] \tag{3.68}$$

where the function $sign(\varsigma_i)$ is defined as:

$$sign(\varsigma_i) = \begin{cases} 1, & \varsigma_i(t) > 0 \\ 0, & \varsigma_i(t) = 0 \\ -1, & \varsigma_i(t) < 0 \end{cases} \tag{3.69}$$

For the stability analysis under the ISM, consider the following Lyapunov positive definite function of class c^2 and $\mathcal{K}_\infty$ function.

$$V(t) = \frac{1}{2}\varsigma(t)^T\varsigma(t) \tag{3.70}$$

Considering the uncertainties, the derivative yields:

$$\begin{aligned}\dot{V}(t) =&\frac{1}{2}\dot{\varsigma}(t)^T\varsigma(t) + \frac{1}{2}\varsigma(t)^T\dot{\varsigma}(t) = \varsigma(t)^T\dot{\varsigma}(t)\\ =&\varsigma(t)^T(\Xi B\nu_{ISM}(t) + \Xi B\omega_0(t))\end{aligned} \tag{3.71}$$

Substituting (3.68) into (3.71) yields:

$$\dot{V}(t) = \varsigma(t)^T(-K(\varsigma(t))sign(\varsigma(t)) + \Xi B\omega_0(t)) \tag{3.72}$$

The asymptotic stability is ensured considering Assumption 2. Thus, the following inequality holds:

$$\|K(\varsigma(t))\| > \|\Xi B\|\,\omega_{max},\ \forall\, t \tag{3.73}$$

From Equation (3.73), we ensure that the uncertain linear system (3.59) is asymptotically stable under the feedback controller (3.68). Thus, the Lyapunov function (3.70) is $\mathcal{KL}$ function. Although, the ISM by itself is able to fully reject the uncertainties. However, a high chattering will appear. So, to limit the chattering level, lower switching gains should be chosen. Thus, the ISM deals with a part of the model uncertainties while the remaining part will be handled by the robust MPC without stability lost. To that end, consider the resulting uncertain system from (3.59) under the virtual control (3.61):

$$\dot{z}(t) = Az(t) + B\nu_{mpc}(t) + B\omega(t) \tag{3.74}$$

where $\omega(t) = \nu_{ISM}(t) + \omega_0(t)$ is the remaining model disturbance.

Assumption 3 The unknown model disturbance $\omega(t) \in \mathcal{W}$ is locally Lipschitz, in other words, $\omega(t)$ is locally continuous and bounded by the constant ω_2.

$$0 < \|\omega(t)\| \leq \omega_2 < \omega_{max} < \infty \tag{3.75}$$

Note that **Assumption 3** highlights the fact that $\|\omega_0(t)\| > \|\omega(t)\|$.

3.5.3 Optimal control problem

The following optimal control problem is solved by an optimization procedure subject to input and state constraints (Bellman, 1966). The optimization seeks tradeoffs between constraint satisfaction (Mesbah, 2016) and the control performance at each time t.

$$\begin{aligned} \min_{\mathbf{\Pi}(t)} \quad & J(\bar{\mathbf{z}}(t), \mathbf{\Pi}(t)) \\ \text{s.t.:} \quad & \dot{\bar{z}}(\tau) = A\bar{z}(\tau) + B\nu_{mpc}(\tau),\ I.C.\ \bar{z}(t) := z(t) \\ & \Pi(\tau) \in \mathcal{U} \ominus K\Omega_{tube}, \quad \bar{z}(t) \in \mathcal{Z} \ominus \Omega_{tube} \\ & \bar{e}(t_f) \in \Omega_\alpha \end{aligned} \tag{3.76}$$

where the nominal prediction model is:

$$\dot{\bar{z}}(t) = A\bar{z}(t) + B\nu_{mpc}(t) \tag{3.77}$$

and where the tube set Ω_{tube} is a convex hull set built as shown in (Aswani *et al.*, 2013) by using the model mismatch from a previous trial. The prediction time τ goes from initial time t to the end of the prediction horizon at time t_f. At each sampling time, the state of the nominal prediction model $\bar{z}(t)$ is initialized using the measured state $z(t)$, and the predicted error at the end of the prediction horizon $\bar{e}(t_f)$ must reach the terminal region Ω_α to be defined later. And $\mathbf{\Pi}(t) = [\Pi(t), ..., \Pi(t_f)]$ is a sequence of control inputs and $\bar{\mathbf{z}}(t) = [\bar{z}(t), ..., \bar{z}(t_f)]$ is the

sequence of predicted states using the nominal model. The horizon cost function is defined by:

$$J(\bar{\mathbf{z}}(t), \mathbf{\Pi}(t)) = L(\bar{z}(t_f)) + \int_t^{t_f} I(\bar{z}(\tau), \Pi(\tau))d\tau \tag{3.78}$$

where $I(\bar{z}(\tau), \Pi(\tau)) \in \mathbb{R}$ is the stage cost function at prediction time τ defined as:

$$I(\bar{z}(\tau), \Pi(\tau)) = \frac{1}{2}\bar{e}(\tau)^T Q\bar{e}(\tau) + \frac{1}{2}\Pi(\tau)^T R\Pi(\tau) \tag{3.79}$$

with terminal cost function:

$$L(\bar{z}(t_f)) = \frac{1}{2}\bar{e}(t_f)^T P\bar{e}(t_f) \tag{3.80}$$

where the predicted error at time τ is:

$$\bar{e}(\tau) = \bar{z}(\tau) - z_d(\tau) \tag{3.81}$$

And $Q \in \mathbb{R}^{2n\times 2n}$ and $R \in \mathbb{R}^{n\times n}$ are positive diagonal matrices. And $P \in \mathbb{R}^{2n\times 2n}$ is a symmetric and positive definite matrix to be defined later. The dynamic programming method was developed (Bellman, 1966) to solve the optimal control problem (3.76) but suffers from dimensional problems that reduce the area of application. Then, we choose the following feedback control law:

$$\nu_{mpc}(\tau) = K_{mpc}\bar{e}(\tau) + \Pi(\tau) \tag{3.82}$$

Assumption 4 In (3.78) $L(\bar{z}(t_f))$ and $I(\bar{z}(\tau), \Pi(\tau))$ are continuous and $\mathcal{K}$ functions, satisfying $L(0) = 0$ and $I(0, 0) = 0$.

3.5.4 Terminal ingredients

The terminal ingredients are the terminal penalty and terminal region, the latter will serve as the terminal constraint. Consider the terminal feedback controller that is not necessarily used in the real implementation.

$$\bar{u}(t) = K_{mpc}\bar{e}(t) \tag{3.83}$$

where the gain $K_{mpc} \in \mathbb{R}^{n\times 2n}$ is used to calculate the terminal penalty and terminal region offline.

$$K_{mpc} = [diag(k_{11}, ..., k_{n1}), diag(k_{12}, ..., k_{n2})] \tag{3.84}$$

where $k_{i1} < 0$, $k_{i2} < 0$ and $k_{i2}^2 + 4k_{i1} > 0$ for $i = 1, ..., n$ (Dai *et al.*, 2020). By following the procedure proposed in (Chen & Allgöwer, 1998), we ensure recursive feasibility and closed loop stability. It can be proved that the linear model (3.77) is controllable, then we choose a feedback gain K_{mpc} such that $A + BK_{mpc}$ is stable. We define a positive constant η satisfying $\eta < -\lambda_{max}(A + BK_{mpc})$. Now the terminal penalty matrix P is determined by solving the following equation.

$$A_\eta^T P + PA_\eta + Q + K_{mpc}^T R K_{mpc} \leq 0 \tag{3.85}$$

where

$$A_\eta = A + BK_{mpc} + \eta I_{2n\times 2n} \tag{3.86}$$

It is assumed that at the prediction time t_f, a proper symmetric positive definite P matrix enables optimization over an infinity horizon (Mayne *et al.*, 2000). Thus, we find the largest possible constant α such that $\bar{u} \in K_{mpc}\Omega_a, \forall \bar{e} \in \Omega_\alpha$. With $\Omega_\alpha = \{\bar{e} \in \mathbb{R}^{2n} | \bar{e}^T P \bar{e} \leq \alpha\}$. And α can be computed as proposed in (Chen & Allgöwer, 1998). Considering the value of P obtained from (3.85) as a proper solution, and using Lemma 1 from (Chen & Allgöwer, 1998) we ensure that equation (3.78) satisfies:

$$\dot{L} + I \leq 0 \tag{3.87}$$

Which means that the feedback controller $\bar{u}(t)$ is invariant in the terminal region Ω_α satisfying only input constraints. In other words, once the error $\bar{e}$ gets into Ω_α, it will always stay in this region.

3.5.5 Recursive feasibility

For the recursive feasibility consider the following standard assumptions:

Assumption 5 For the problem (3.76) at time instant t there is a feasible and unique optimal control sequence solution.

$$\mathbf{\Pi}^*(t) = [\Pi^*(t), .., \Pi^*(t_f)] \tag{3.88}$$

And optimal predicted state sequence.

$$\bar{\mathbf{z}}^*(t) = [\bar{z}^*(t), ..., \bar{z}^*(t_f)] \tag{3.89}$$

Assumption 6 For the problem (3.76) at time instant $t + \Delta t$ by construction, there exists, a feasible but not necessarily optimal control sequence solution.

$$\mathbf{\Pi}(t + \Delta t) = [\Pi^*(t + \Delta t), .., \Pi^*(t_f), \bar{u}(t_f + \Delta t)] \tag{3.90}$$

And feasible predicted state sequences

$$\bar{\mathbf{z}}(t + \Delta t) = [\bar{z}^*(t + \Delta t), ..., \bar{z}^*(t_f), \bar{z}(t_f + \Delta t)] \tag{3.91}$$

where Δt is the sampling time. Due to the feasibility of the system (Assumption 5) $\bar{z}^*(t_f) \in z_d(t_f) \oplus \Omega_\alpha$. By (3.87) we know that the terminal region is invariant for the terminal controller $\bar{u}(t_f + \Delta t)$ (Mayne *et al.*, 2011), this indicates that $\bar{z}(t_f + \Delta t) \in z_d(t_f) \oplus \Omega_\alpha$. Recursive feasibility is obtained by induction. Also, it is obvious that the constraints are satisfied, $\bar{\mathbf{z}}(t) \in \mathcal{Z} \ominus \Omega_\alpha$ and $\mathbf{\Pi}(t) \in \mathcal{U} \ominus K_{mpc}\Omega_\alpha$. It can be said that if the problem (3.76) is feasible at the initial time, thus, it is feasible for any $t > 0$ satisfying input and state constraints.

Note that **Assumptions** 5 and 6 indicate that the system's nominal state error $\bar{e}(t)$, must be heuristically initialized at least inside the feasible region as shown in Fig. 3.9. Also, note that the feasible region size depends on the prediction horizon length for robust MPC optimization with terminal constraints. The idea is that the system's nominal state error $\bar{e}(t)$ reaches the terminal region at the end of the prediction horizon to ensure optimization feasibility. Since the terminal controller is able to maintain the nominal state error $\bar{e}(t)$ inside the terminal region for

the following optimization procedures, therefore recursive feasibility is obtained. If the reader would like to initialize the states at any point outside the feasible region, the dual-mode MPC approach can be used (Yang *et al.*, 2021a).

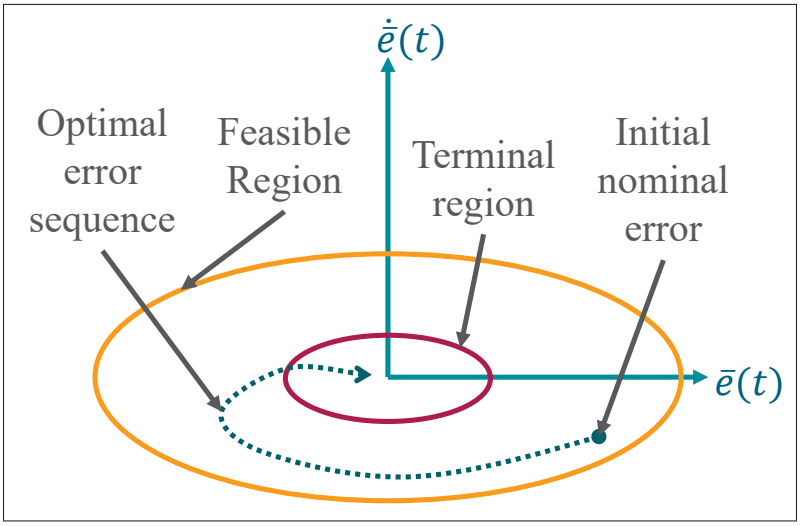

Figure 3.9 Physical implication of **Assumptions** 5 and 6

3.5.6 Closed-loop stability

The cost function (3.78) at time t under $\mathbf{\Pi}^*(t)$ is

$$J(\bar{\mathbf{z}}^*(t), \mathbf{\Pi}^*(t)) = L(\bar{z}^*(t_f)) + \int_t^{t_f} I(\bar{z}^*(\tau), \Pi^*(\tau))d\tau \tag{3.92}$$

And the cost at time $t + \Delta t$ under $\mathbf{\Pi}(t + \Delta t)$ is

$$\begin{aligned} J(\bar{\mathbf{z}}(t + \Delta t), \mathbf{\Pi}(t + \Delta t)) &= L(\bar{z}(t_f + \Delta t)) \\ + \int_{t+\Delta t}^{t_f} I(\bar{z}^*(\tau), \Pi^*(\tau))d\tau &+ \int_{t_f}^{t_f+\Delta t} I(\bar{z}(\tau), \bar{u}(\tau))d\tau \end{aligned} \tag{3.93}$$

The arithmetic difference between (3.93) and (3.92) is defined as:

$$\Delta J(t+\Delta t) = J(\bar{\mathbf{z}}(t+\Delta t), \mathbf{\Pi}(t+\Delta t)) - J(\bar{\mathbf{z}}^*(t), \mathbf{\Pi}^*(t)) \tag{3.94}$$

Expanding (3.94) yields

$$\begin{aligned} \Delta J(t+\Delta t) &= L(\bar{z}(t_f+\Delta t)) - L(\bar{z}^*(t_f)) \\ &+ \int_{t_f}^{t_f+\Delta t} I(\bar{z}(\tau), \bar{u}(\tau))d\tau - \int_{t}^{t+\Delta t} I(\bar{z}^*(\tau), \Pi^*(\tau))d\tau \end{aligned} \tag{3.95}$$

By integrating (3.87) from $\tau = [t_f, t_f + \Delta t]$ considering $\bar{\mathbf{z}}(t+\Delta t)$ yields.

$$\int_{t_f}^{t_f+\Delta t} I(\bar{z}(\tau), \bar{u}(\tau))d\tau + L(\bar{z}(t_f+\Delta t)) - L(\bar{z}^*(t_f)) \leq 0 \tag{3.96}$$

Thus

$$\Delta J(t+\Delta t) \leq -\int_{t}^{t+\Delta t} I(\bar{z}^*(\tau), \Pi^*(\tau))d\tau \leq 0 \tag{3.97}$$

The optimal cost satisfies the following condition (Jadbabaie *et al.*, 2001):

$$J(\bar{z}^*(t+\Delta t), \Pi^*(t+\Delta t)) \leq J(\bar{z}^*(t), \Pi^*(t)) \tag{3.98}$$

Then from (3.97), considering **Assumption 4**, we ensure that $\Delta J(t+\Delta t) \leq 0$ and the closed-loop system is asymptotically stable. And there exist a $\mathcal{K}_\infty$ function $\beta(\cdot,\cdot)$ and $\mathcal{K}$ function $\psi(\cdot)$ (Limon *et al.*, 2009) that satisfy:

$$\|\bar{e}^*(t)\| \leq \beta(\|\bar{e}^*(0)\|, t) + \psi(\omega_2) \tag{3.99}$$

For a sufficient small Δt the measured value of z satisfies $z(t+\Delta t) \in \bar{z}^*(t+\Delta t) \oplus \Omega_{tube}$ implying recursive feasibility and closed-loop stability in the real implementation (Dai *et al.*, 2020). Now, we describe Algorithm 3.1.

Algorithm 3.1 Proposed controller

Offline:
Choose the weighting matrices Q, R and a proper K_{mpc}.
Choose η satisfying the condition $\eta < -\lambda_{max}(A + BK_{mpc})$.
Compute the terminal penalty P by (3.85) and terminal region Ω_α.
Build the set Ω_{tube} from the model mismatch from a previous trial.
Set the total trajectory time T.
Online:
while $(t < T)$ **do**
    Measure the system's states $\theta(t)$ and $\dot{\theta}(t)$.
    Solve the optimal control problem (3.76) and obtain ν_{mpc} (3.82).
    Compute $\nu_{ISM}(t)$ (3.68) and the virtual control law $\nu(t)$ (3.61).
    Calculate $\Gamma(t)$ by (3.58) and apply to the real system (3.56).
    Let $t = t + \Delta t$
end while

3.6 Experimental set-up and results

3.6.1 Real-time system

The real-time system is composed of three processing units (see Fig. 3.10): a real-time PC (NI PXI-8108), FPGA (NI PXI-7813R), and a Host PC. The real-time PC has an Intel dual-core @2.53 GHz processor, and 8 GB of RAM, where the proposed HIK, the proposed controller, and the dynamics of the 7-DOF Robotic Exoskeleton ETS-MARSE are executed. The PXI-7813R's FPGA is used for the analog and digital inputs and outputs to the actuators and sensors (hall effect position sensor and current sensor). The Host PC has an Intel Core i7-4770 CPU @3.4 GHz, and 16 GB installed RAM where the GP training and discomfort index weight tuning were performed, also for the real-time tests the Host PC is used as the user interface, for storing and displaying the results from completed trials.

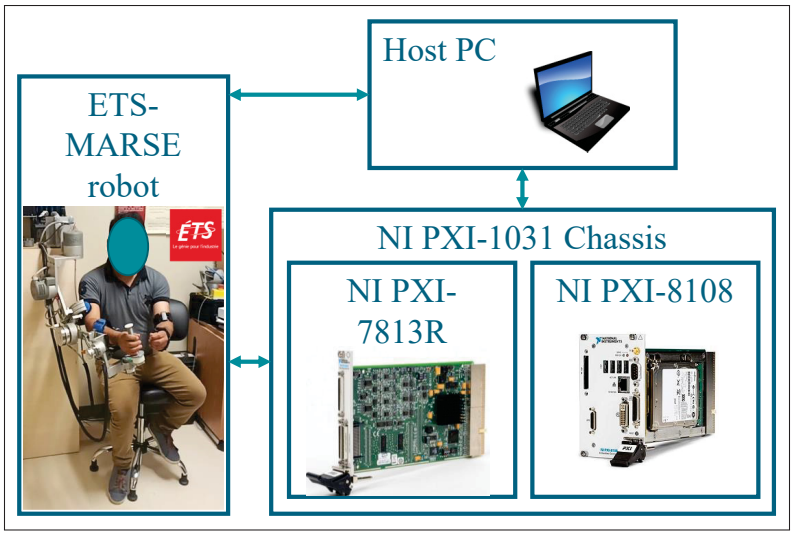

Figure 3.10 Real-time system

3.6.2 Swivel angle estimation details

The GP training and the discomfort index weights tuning procedure were executed offline in the Host PC, the details are shown in Table 3.2. Both procedures start with 100 random initial model guesses to avoid local minima models, then by evaluating their performance the best 4 models are chosen (elite) and optimized using 500 gradient descent iterations. Finally, the model with the smaller RMS validation error (the model with the best generalization) is chosen as the final model for comparison and its RMS test error is computed (See Table 3.2). For both the GP and discomfort index approaches, the length of the training sets are chosen differently due to the nature of the approaches, because the number of training points defines the GP's complexity, it was trained choosing $p = 20$ random points taken from the training data sets (3.38) (see Fig. 3.7 a)). On the other hand, the discomfort index weight tuning (see Fig. 3.6 a)) was performed with $p = 300$ data points from the training data (3.38). Thus, the tuned weights are as follows:

$$E_1 = [26.5, -3.9, -46.9, 84.6, 90.7, 199.2, 40.6]^T$$

$$E_2 = 762.5, \quad E_3 = 0.2N \cdot rad, \quad \psi_n = 0.5rad$$

Table 3.2 Offline training details

	Discomfort index approach	**Gaussian process approach**
Initial number of models	100	100
Number of models to optimize (elite)	4	4
Optimizer	Gradient descent	Gradient ascent
Cost function	RMS error	Log marginal likelihood
Parameters to optimize	10	8
Number of iterations	500	500
Training set length	300	20
Validation set length	300	300
Test set length	4300	4300
Offline training time (sec)	1623.23	58.64
RMS validation error ($\sqrt{rad}$)	0.431	0.1134
RMS test error ($\sqrt{rad}$)	0.4129	0.11725

And the chosen parameters are:

$$\theta_m = [5, 5, 0, 90, 0, 0, 0]^T, m_e = 5kg$$

The online implementation details are shown in Table 3.3, the methods were executed in

the real-time PC previously described. For online implementation of the Jacobian approach and discomfort index, for each task-space target, we set 30 iterations maximum with stopping conditions. Note that for the first task-space target the initial solution guess is provided by the authors but for the next task-space targets the previous solution is used because it might be closer to the new optimal solution. Then, in some cases the stopping condition can be reached before the maximum number of iterations are completed, reducing computation time. Therefore, the online computation time (Table 3.3) is an average approximation in both Jacobian and discomfort index approaches. The stopping condition for the Jacobian approach of $1e-15$ meters is chosen because the swivel angle approach (which is used in the discomfort index and GP approaches) achieves a similar Cartesian precision for any given swivel angle. And the stopping condition of the discomfort index approach is chosen as $1e-6$ radians because this precision is enough for swivel angle comparison.

We have compared the swivel angle estimation approaches described in Section 3.4, to evaluate their performance in three designed upper-limb motions a) Pick and place, b) Diagonal reaching, and c) Natural elbow flexion/extension (which represent unseen data) as shown in Fig. 3.11.

3.6.3 Controller experiment details

For trajectory tracking comparison, two sets of tests were performed on the ETS-MARSE robotic exoskeleton in real-time for the three upper-limb motions shown in Fig. 3.11. The first set of tests was performed without a subject but in the presence of model uncertainties due to unmodeled dynamics. The second set of tests was performed with three different subjects who apply a changing force in the gravity direction using the forearm. The controllers used for the tests without a subject are the conventional SM used in (Fallaha *et al.*, 2011), the robust MPC without the ISM, and the proposed controller (robust MPC with ISM). Due to high chattering, the conventional SM was excluded from the tests with subjects. The feedback linearization technique was performed using the nominal physical parameters of the ETS-MARSE that can be found in (Fallaha *et al.*, 2020). Note that the controllers do not consider the physical characteristics of the subjects which are shown in Table 3.4. We consider the design parameters

Table 3.3 Online implementation details

	Jacobian approach	**Discomfort index approach**	**Gaussian process approach**
Initial condition	$[5\ 5\ 0\ 90]^T$ or previous solution	0 or previous solution	NA
Optimizer	Newton-Raphson	Gradient descent	NA
Number of iterations	30	30	NA
Stopping condition	$\|\|e_w\|\| < 1e-15$	$\delta_\psi < 1e-6$	NA
Optimization coefficient	$\alpha_w = 0.01$	$\alpha_\psi = 0.1$	NA
Average computation time (ms)	0.810	1.437	0.042

Table 3.4 Physical characteristics of three subjects

Subject	Sex	Height	Weight	Age
subject-1	Male	160 cm	75 kg	31 y/o
subject-2	Female	163 cm	63 kg	25 y/o
subject-3	Male	165 cm	87 kg	41 y/o

of the conventional SM and ISM shown in Table 3.5. For the robust MPC, a prediction horizon of 10 steps with $\Delta t = 1\ ms$ was used. The robust MPC's terminal region size is specified by

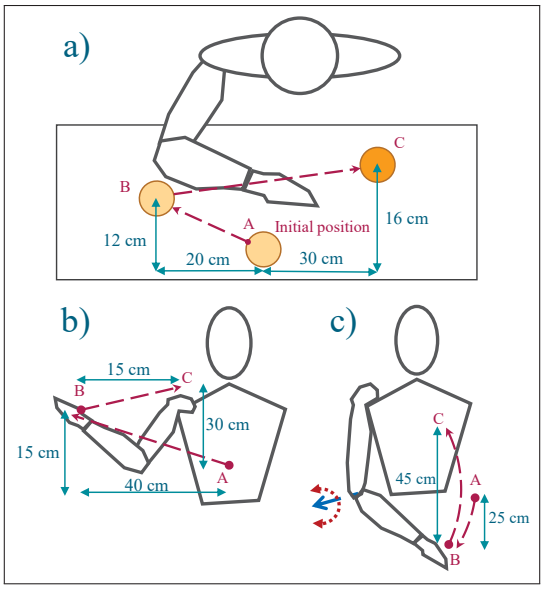

Figure 3.11 Three upper-limb motions used for swivel angle estimation: a) Pick and place, b) Diagonal reaching, and c) Natural elbow flexion/extension

$\alpha = 1.2$. The weighting matrices Q and R are:

$$Q = diag(660, 3100, 100, 2500, 650, 50, 10, 1, 1, 1, 1, 1, 1, 1)$$

$$R = 0.01 I_{7\times 7}$$

Choosing the following K_{mpc}.

$$K_{mpc} = [-K_{mpc1}, -K_{mpc2}]$$

$$K_{mpc1} = diag(253, 547, 99, 492, 252, 70, 31)$$

$$K_{mpc2} = diag(30, 33, 14, 32, 23, 12, 8)$$

Table 3.5 SM design parameters

	Conventional SM (Fallaha *et al.*, 2011)	Proposed controller
k_i	$diag([600\ 600\ 600\ 200\ 800\ 200\ 100])$	$diag([20\ 25\ 25\ 15\ 50\ 45\ 35])$
λ_i	$diag([32\ 26\ 13\ 38\ 13\ 5\ 4])$	-
p_i	-	1
α_i	-	20
δ_i	-	0.1
Ξ	-	$[0_{7\times7}, I_{7\times7}]$
Online computation time (ms)	0.273	0.387

By the constant $\eta = 18.61$ satisfying $\eta < -\lambda_{max}(A + BK_{mpc})$, we obtain the following elements of the matrix P by solving (3.85).

$$P_{11} = diag(662, 879, 600, 831, 661, 594, 589)$$

$$P_{12} = P_{21} = diag(15.9, 18.9, 14.9, 18.2, 15.9, 14.8, 14.8)$$

$$P_{22} = diag(0.78, 0.83, 0.77, 0.82, 0.78, 0.77, 0.77)$$

The torque constraints in Nm units are:

$$\Gamma_{max} = [25.38\ 50.76\ 30.60\ 17.62\ 15.93\ 6.37\ 6.37]^T$$

$$\Gamma_{min} = -1 \cdot [25.38\ 16.92\ 30.60\ 17.62\ 15.93\ 6.37\ 6.37]^T$$

The state constraints in deg and deg/sec units are:

$$\theta_{max} = [70\ 120\ 80\ 115\ 85\ 20\ 60]$$

$$\dot{\theta}_{max} = [\ 100\ 100\ 100\ 100\ 100\ 100\ 100]$$

$$z_{max} = [\theta_{max}\ \dot{\theta}_{max}]^T$$

$$\theta_{min} = -1 \cdot [20\ 5\ 85\ 5\ 85\ 25\ 55]$$

$$z_{min} = [\theta_{min}\ \ -\dot{\theta}_{max}]^T$$

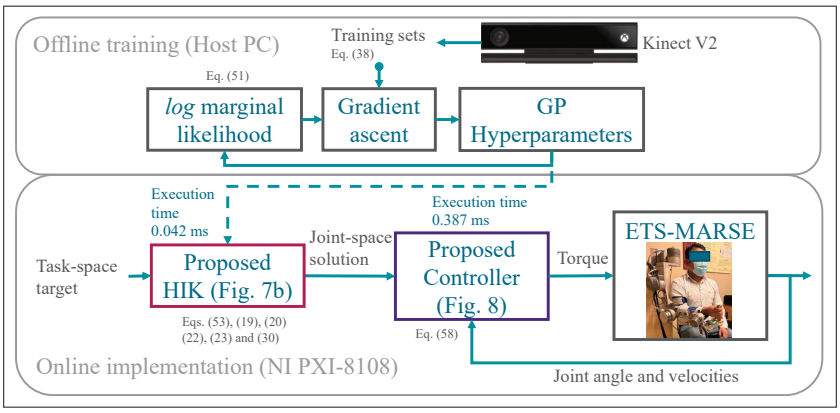

Figure 3.12 Block diagram of the proposed approaches

3.6.4 Results and discussion

The overall proposed scheme shown in Fig. 3.12 requires 58.64 seconds in the offline training for 500 iterations each with 117.28 ms computation time. And the online computations require a total time of 0.429 ms where 0.042 ms are from the proposed HIK and 0.387 ms are from the proposed controller. For the proposed HIK part, the swivel angle, the wrist position, and orientation were measured for each upper-limb motion in Fig. 3.11. Using the previous data, the RMS estimation error with respect to the measured swivel angle was computed, and compared for each estimation method (See Table 3.6). Considering the obtained results, the

Table 3.6 Comparison results: RMS estimation error (rad)

	Jacobian approach	Discomfort approach	Proposed approach
Pick and place	0.5082	0.3506	0.2154
Diagonal reaching	0.7263	0.56	0.3970
Elbow Flexion/extension	0.3731	0.2244	0.1151

proposed swivel angle estimation method shows a lower RMS error with respect to measured data, implying more natural postures, therefore reducing the uncoordinated movements between wearer and exoskeleton as demonstrated in (Kim *et al.*, 2012a). The main disadvantage of the parameter tuning process for the discomfort index approach is that it requires a lot of effort to find appropriate weight values for different users and for different tasks as the offline computation time shows in Table 3.2. Also, another disadvantage of the Jacobian and Discomfort index approaches is the online computation time shown in Table 3.3, which are intractable for real-time implementation.

A comparison with several related works using a learning approach for swivel angle estimation is shown in Table 3.7 where the obtained online calculation time verifies the low computational burden of the proposed HIK approach against models which are computationally complex or are based on optimization methods.

For the proposed controller part, the mean of the seven joint's RMS tracking errors are shown in Table 3.8 (without a subject) and Table 3.9 (with three subjects). We include several figures for subject-1 to show the performance of the robust MPC with and without ISM considering the b) upper-limb motion in Fig. 3.11. The Cartesian tracking (See Fig. 3.13) shows a smaller Cartesian error for the proposed controller. The trajectory tracking performance is shown in

Table 3.7 Learning approaches comparison

Reference	Number of DoF	Swivel estimation time (ms)	Number of validation tests	Controller used
(Su *et al.*, 2018)	7	0.5	2	Not specified
(Lauretti *et al.*, 2018)	5	0.4	3	PID
(Su *et al.*, 2020)	7	80	1	Not specified
This Paper	7	0.042	3	Robust MPC with ISM

Fig. 3.14 where multiple joints are active during the test. The tracking error is shown in Fig. 3.15 where the proposed controller shows a higher precision which is not strictly necessary for rehabilitation exercises but important for robotic systems to maintain stability and tracking accuracy in the presence of unknown and unexpected external disturbances. The control activity is shown in Fig. 3.16 where the force applied by subject-1 directly affects joint 4, where a higher control activity is obtained to reject the external disturbances. Furthermore, our analysis includes several figures illustrating the test results without a subject. Figure 3.17, 3.18, and 3.19 clearly demonstrate the inadequate tracking performance of the conventional SM. In addition, Figure 3.20 reveals the presence of chattering during the test. It is worth noting that in order to improve the tracking performance using the conventional SM, higher switching gains would need to be selected. However, this would lead to an unacceptable level of chattering for the robot's motors. Note that the robust MPC allows the choice of a smaller switching gain k_i for the proposed controller, improving the tracking and reducing the chattering without losing stability.

Table 3.8 RMS tracking error without Subject
$(1 \times 10^{-4} rad)$

	Pick and Place	Diagonal reaching	Elbow Flexion/ Extension
Conventional SM	2109	2979	2202
Robust MPC	158	156	169
Proposed controller	88	94	109

Table 3.9 RMS tracking error with three subjects
$(1 \times 10^{-4} rad)$

	Pick and Place			Diagonal reaching			Elbow Flexion/ Extension		
Subject number	1	2	3	1	2	3	1	2	3
Robust MPC	264	210	274	262	205	276	332	271	339
Proposed controller	103	95	107	110	98	116	149	120	148

3.7 Conclusions

In this paper, a comparison between the Jacobian approach, the discomfort index approach, and the Gaussian Process for swivel angle estimation has been done. The proposed approach shows better performance estimating the swivel angle from three upper-limb motions (not included in the training data (3.38)), where the position and orientation of the wrist are considered. The proposed swivel angle estimation approach has a lower complexity compared to existing learning methods, achieving real-time accurate and unique human inverse kinematic solutions. By taking advantage of the obtained GP's low computational complexity, as future work, we will

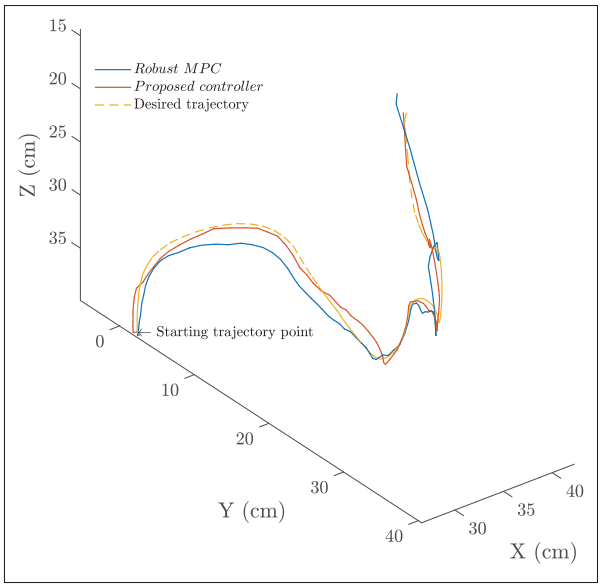

Figure 3.13 Cartesian tracking comparison
(Diagonal reaching with subject-1)

propose an approach to generate safer trajectories for upper-limb rehabilitation by considering the computed uncertainty in the GP output. In addition, a robust MPC with ISM is proposed for trajectory tracking ensuring constraint satisfaction with high accuracy while obtaining a trade-off between tracking performance and control effort. A set of experimental results considering three different upper-limb motions without a subject, and with three different subjects confirms the proposed controller's effectiveness. Also, as future work, we will investigate the performance of the GP technique to enhance the nominal prediction model in the robust MPC to improve optimality in the trajectory tracking using the 7 DoF exoskeleton ETS-MARSE, where the high dimensionality will be an arduous challenge.

Acknowledgments

The first author would like to thank to CONAHCYT (Mexican National Council of Humanities, Science and Technology) for the support under grant 739833. CONAHCYT was previously named CONACYT (Mexican National Council of Science and Technology).

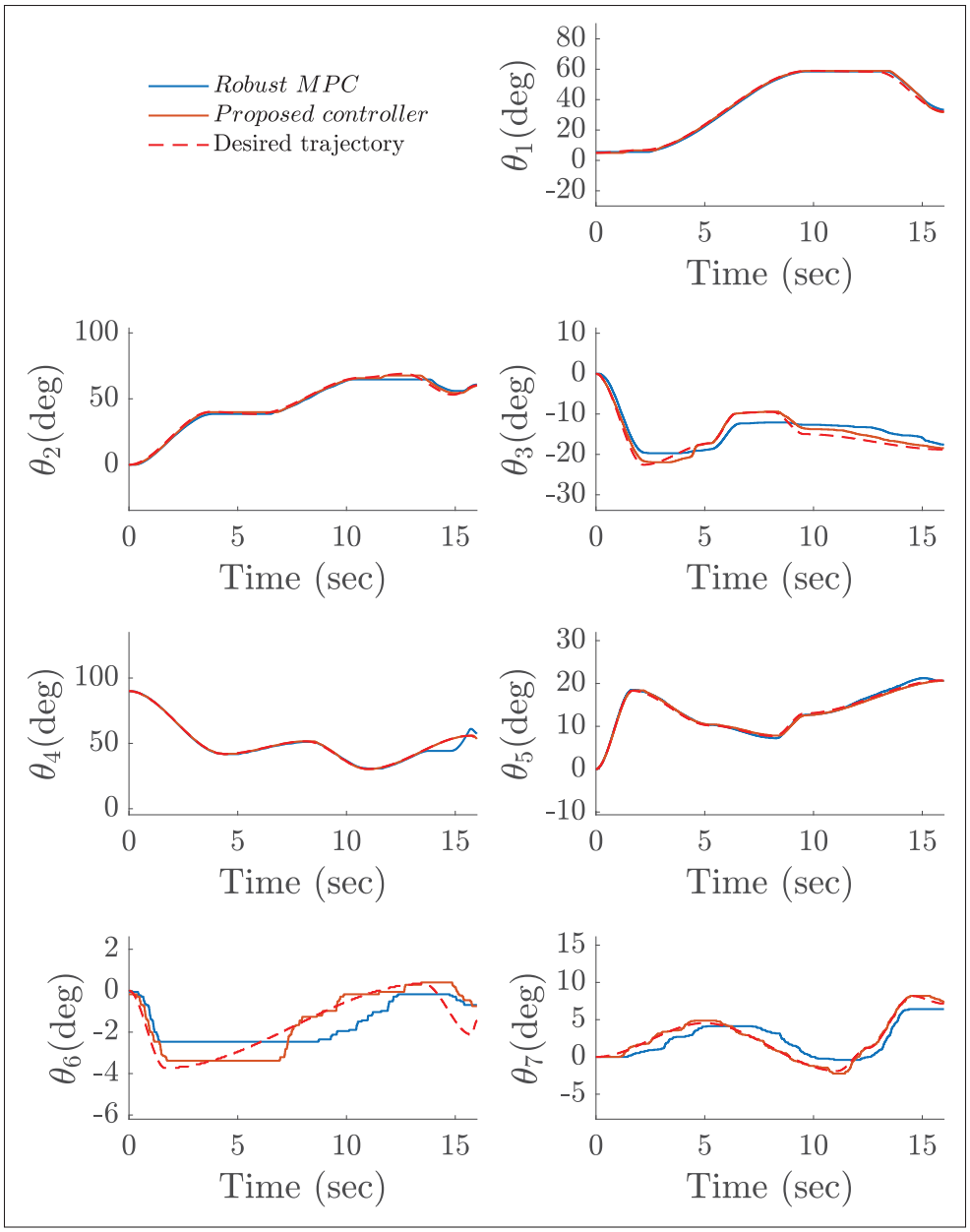

Figure 3.14 Trajectory tracking comparison (Diagonal reaching with subject-1)

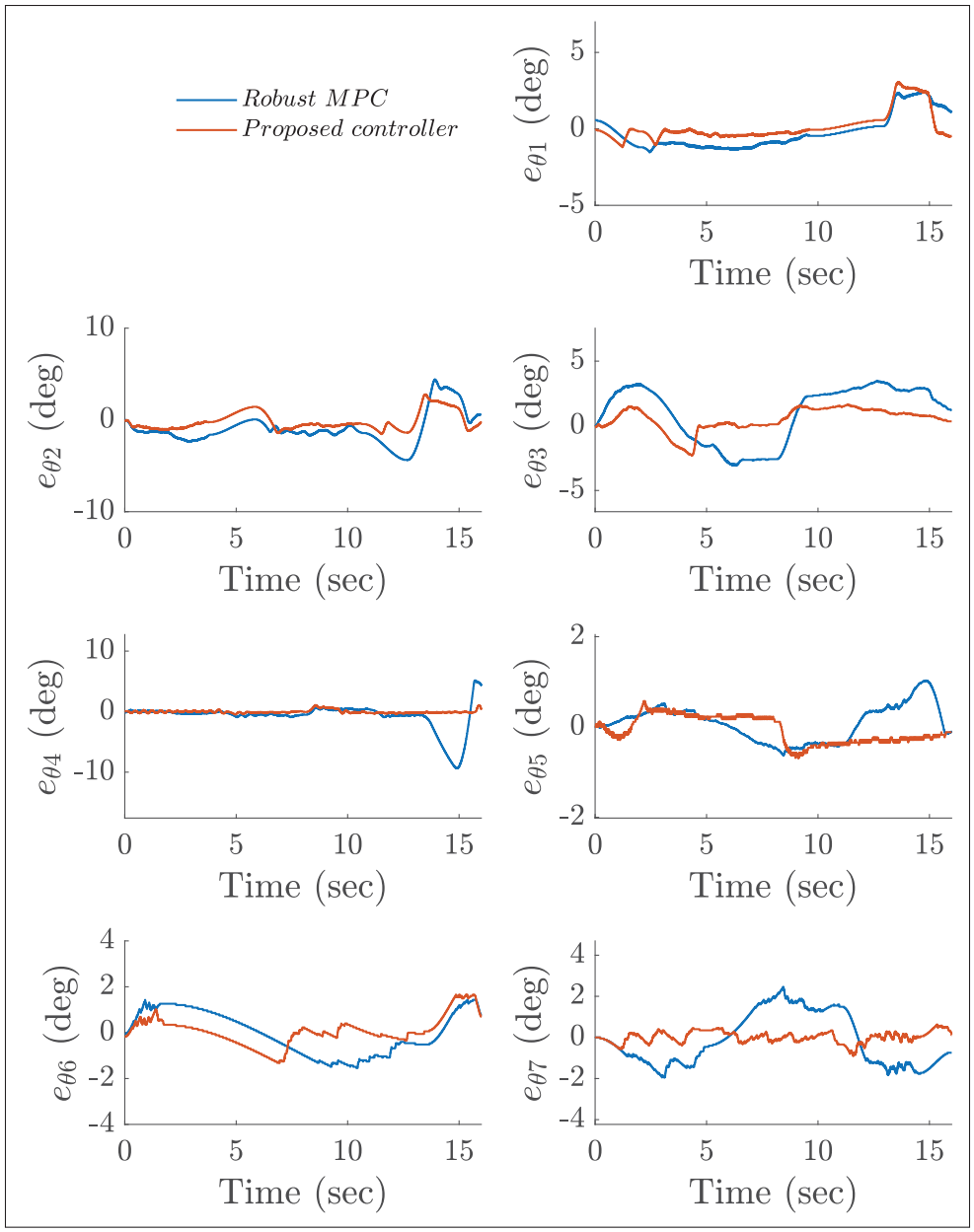

Figure 3.15 Tracking error comparison (Diagonal reaching with subject-1)

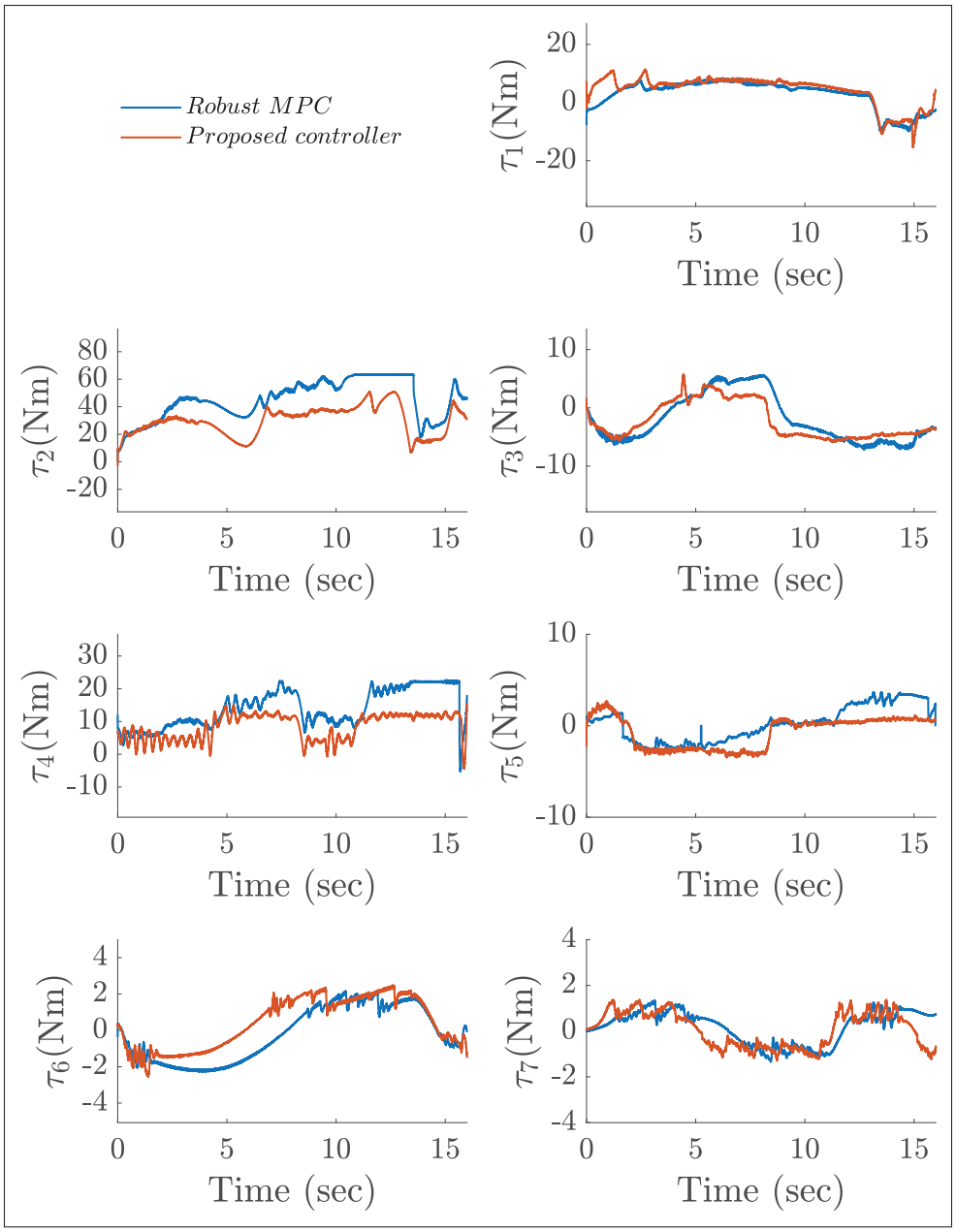

Figure 3.16 Control signal comparison (Diagonal reaching with subject-1)

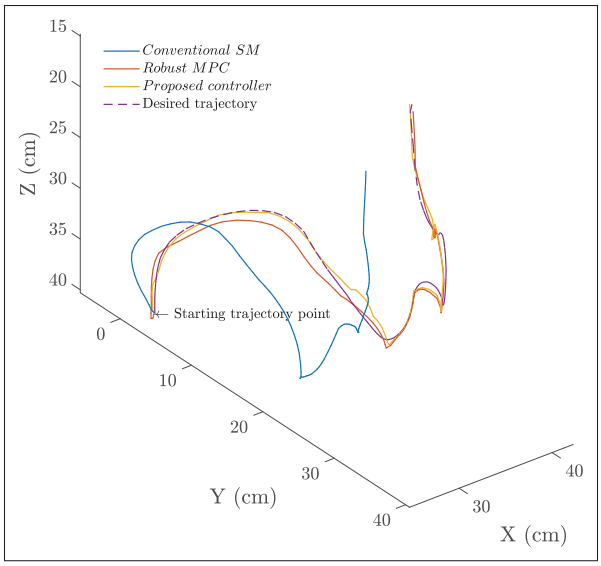

Figure 3.17 Cartesian tracking comparison (Diagonal reaching without a subject)

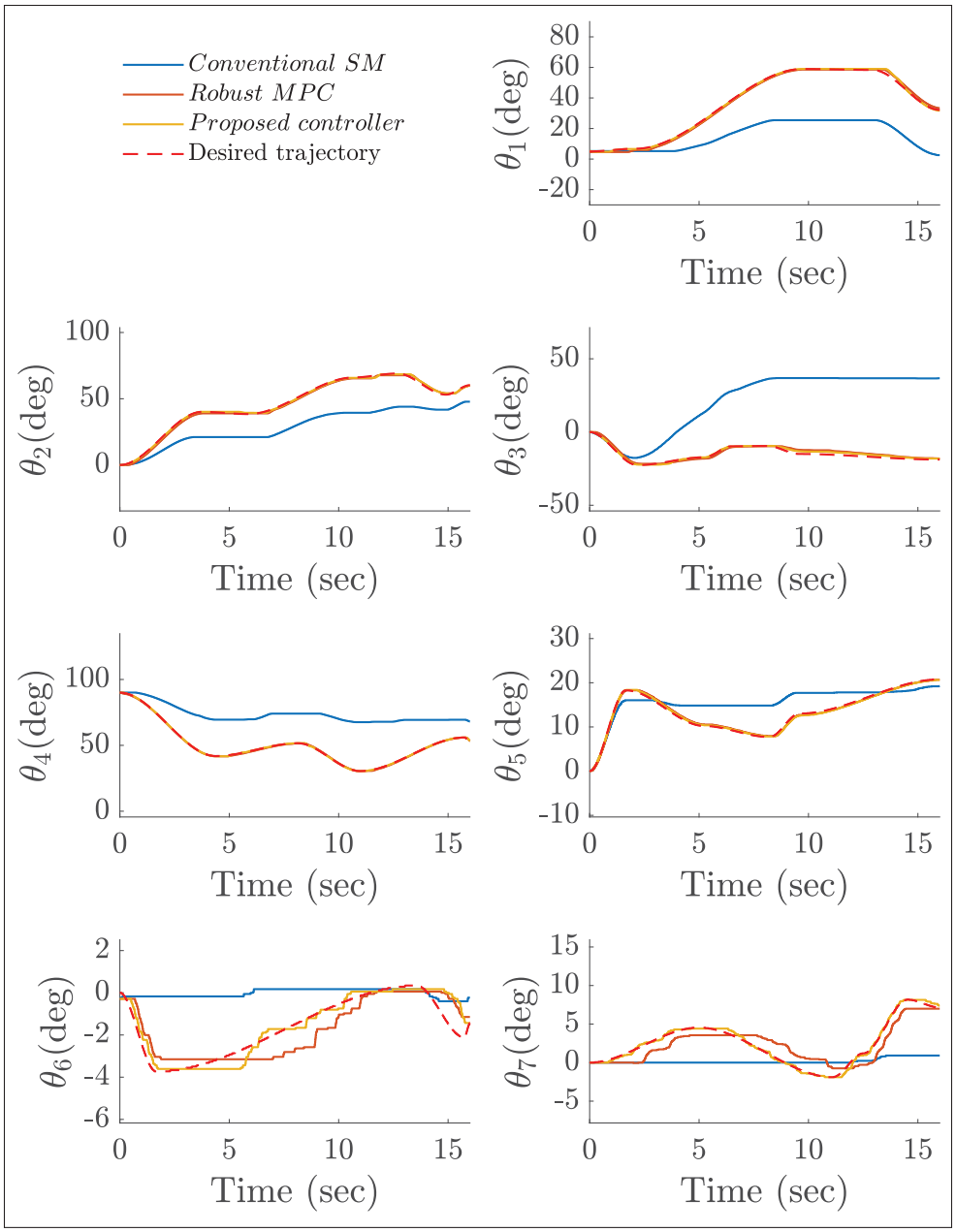

Figure 3.18 Trajectory tracking comparison (Diagonal reaching without a subject)

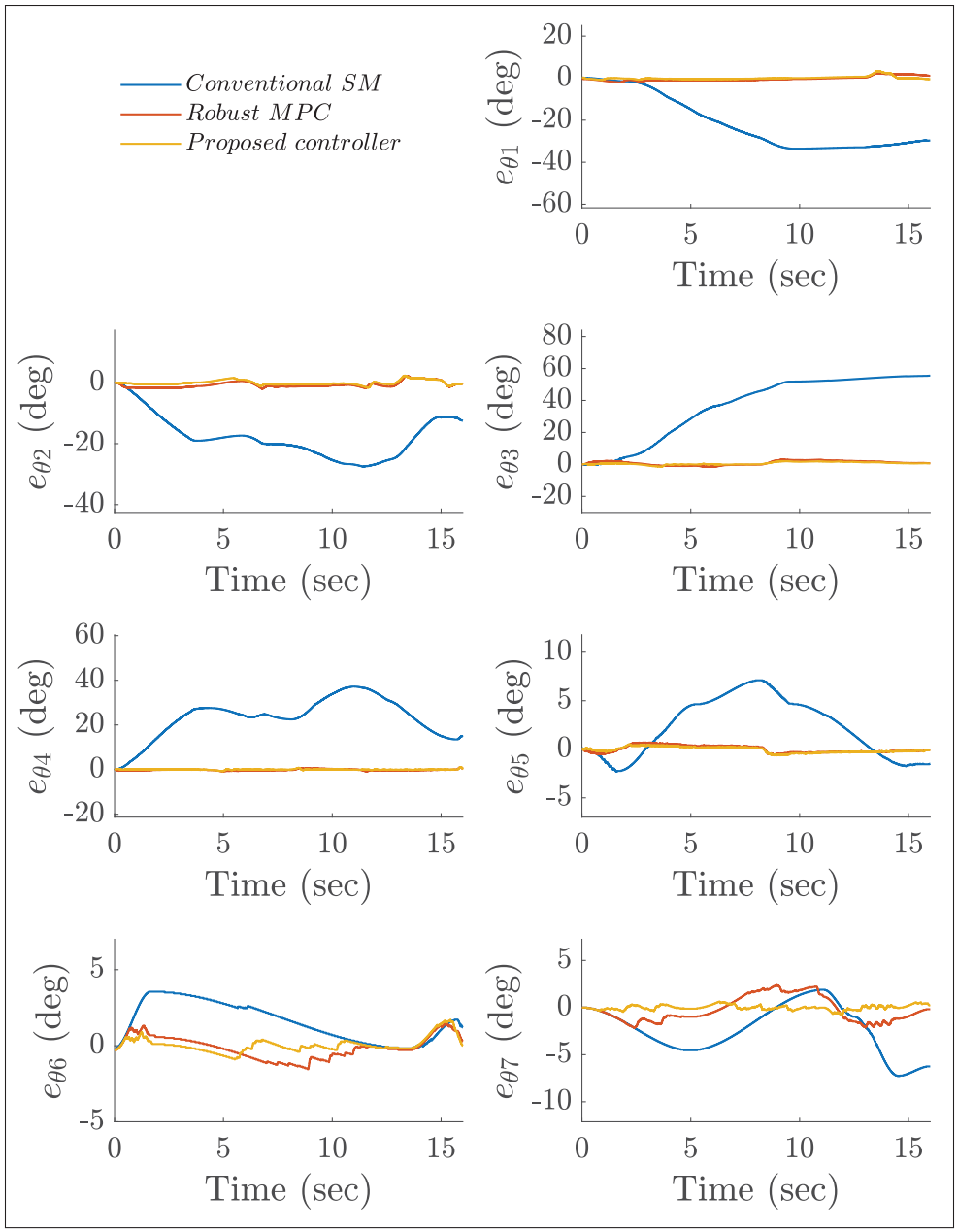

Figure 3.19 Tracking error comparison (Diagonal reaching without a subject)

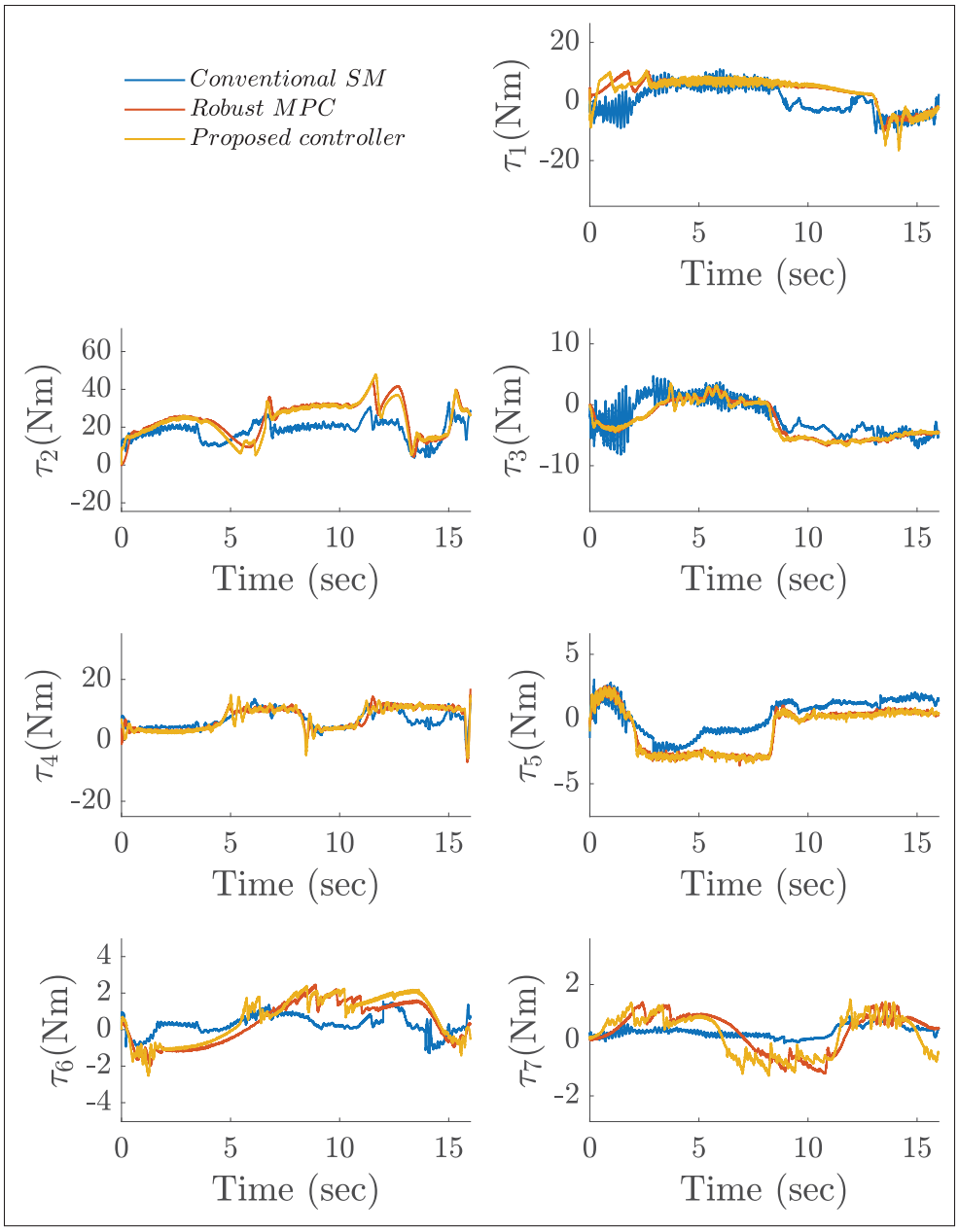

Figure 3.20 Control signal comparison (Diagonal reaching without a subject)

CHAPTER 4

ROBUST LEARNING-BASED NONLINEAR CONTROL FOR UPPER-LIMB ROBOTIC REHABILITATION

David Bedolla-Martinez[1] , Yassine Kali[1] , Maarouf Saad[1] , Cristobal Ochoa-Luna[2] , Mohammad Habibur Rahman[3]

[1] Département de Génie Mécanique, École de Technologie Supérieure, 1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

[1] Tecnologico de Monterrey, School of Engineering and Sciences, Ave. Eugenio Garza Sada 2501, Monterrey, N.L. México, PC: 64849

[1] Mechanical and Biomedical Engineering Department, University of Wisconsin-Milwaukee, Milwaukee, WI, USA, PC: 53211

Paper submitted to the *ISA Transactions*, September 2023.

4.1 Introduction

Wearable robots, such as exoskeletons, are becoming increasingly popular in the field of robotics. These complex electromechanical devices are designed to be compliant and provide physical training to the biomechanical structure of a human limb. Their primary objective is to aid individuals in their everyday tasks, and they have been demonstrated to serve as a highly efficient substitute for conventional physical therapy in medical rehabilitation (Barrios-Muriel *et al.*, 2020). This is especially beneficial due to the lack of health centers and professionals available (Teasell *et al.*, 2008).

However, passive robotic rehabilitation, where the exoskeleton fully guides the impaired arm, is a difficult task. During a rehabilitation task to ensure stability or avoid a decrease in performance, the feedback controller must be able to manage a series of challenges. These include 1) unmodeled dynamics of highly nonlinear redundant exoskeletons, 2) unexpected and unknown perturbations from the interaction between the patient and robot, 3) various limb spasticity levels, 4) different physiological characteristics (weights, heights, and limb lengths) of different wearers, and 5) changing operational circumstances, such as exercises performed at different velocities.

In the literature, it has been noted that the PID controller is ill-suited for upper-limb rehabilitation tasks because of the pronounced unknown dynamics inherent in exoskeleton dynamics (Yu & Rosen, 2013). In response to this challenge, several nonlinear control methodologies have emerged and been applied to robotic exoskeletons. These include time delay estimation (TDE) (Fei *et al.*, 2017), computed torque control (CTC) (Brackbill *et al.*, 2009), backstepping (Brahmi *et al.*, 2016) and sliding mode control (SMC) (Rahmani & Rahman, 2019). However, these control techniques lack an inherent safety validation process, where the performance can deteriorate when factoring in input and state constraints (Dai *et al.*, 2020). To achieve both safe operation and optimal performance in the context of upper-limb rehabilitation while adhering to constraints. Thus, the current paper advocates for the adoption of a model predictive control (MPC) (Mayne *et al.*, 2000).

While the conventional MPC controller encounters challenges in ensuring the stability of highly uncertain nonlinear systems (Grüne *et al.*, 2017), the literature presents two robust MPC methodologies: Min-Max (Scokaert & Mayne, 1998) and Tube MPC (Langson *et al.*, 2004). It is worth noting that Min-Max, although effective, suffers from computational inefficiency, whereas Tube MPC bears a computational burden comparable to conventional MPC. However, Tube MPC adopts a more conservative stance by prioritizing stability over optimality. Despite these advancements, robust MPC still grapples with certain unresolved issues. These include **1.** the computational load associated with longer prediction horizons required to satisfy terminal constraints and **2.** the imperative need for an accurate prediction model to enhance optimality.

1. For computation burden reduction, the combination of MPC with SMC has been proposed (Zhou *et al.*, 2001). Beyond adding robustness, the SMC brings additional benefits to the MPC. The SMC approach enables prediction horizon length shortening obtaining a tractable Nonlinear MPC (Jafari Fesharaki *et al.*, 2020). The conventional sliding mode control (SMC) serves to reject unknown disturbances and minimize the disparity between the measured states and the nominal model prediction (addressing model uncertainty), thereby enhancing the conservatism in MPC (Spasic *et al.*, 2016). Furthermore, when Integral Sliding Mode (ISM) is combined with MPC, it adapts itself to address model uncertainty (Rubagotti *et al.*, 2009) and (Incremona *et al.*,

2017). The fusion of SMC and MPC effectively constrains model uncertainty in the context of rehabilitation tasks (Bao *et al.*, 2020), and improves the computation burden by obtaining a linear plant with reduced model uncertainty through feedback linearization (Rubagotti *et al.*, 2010). Besides the conventional SMC approach, other versions to reduce the chattering while improving the performance are: exponential reaching law (ERL) (Fallaha *et al.*, 2011) for faster convergence and super-twisting for accuracy in the trajectory tracking (Kali *et al.*, 2018).

2. For optimality improvement, an accurate prediction model is needed for MPC. For exoskeleton robots, the dynamic model's complexity grows drastically with a few DOF and it is a problem while solving numerical optimization in real-time. Thus, the accuracy of the prediction model had been addressed by nonlinear models (Xie *et al.*, 2021), using Taylor approximation (Hedjar *et al.*, 2005), and fixed-state approximation (Terry *et al.*, 2017). Another approach based on machine learning methods can build accurate models using data from previous trials. Besides nominal machine learning models, e.g., recurrent neural networks (Wang *et al.*, 2021), probabilistic methods assign an uncertainty value to each predicted state, making it feasible to deal with insufficient data (Hewing *et al.*, 2020b). This is the case for the Gaussian process (GP), a regression method recently used to build accurate prediction models for robots. However, the computation burden for large training data sets, shrinks the application area. On its own, the GP is not better than the usual prediction techniques (Maiworm *et al.*, 2018). As a result, using the GP as an additive model (disturbance model) to match for uncertainties has given good results (Ostafew *et al.*, 2016a). This approach is called Learning-based MPC and an extensive review can be found in (Hewing *et al.*, 2020a). Researchers have previously proven an improvement in the tracking performance from trial to trial by learning from collected data (Ostafew *et al.*, 2016a), (Ostafew *et al.*, 2016b).

In order to ensure optimal performance of the ETS-MARSE in rehabilitation tasks while external disturbances are produced from wearers, in this paper, a hierarchical approach inspired by (Incremona *et al.*, 2017) as shown in Fig. 4.1 is proposed. In this architecture, and thanks to feedback linearization, the highly uncertain nonlinear system is converted to an equivalent uncertain linear one. Then, theoretically using a high gain, the ISM fully rejects the effects

of the unknown disturbances. Nevertheless, the chattering problem will appear which is an undesirable behavior that can damage the motors. To avoid this problem, lower switching gains are used to attenuate the unknown disturbances, resulting in a linear system with reduced model uncertainty. The remaining uncertainty is modeled using an additive stochastic model (disturbance model) to enhance the accuracy of the nominal prediction model, thus improving optimality. As a result, the ISM rejects unknown disturbances while reducing the control activity and the Learning-based MPC ensures the optimal trajectory tracking of the uncertain system subject to input and state constraints. The Learning-based MPC's closed-loop stability is proved by a proper positive invariant region to ensure feasibility. The proposed combination of Learning-based MPC with the ISM, is designed to ensure robustness, optimal performance, and safe operation in the trajectory tracking problem for highly nonlinear systems as the 7-DOF exoskeleton redundant robot that has dynamic and kinematic uncertainties.

To the best of our knowledge, the original contributions are:

1. Develop a Learning-based nonlinear control to enhance optimality from trial to trial in the ISM and MPC combination in the presence of input and state constraints, which existing controllers don't consider on our robotic exoskeleton, thus, exploiting its capabilities for upper-limb rehabilitation.
2. Ensure optimal performance and safe operation with the proposed controller even when there are unknown and unexpected interaction forces between the robot and wearer, this is done by proposing a tractable and constrained optimal control problem.
3. Provide a framework for a tractable Learning-based MPC and thus usable in practice. This is confirmed by performing several tests on an upper-limb robotic exoskeleton. To our knowledge, this represents the inaugural hardware implementation of the Learning-based MPC using the GP model combined with the ISM on a highly nonlinear system as the ETS-MARSE robotic exoskeleton. Furthermore, the suggested approach is not limited to frameworks with terminal inequalities. it is also applicable and can be implemented without the inclusion of terminal constraints e.g. (Limón *et al.*, 2006).

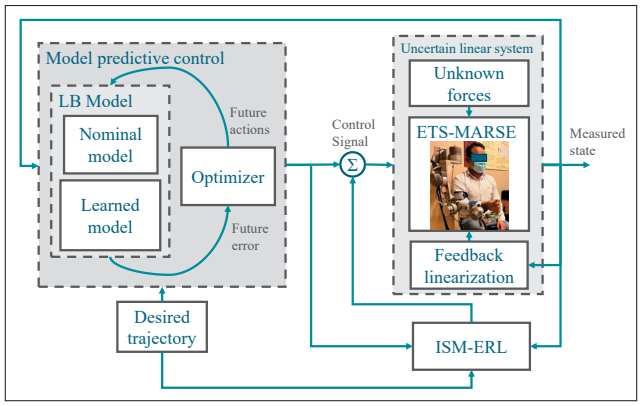

Figure 4.1 Proposed Learning-based MPC controller

This paper is organized as follows. Section 4.2 gives the preliminaries of the work as the notations and definitions used. In addition, the dynamic and kinematic modeling of ETS-MARSE and the feedback linearization are presented. In Section 4.3, the ISM is designed and the closed-loop stability of the system under the proposed controller is provided, the learning-based approach and the learning-based optimal control problem are defined. Also, the feasibility of the learning-based optimization and the closed-loop stability are analyzed by the terminal constraint approach. In section 4.4, the hardware setup, the controller setting, and the proposed approach's experimental results with a healthy subject are shown and compared with two robust controllers. In section 4.5, the discussion is developed. Finally, in Section 4.6, the conclusions are summarized.

4.2 Preliminaries

4.2.1 Notations and Definitions

Let $I_{n\times n} \in \mathbb{R}^{n\times n}$ be the $n \times n$ size identity matrix. $z(t)$ is the measured state z at time t. The predicted values of the state z at a future time τ are defined by $\tilde{z}(\tau)$ and $\bar{z}(\tau)$ using a learned model and a nominal linear model respectively. A function of class c^2 is continuous and its two first derivatives exist. A $\mathcal{K}$ function $\beta(\cdot)$ is strictly increasing and satisfies $\beta(0) = 0$. A $\mathcal{K}_\infty$

function $\beta(\cdot)$ is a $\mathcal{K}$ function and it is unbounded. A $\mathcal{KL}$ function $\psi(\cdot, t)$ is $\mathcal{K}_\infty$ and satisfies $lim_{t\to\infty}\psi(\cdot, t) = 0$.

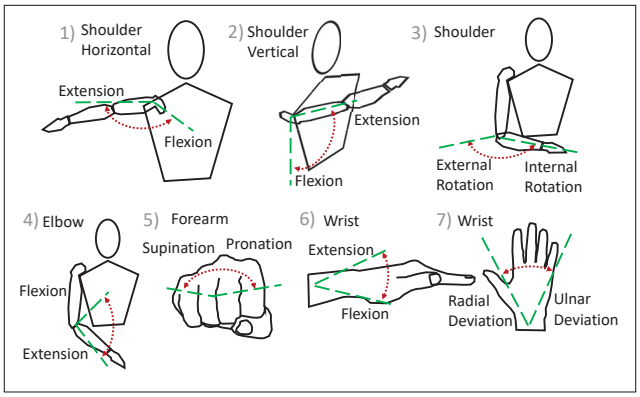

Figure 4.2 Range of motion

4.2.2 Exoskeleton ETS-MARSE

The ETS-MARSE is a 7-DOF highly nonlinear robot able to perform passive, assistive, and active exercises to an upper limb (Rahman, 2012) as shown in Fig. 4.2. To provide safety to the wearer, each joint has a lower and upper bound (see Table 4.1). Based on the placement of the frames shown in Fig. 4.3, the modified Denavit-Hartenberg (D-H) parameters were obtained (see Table 4.2).

Table 4.1 ETS-MARSE Workspace

Joints	**Arm joint**	**Range** (deg)
1	Shoulder	0 to 140
2	Shoulder	140 to 0
3	Shoulder	-85 to 75
4	Elbow	120 to 0
5	Forearm	-85 to 85
6	Wrist	-30 to 20
7	Wrist	-50 to 60

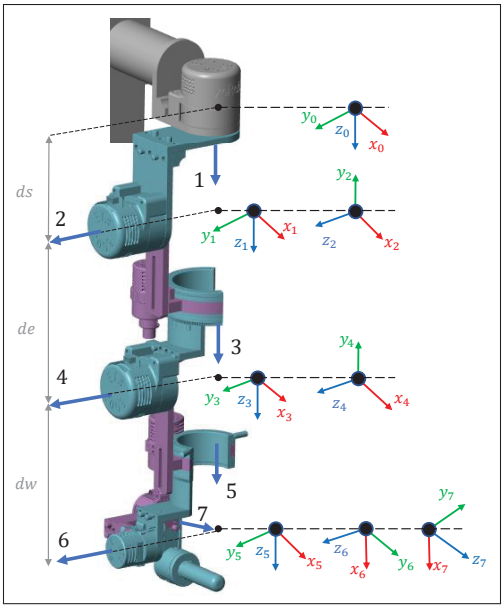

Figure 4.3 Link frame attachment

4.2.3 Feedback Linearization

The goal of this part is to obtain from the nonlinear exoskeleton model a decoupled equivalent linear one. The ETS-MARSE's dynamic model is written in the following differential equation form:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) = \Gamma(t) + \Gamma_{dis} \tag{4.1}$$

Table 4.2 Modified DH Parameters

$\alpha_{i-1}\ (rad)$	$a_{i-1}\ (m)$	$d_i\ (m)$	$q_i\ (rad)$
0	0	d_s	q_1
$-\pi/2$	0	0	q_2
$\pi/2$	0	d_e	q_3
$-\pi/2$	0	0	q_4
$\pi/2$	0	d_w	q_5
$-\pi/2$	0	0	$q_6 - \pi/2$
$-\pi/2$	0	0	q_7

where $q \in \mathbb{R}^n$ is the angular position vector, with $n = 7$ DOF, $\dot{q} \in \mathbb{R}^n$ is the joint angular velocity, $\Gamma(t) \in \mathbb{R}^n$ is the torque applied by the motors, $\Gamma_{dis} \in \mathbb{R}^n$ are unknown forces, $M(q) \in \mathbb{R}^{n \times n}$ is the positive definite inertia matrix, $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ is the Coriolis matrix, $G(q) \in \mathbb{R}^n$ is the gravitational vector, and $F(\dot{q}) \in \mathbb{R}^n$ is the friction. The matrix $M(q)$ is the sum of the nominal M_k and unknown M_u matrices. Analogously, the terms $C(q, \dot{q})$ is composed of C_k, C_u, and $G(q)$ is composed of G_k and G_u as shown below.

$$\begin{aligned} M(q) &= M_k + M_u \\ C(q, \dot{q}) &= C_k + C_u \\ G(q) &= G_k + G_u \end{aligned} \tag{4.2}$$

By substituting (4.2) into (4.1), grouping by nominal and unknown terms and isolating $M_k \ddot{q}$. The acceleration of all the joints is written as:

$$\ddot{q} = M_k^{-1} \left(\Gamma(t) - C_k \dot{q} - G_k \right) + \omega_0(t) \tag{4.3}$$

where $\omega_0(t) \in \mathbb{R}^n$ represents the model uncertainty defined as:

$$\omega_0(t) = M_k^{-1} \left(\Gamma_{dis} - M_u \ddot{q} - C_u \dot{q} - G_u - F(\dot{q}) \right) \tag{4.4}$$

The nominal torque from (4.1) gives:

$$\Gamma(t) = M_0 \nu(t) + C_k \dot{q} + G_k \tag{4.5}$$

and $\nu(t) \in \mathbb{R}^n$ is the virtual control to achieve the desired reference. By substituting (4.5) into (4.3), the following uncertain linear system is obtained.

$$\ddot{q} = \nu(t) + \omega_0(t) \tag{4.6}$$

Rewriting (4.6) in a matrix form yields:

$$\dot{z}(t) = Az(t) + B\nu(t) + B\omega_0(t) \tag{4.7}$$

where $z(t) \in \mathbb{R}^{2n}$ is the measured state $z(t) = \left[q^T, \dot{q}^T\right]^T$. The linear system's constant matrices $A \in \mathbb{R}^{2n\times 2n}$ and $B \in \mathbb{R}^{2n\times n}$ are given as:

$$A = \begin{bmatrix} 0_{n\times n} & I_{n\times n} \\ 0_{n\times n} & 0_{n\times n} \end{bmatrix}, \quad B = \begin{bmatrix} 0_{n\times n} \\ I_{n\times n} \end{bmatrix} \tag{4.8}$$

The control $\nu(t)$ for (4.7) is as follows:

$$\nu(t) = \nu_{sm}(t) + \nu_{mpc}(t) \tag{4.9}$$

where $\nu_{sm}(t)$ is computed from the ISM to reject part of $\omega_0(t)$ while $\nu_{mpc}(t)$ is computed by optimizing a learning-based control problem. Both are defined at a later stage.

4.3 Learning-Based Model Predictive Control with Integral Sliding Mode

In this section, the proposed controller is developed and its schematic diagram is shown in Fig. 4.4. The subsequent assumptions are taken into account within this paper.

Assumption 1: The polytopes sets $\mathcal{U} \in \mathbb{R}^n$ and $\mathcal{Z} \in \mathbb{R}^{2n}$ which define the input and state constraint respectively, are compact and bounded. The set $\mathcal{U} = \{\nu(t)|{M_k}^{-1}\Gamma_{min} \leq \nu(t) \leq {M_k}^{-1}\Gamma_{max}\}$ and $\mathcal{Z} = \{z(t)|z_{min} \leq z(t) \leq z_{max}\}$. The values of $\dot{q}$ and $\ddot{q}$ are bounded and continuous.

Assumption 2: The disturbance $\omega_0(t) \in \mathcal{W}$ is locally continuous and upper-bounded by the constant ω_{max}.

$$0 < \|\omega_0(t)\| \leq \omega_{max} < \infty \tag{4.10}$$

4.3.1 Integral Sliding Mode Exponential Reaching Law

The following integral sliding surface (Fridman *et al.*, 2014) was used for the design of the ISM controller.

$$\varsigma(t) = T_{\Xi}\left[e(t) - e(0) - \int_0^t \left(Az(t) + B\nu_{mpc}(t) - \dot{z}_r(t)\right) dt\right] \tag{4.11}$$

where the constant matrix $T_{\Xi} \in \mathbb{R}^{n\times 2n}$ is chosen to obtain an invertible matrix $T_{\Xi}B$. The state error is defined as $e(t) = z(t) - z_r(t) \in \mathbb{R}^{2n}$ with $z_r(t) \in \mathbb{R}^{2n}$ as the desired state vector. The derivative with respect to time of the sliding surface results in:

$$\begin{aligned}
\dot{\varsigma}(t) =& T_{\Xi}(\dot{e}(t) - Az(t) - B\nu_{mpc}(t) + \dot{z}_r(t)) \\
=& T_{\Xi}(\dot{z}(t) - \dot{z}_r(t) - Az(t) - B\nu_{mpc}(t) + \dot{z}_r(t)) \\
=& T_{\Xi}(Az(t) + B(\nu_{sm}(t) + \nu_{mpc}(t)) - \dot{z}_r(t) \\
& - Az(t) - B\nu_{mpc}(t) + \dot{z}_r(t)) \\
=& T_{\Xi}B\nu_{sm}(t)
\end{aligned} \tag{4.12}$$

The ERL proposed in (Fallaha *et al.*, 2011) is as follows:

$$\dot{\varsigma}(t) = -K(\varsigma(t))sig(\varsigma(t)) \tag{4.13}$$

with each element of the vector $sig(\varsigma(t))$ defined as:

$$sign(\varsigma_i) = \begin{cases} 1, & \varsigma_i > 0 \\ 0, & \varsigma_i = 0 \\ -1, & \varsigma_i < 0 \end{cases} \tag{4.14}$$

and the $K(\varsigma(t))$ being for $i = 1, \cdots, n$:

$$K(\varsigma_i) = diag\left(\frac{k_i}{\chi_i + (1-\chi_i)e^{-\gamma_i|\varsigma_i|^{\rho_i}}}\right) \tag{4.15}$$

where $k_i \in \mathbb{R}^+$, $\chi_i \in [0,1]$, $\gamma_i \in \mathbb{R}^+$ and $\rho_i \in \mathbb{R}^+$ are constants. Substituting (4.12) into (4.13), yields the following ISM law:

$$\nu_{sm}(t) = (T_\Xi B)^{-1}[-K(\varsigma(t))sig(\varsigma(t))] \tag{4.16}$$

To ensure the closed-loop stability of the designed method, let us consider the following Lyapunov positive definite function:

$$V = \frac{1}{2}\varsigma(t)^T\varsigma(t) \tag{4.17}$$

And the uncertain linear system (4.7) under the virtual controller (4.9), the derivative of the Lyapunov function yields:

$$\begin{aligned}\dot{V} =& \varsigma(t)^T\dot{\varsigma}(t)\\ =& \varsigma(t)^T(T_\Xi(Az(t) + B(\nu_{sm}(t) + \nu_{mpc}(t) + \omega_0(t))\\ & - \dot{z}_r(t) - Az(t) - B\nu_{mpc}(t) + \dot{z}_r(t))\\ =& \varsigma(t)^T(T_\Xi B\nu_{sm} + T_\Xi B\omega_0)\end{aligned} \tag{4.18}$$

Substituting (4.16) into (4.18) yields:

$$\varsigma(t)^T(-K(\varsigma(t))sig(\varsigma(t)) + T_\Xi B\omega_0(t)) < 0 \tag{4.19}$$

By considering **Assumption 2**, the asymptotic stability is ensured and the following inequality is satisfied:

$$\|K(\varsigma(t))\| > \|T_\Xi B\|\,\omega_{max}\ \forall\, t \tag{4.20}$$

From (4.20), it can be verified that the stability of the uncertain linear system (4.7) is ensured. However, the ISM when rejecting the unknown disturbances by itself, will produce a high chattering on the motors. In order to reduce the required control activity, smaller gains are selected. Thus, the ISM attenuates the unknown disturbances while the model uncertainty due to the remaining disturbances is addressed by the Learning-based MPC. Note that a smaller

model uncertainty allows the Learning-based MPC approach to improve conservatism (Mayne *et al.*, 2011). For that, let us consider the uncertain system that results from applying the virtual control (4.9) to the system (4.7).

$$\dot{z}(t) = Az(t) + B\nu_{mpc}(t) + B\omega_r(t) \tag{4.21}$$

where $\omega_r(t) = \nu_{sm}(t) + \omega_0(t)$ is the remaining unknown disturbance, leading to the following assumption:

Assumption 3: The disturbance $\omega_r(t)$, an unknown variable belonging to the set $\mathcal{W}$, exhibits local continuity and is bounded from above by the constant ω_2.

$$0 < \|\omega_r(t)\| \leq \omega_2 < \omega_{max} < \infty \tag{4.22}$$

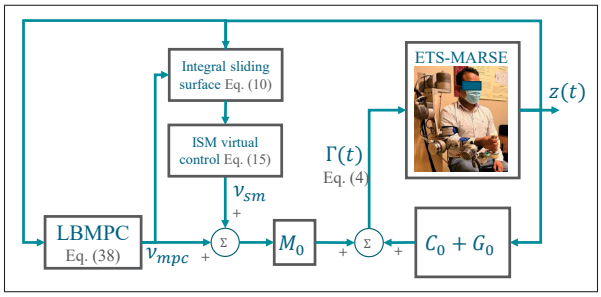

Figure 4.4 Schematic of the Learning-based MPC with ISM

4.3.2 Gaussian Process Disturbance Model

To improve the optimality of the uncertain system (4.21) under the ISM and MPC controllers, a learned prediction model is built by using a disturbance model and a nominal prediction model (4.23).

$$\dot{\bar{z}}(t) = A\bar{z}(t) + B\nu_{mpc}(t) \tag{4.23}$$

The learned prediction model (4.24) represents an accurate prediction model of the exoskeleton-user system. It ensures an optimal performance for rehabilitation tasks by learning the uncertainty introduced by new wearers and improving from trial to trial (Ostafew *et al.*, 2016a). The terms $\dot{\tilde{z}}(t) \in \mathbb{R}^{2n}$ and $\dot{\tilde{z}}(t) \in \mathbb{R}^{2n}$ are the derivatives of the predicted state using the learned model and nominal model respectively.

$$\dot{\tilde{z}}(t) = A\tilde{z}(t) + B\nu_{mpc}(t) + \omega_{\mu}(t) \tag{4.24}$$

The disturbance model $\omega_{\mu}(t) \in \mathbb{R}^{2n}$ (to be defined later) is an approximation of the model uncertainty caused by the remaining unknown disturbance $\omega_r(t)$ in the system (4.21). The model $\omega_{\mu}(t)$ was built using the Gaussian Process learning technique, which is trained using collected data from at least a previous trial. Let us define the measured model uncertainty $\xi(t) \in \mathbb{R}^{2n}$ as:

$$\xi(t) = \dot{z}(t) - \dot{\tilde{z}}(t) \tag{4.25}$$

And the measured regressor $\zeta(t) \in \mathbb{R}^{3n}$ as:

$$\zeta(t) = \left[z(t)^T, \nu_{mpc}(t)^T\right]^T \tag{4.26}$$

During a rehabilitation task, the model uncertainty is measured and stored, and from the collected data, m random measurements are chosen as inducing points to train the disturbance model. Thus the m inducing points set is defined as $\mathcal{D} = [\mathbf{w}^T, \mathbf{a}^T]^T \in \mathbb{R}^{5n \times m}$, where $\mathbf{w} = [\xi_1, ..., \xi_m]$ is composed of the collected model uncertainty ξ_i, and $\mathbf{a} = [\zeta_1, ..., \zeta_m]$ is composed of the collected regressor ζ_i, where $i = 1 \rightarrow m$. For simplicity, we consider each dimension in $\mathbf{w}$ separately for regression, let us define the kth dimension (row) of the collected model uncertainty $\mathbf{w}$ as $\mathbf{w}_k \in \mathbb{R}^{1 \times m}$ and the kth dimension of the modeled disturbance $w(t)$ as $w_k(t) \in \mathbb{R}$ to be defined later.

For a given regressor input $\zeta(t) \in \mathbb{R}^3$ ($n = 1$, for simplicity), we define jointly the collected

model uncertainty $\mathbf{w}_k$ and the modeled disturbance $w_k(t)$ as a normally Gaussian distribution.

$$\begin{bmatrix} \mathbf{w}_k^T \\ w_k(t) \end{bmatrix} \sim \mathcal{N}\left(0_{m+1\times 1}, \begin{bmatrix} \Upsilon & k_d(\zeta(t))^T \\ k_d(\zeta(t)) & k_r(\zeta(t),\zeta(t)) \end{bmatrix}\right) \tag{4.27}$$

where the (i, j) elements of the covariance matrix $\Upsilon \in \mathbb{R}^{m\times m}$ are equal to $k_r(\zeta_i, \zeta_j)$, $i = 1 \rightarrow m$, $j = 1 \rightarrow m$. Given a regressor input $\zeta(t)$, the term $k_d(\zeta(t)) \in \mathbb{R}^{1\times m}$ is defined as:

$$k_d(\zeta(t)) = [k_r(\zeta(t), \zeta_1), ..., k_r(\zeta(t), \zeta_m)] \tag{4.28}$$

In this paper, the square exponential (Williams & Rasmussen, 2006) was used as the covariance function.

$$k_r(\zeta_i, \zeta_j) = \sigma_f^2 \exp\left(-\frac{1}{2}(\zeta_i - \zeta_j)^T l^{-2}(\zeta_i - \zeta_j)\right) + \sigma_n^2 \delta_{ij} \tag{4.29}$$

We assume that the observed disturbance has variance $\sigma_n \in \mathbb{R}^+$ with zero mean. The Kronecker delta $\delta_{ij} \in \{0, 1\}$ is defined as:

$$\delta_{ij} = \begin{cases} 1, & i == j \\ 0, & Otherwise \end{cases} \tag{4.30}$$

The hyper parameters $\sigma_f \in \mathbb{R}^+$, $l = diag(l_{m1}, l_{m2}, l_{m3}) \in \mathbb{R}^{3\times 3}$ and σ_n are computed by solving the log marginal likelihood (Williams & Rasmussen, 2006). The modeled disturbance $w_k(t)$ has a mean and variance given an arbitrary input $\zeta(t)$ which are expressed as follows (Williams & Rasmussen, 2006):

$$w_k(t) \sim \mathcal{N}(\omega_{\mu,k}(t), \omega_{\Sigma,k}(t)) \tag{4.31}$$

Where $\omega_{\mu,k}(t)$ and $\omega_{\Sigma,k}(t)$ are the modeled disturbance's mean and variance respectively and are computed as:

$$\omega_{\mu,k}(t) = k_d(\zeta(t))\Upsilon^{-1}\mathbf{w}_k^T \tag{4.32}$$

$$\omega_{\Sigma,k}(t) = k_r(\zeta(t), \zeta(t)) - k_d(\zeta(t))\Upsilon^{-1}k_d(\zeta(t))^T \tag{4.33}$$

Using the mean of each kth dimension $\omega_{\mu,k}(t)$, we build the disturbance model $\omega_\mu(t) = [\omega_{\mu,1}(t), ..., \omega_{\mu,2n}(t)]^T$. Equations (4.32) and (4.33) are developed using the Gaussian linear transformation through (4.27). The variance value (4.33) is not used in this paper but for a future paper about Cautious MPC. For the next subsection, let us treat the stochastic disturbance model as a simple propagation of mean obtaining a deterministic model as reported in (Hewing *et al.*, 2020b).

4.3.3 Learning-Based Optimal Control Problem

A Learning-based approach is designed to guarantee safe rehabilitation tasks by considering input and state constraints. The problem of optimal control seeks trade-offs between fulfilling constraints and achieving the control objective (Mesbah, 2016).

$$\begin{aligned}
\min_{\mathbf{\Pi}(t)} \quad & J(\tilde{\mathbf{z}}(t), \mathbf{\Pi}(t)) \\
\text{s.t.:} \quad & \dot{\tilde{z}}(\tau) = A\tilde{z}(\tau) + Bv_{mpc}(\tau) + \omega_\mu(\tau), \quad I.C.\ \tilde{z}(t) := z(t) \\
& \dot{\bar{z}}(\tau) = A\bar{z}(\tau) + Bv_{mpc}(\tau), \quad I.C.\ \bar{z}(t) := z(t) \\
& O_\Pi(\tau) \in \mathcal{U} \ominus K\Omega_{tube}, \quad \bar{z}(t) \in \mathcal{Z} \ominus \Omega_{tube} \\
& \bar{e}(t_{end}) \in \Omega_\epsilon
\end{aligned} \tag{4.34}$$

where the future time τ that starts from time t to the end of the prediction horizon t_{end}. The states of the nominal and learned prediction models are initialized with the measured state $z(t)$ at each sampling time. The set Ω_{tube} is a convex hull (Aswani *et al.*, 2013) built by using the residual of the learning process. The terminal region Ω_ϵ is used as terminal constraint to ensure the feasibility of the learning-based optimization. The term $\mathbf{\Pi}(t) = [O_\Pi(t), ..., O_\Pi(t_{end})]$ is a control input sequence and $\tilde{\mathbf{z}}(t) = [\tilde{z}(t), ..., \tilde{z}(t_{end})]$ is the resultant predicted states sequence obtained from the learned model. The future cost is:

$$J(\tilde{\mathbf{z}}(t), \mathbf{\Pi}(t)) = L(\tilde{z}(t_{end})) + \int_t^{t_{end}} \Lambda(\tilde{z}(\tau), O_\Pi(\tau))d\tau \tag{4.35}$$

where $\Lambda \in \mathbb{R}$ is the stage cost at future time τ is:

$$\Lambda(\tilde{z}(\tau), O_{\Pi}(\tau)) = \frac{1}{2}\tilde{e}(\tau)^T Q\tilde{e}(\tau) + \frac{1}{2}O_{\Pi}(\tau)^T R O_{\Pi}(\tau) \tag{4.36}$$

and the terminal cost is:

$$L(\tilde{z}(t_{end})) = \frac{1}{2}\tilde{e}(t_{end})^T P\tilde{e}(t_{end}) \tag{4.37}$$

and the nominal $\bar{e}$ and learned $\tilde{e}$ predicted errors at time τ are:

$$\bar{e}(\tau) = \bar{z}(\tau) - z_r(\tau) \tag{4.38}$$

$$\tilde{e}(\tau) = \tilde{z}(\tau) - z_r(\tau) \tag{4.39}$$

The matrix $P \in \mathbb{R}^{2n\times 2n}$ is both symmetric and positive definite, while $Q \in \mathbb{R}^{2n\times 2n}$ and $R \in \mathbb{R}^{n\times n}$ are positive diagonal matrices whose specific definitions will be provided later. When addressing the optimal control problem (4.34), it's worth noting that the dynamic programming method, as described in (Bellman, 1966), grapples with the challenge of the curse of dimensionality.

Considering the Gaussian Process disturbance model (4.24), the optimization problem (4.34) is computationally expensive for the feedback gains (Hewing *et al.*, 2020a). Then, the following feedback control law is chosen:

$$\nu_{mpc}(\tau) = K_{mpc}\bar{e}(\tau) + O_{\Pi}(\tau) \tag{4.40}$$

where K_{mpc} is a constant optimal gain (to be designed offline) and $O_{\Pi}(\tau)$ is the optimization term. For the next subsection, we consider the following assumptions:

Assumption 4: In (4.35) Λ and L are $\mathcal{K}$ type functions, continuous, and fulfill $L(0) = 0$ and $\Lambda(0, 0) = 0$.

Assumption 5: Using the continuous differential function (4.29), the learned prediction model (4.24) is of class c^2.

4.3.4 Terminal Ingredients

The terminal ingredients are designed by using the nominal prediction model (Chen & Allgöwer, 1998) to ensure recursive feasibility and closed-loop stability of the online optimization. With this goal in mind, contemplate the subsequent terminal feedback controller, which aids in the offline determination of the terminal region and terminal penalty.

$$\bar{u}(t) = K_{mpc}\bar{e}(t) \tag{4.41}$$

where the gain $K_{mpc} \in \mathbb{R}^{n\times 2n}$ is defined as:

$$K_{mpc} = [diag(k_{11}, ..., k_{n1}), diag(k_{12}, ..., k_{n2})] \tag{4.42}$$

where $k_{i1} < 0$, $k_{i2} < 0$ and $k_{i2}^2 + 4k_{i1} > 0$ for $i = 1, ..., n$ (Dai *et al.*, 2020). It can be shown that the nominal model (Eq. 4.23) is controllable. Consequently, a feedback gain K_{mpc} is computed such that $A + BK_{mpc}$ ensures stability. The positive constant η that satisfies $\eta < -\lambda_{max}(A + BK_{mpc})$ leads to a unique terminal penalty P which is computed by Eq. (4.43).

$$A_\eta^T P + PA_\eta + Q + K_{mpc}^T R K_{mpc} \leq 0 \tag{4.43}$$

where:

$$A_\eta = A + BK_{mpc} + \eta I_{2n\times 2n} \tag{4.44}$$

The theory of MPC assumes that at future time t_{end}, a proper P matrix allows optimization along an infinite horizon (Mayne *et al.*, 2000). For the terminal region, the largest possible constant ϵ is found such that $\bar{u} \in K_{mpc}\Omega_\epsilon, \forall \bar{e} \in \Omega_\epsilon$, where $\Omega_\epsilon = \{\bar{e} \in \mathbb{R}^{2n} | \bar{e}^T P \bar{e} \leq \epsilon\}$. And ϵ is computed by optimizing equation (19) in (Chen & Allgöwer, 1998) to ensure recursive feasibility. Taking the matrix P from Eq. (4.43) as a valid matrix solution, and applying Lemma 1 from (Chen & Allgöwer, 1998), we guarantee that equation (4.35) fulfills:

$$\dot{L} + \Lambda \leq 0 \tag{4.45}$$

In other words, when the error $\bar{e}$ enters the region Ω_ϵ, it remains within that region while adhering to input constraints. Consequently, the nominal control $\bar{u}(t)$ remains invariant within the terminal region Ω_ϵ.

4.3.5 Recursive Feasibility

For this subsection, let us consider the following conventional assumptions:

Assumptions 6: At time instant t, there exists a singular optimal control sequence for the optimal control problem (4.34).

$$\mathbf{\Pi}^*(t) = [O^*_\Pi(t), .., O^*_\Pi(t_{end})] \tag{4.46}$$

And optimal resultant state sequence.

$$\begin{aligned} \bar{\mathbf{z}}^*(t) &= [\bar{z}^*(t), ..., \bar{z}^*(t_{end})] \\ \tilde{\mathbf{z}}^*(t) &= [\tilde{z}^*(t), ..., \tilde{z}^*(t_{end})] \end{aligned} \tag{4.47}$$

Assumptions 7: At time instant $t + t_s$, as per the construction, there exists a solution for the optimal control problem in Eq. (4.34), which may not be optimal but is, at the minimum, a feasible optimal control sequence.

$$\mathbf{\Pi}(t + t_s) = [O^*_\Pi(t + t_s), .., O^*_\Pi(t_{end}), \bar{u}(t_{end} + t_s)] \tag{4.48}$$

And feasible state sequences

$$\begin{aligned} \bar{\mathbf{z}}(t + t_s) &= [\bar{z}^*(t + t_s), ..., \bar{z}^*(t_{end}), \bar{z}(t_{end} + t_s)] \\ \tilde{\mathbf{z}}(t + t_s) &= [\tilde{z}^*(t + t_s), ..., \tilde{z}^*(t_{end}), \tilde{z}(t_{end} + t_s)] \end{aligned} \tag{4.49}$$

where t_s is the operation time between measurements. Since the optimization is feasible (**Assumption 6**) $\bar{z}^*(t_{end}) \in z_r(t_{end}) \oplus \Omega_\epsilon$. By utilizing Eq. (4.45), we have established the invariance of the feedback terminal controller within the terminal region, as discussed in (Mayne *et al.*, 2011), denoted as $\bar{u}(t_{end} + t_s)$, this indicates that $\bar{z}(t_{end} + t_s) \in z_r(t_{end}) \oplus \Omega_\epsilon$. Therefore,

recursive feasibility is ensured. While the input and state constraints are fulfilled, $\bar{\mathbf{z}}(t) \in \mathcal{Z} \ominus \Omega_\epsilon$ and $\mathbf{\Pi}(t) \in \mathcal{U} \ominus K_{mpc}\Omega_\epsilon$.

4.3.6 Closed-Loop Stability

The cost function (4.35) is used as Lyapunov function and its value at time t under the optimal control sequence $\mathbf{\Pi}^*(t)$ is:

$$J(\tilde{\mathbf{z}}^*(t), \mathbf{\Pi}^*(t)) = L(\tilde{z}^*(t_{end})) + \int_t^{t_{end}} \Lambda(\tilde{z}^*(\tau), O_\Pi^*(\tau))d\tau \tag{4.50}$$

The cost at time $t + t_s$ under $\mathbf{\Pi}(t + t_s)$ is:

$$\begin{aligned} J(\tilde{\mathbf{z}}(t + t_s), \mathbf{\Pi}(t + t_s)) = &\int_{t+t_s}^{t_{end}} \Lambda(\tilde{z}^*(\tau), O_\Pi^*(\tau))d\tau \\ &+ \int_{t_{end}}^{t_{end}+t_s} \Lambda(\tilde{z}(\tau), \bar{u}(\tau))d\tau + L(\tilde{z}(t_{end} + t_s)) \end{aligned} \tag{4.51}$$

Subtracting (4.51) and (4.50) yields:

$$\Delta J(t + t_s) = J(\tilde{\mathbf{z}}(t + t_s), \mathbf{\Pi}(t + t_s)) - J(\tilde{\mathbf{z}}^*(t), \mathbf{\Pi}^*(t)) \tag{4.52}$$

And expanding (4.52) one obtains:

$$\begin{aligned} \Delta J(t + t_s) = &L(\tilde{z}(t_{end} + t_s)) - L(\tilde{z}^*(t_{end})) \\ + \int_{t_{end}}^{t_{end}+t_s} &\Lambda(\tilde{z}(\tau), \bar{u}(\tau))d\tau - \int_t^{t+t_s} \Lambda(\tilde{z}^*(\tau), O_\Pi^*(\tau))d\tau \end{aligned} \tag{4.53}$$

By integrating (4.45) from $\tau = [t_{end}, t_{end} + t_s]$ considering $\tilde{\mathbf{z}}(t + t_s)$ yields:

$$\int_{t_{end}}^{t_{end}+t_s} \Lambda(\tilde{z}(\tau), \bar{u}(\tau))d\tau + L(\tilde{z}(t_{end} + t_s)) - L(\tilde{z}^*(t_{end})) \leq 0 \tag{4.54}$$

Thus:

$$\Delta J(t + t_s) \leq -\int_t^{t+t_s} \Lambda(\tilde{z}^*(\tau), O_\Pi^*(\tau))d\tau \leq 0 \tag{4.55}$$

The optimal cost fulfills the inequality (Jadbabaie *et al.*, 2001)

$$J(\tilde{z}^*(t+t_s), O_\Pi^*(t+t_s)) \leq J(\tilde{z}^*(t), O_\Pi^*(t)) \tag{4.56}$$

Thus from (4.55), we ensure that $\Delta J(t+t_s) \leq 0$ and the asymptotic stability of the system. Considering **Assumption 3**, there is a $\mathcal{K}_\infty$ function $\beta(\cdot, t)$ and a $\mathcal{K}$ function $\psi(\cdot)$ (Limon *et al.*, 2009) that fulfills:

$$\|\tilde{e}^*(t)\| \leq \beta(\|\tilde{e}^*(0)\|, t) + \psi(\omega_2) \tag{4.57}$$

For a given t_s (typically 1 ms for electromechanical systems) the measured stated z fulfills $z(t+t_s) \in \tilde{z}^*(t+t_s) \oplus \Omega_{tube}$ implying recursive feasibility and closed-loop stability. Now, we are able to describe Algorithm 4.1.

Algorithm 4.1 Learning-based MPC with ISM controller.

Offline:

Choose the weighting matrices Q, R and a proper K_{mpc}. Choose η satisfying $\eta < -\lambda_{max}(A + BK_{mpc})$.

Compute the terminal penalty P by (4.43) and terminal region Ω_ϵ.

Train the disturbance model (4.31) using the data $\mathcal{D}$ obtained from a previous trial.

Build the set Ω_{tube} from the residual of the learning process.

Set the total trajectory time T.

Online:

while $(t < T)$ **do**

  Measure the system's states $q(t)$ and $\dot{q}(t)$.

  Solve the optimal control problem (4.34) and obtain ν_{mpc} by (4.40).

  Compute the ISM control signal ν_{sm} by (4.16).

  Compute the virtual control law (4.9).

  Calculate the torque Γ by (4.5) and apply to the real system (4.3).

  Let $t = t + t_s$

end while

4.4 Experimental configuration and results

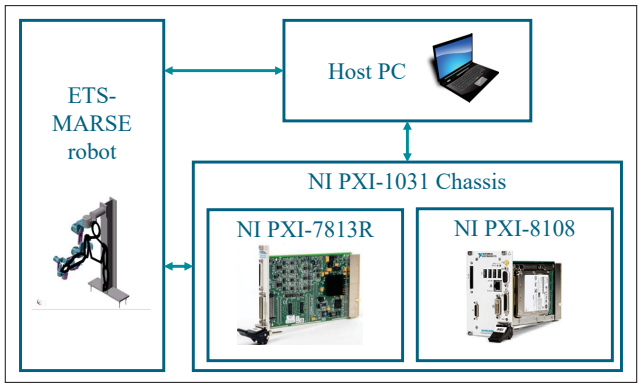

Figure 4.5 Real-time system

4.4.1 Real-Time System

Our real-time system comprises three processing units, as depicted in Fig. 4.5. An industrial PC featuring an Intel dual-core processor running at 2.53 GHz and equipped with 8 GB of RAM, is responsible for performing exoskeleton linearization and executing the predictive controller. The FPGA within the PXI-7813R is dedicated to managing actuators and sensors, including motor current sensors and position hall effect sensors. Lastly, a Host PC serves as the user interface, facilitating the storage and presentation of results from completed trials.

4.4.2 Controller Settings

The objective is to follow a sixteen-second infinite shape trajectory (https://www.youtube.com/watch?v=IXlde9tTQSw) as shown in Fig. 4.7. This challenging trajectory was chosen because multiple joints are active simultaneously during the exercise as shown in Fig. 4.6. The tests were carried out with a healthy subject (32 years old, 160 cm height, and 76 kg weight), who using his whole arm intentionally applied a force in the positive direction of the z_0 axis (see Fig. 4.3). The system was linearized with the physical parameters of the robot without the subject (Fallaha

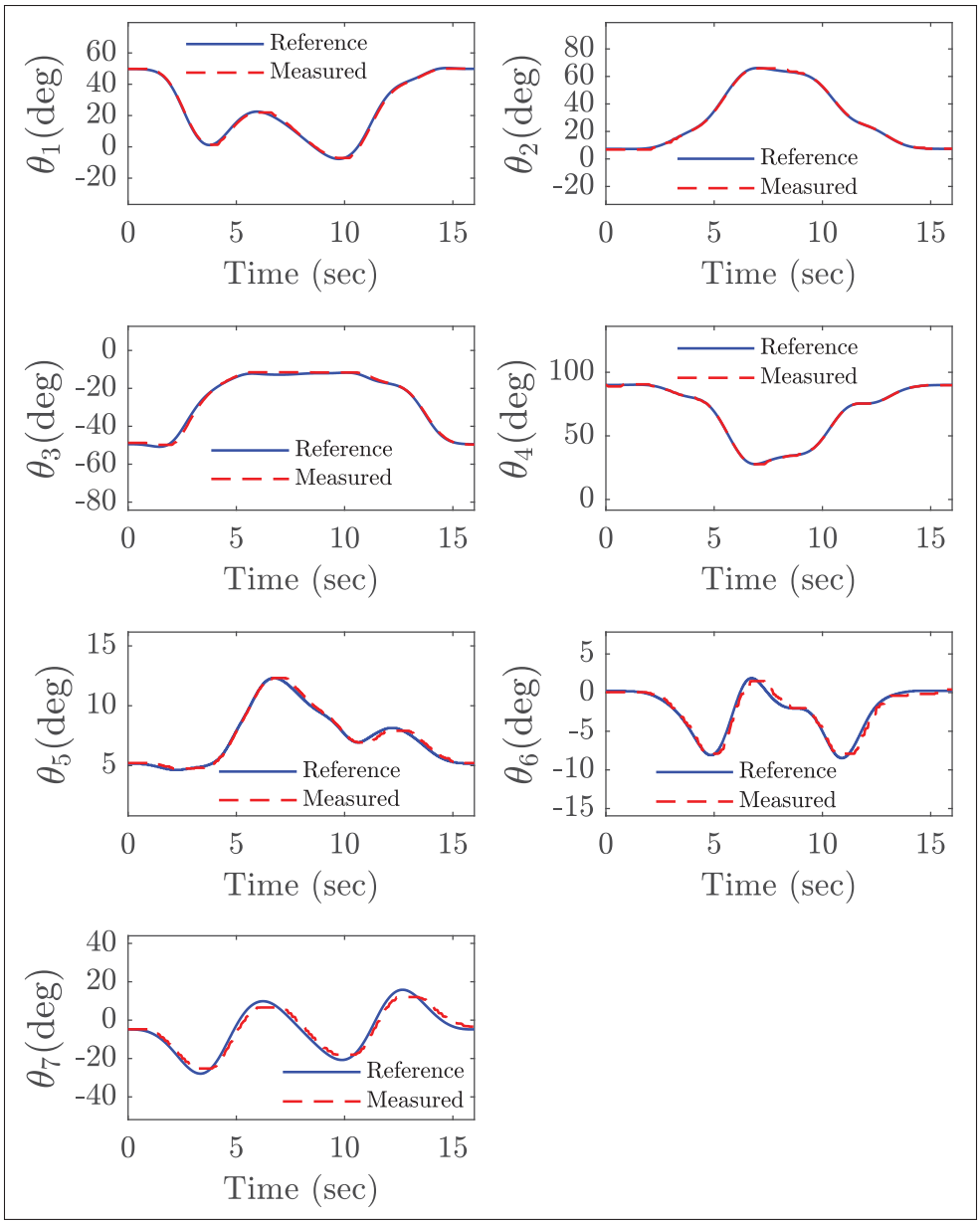

Figure 4.6 Joint trajectory tracking for the proposed controller

et al., 2020). The torque input limits are in Nm.

$$\Gamma_{max} = [25.38\ 50.76\ 30.60\ 17.62\ 15.93\ 6.37\ 6.37]^T$$

$$\Gamma_{min} = -1 \cdot [25.38\ 16.92\ 30.60\ 17.62\ 15.93\ 6.37\ 6.37]^T$$

The position and velocity limits are in deg and deg/sec.

$$q_{max} = [140\ 140\ 75\ 120\ 85\ 20\ 60]$$

$$\dot{q}_{max} = [\ 100\ 100\ 100\ 100\ 100\ 100\ 100]$$

$$z_{max} = [q_{max}\ \dot{q}_{max}]^T$$

$$q_{min} = -1 \cdot [0\ 0\ 85\ 0\ 85\ 30\ 50]$$

$$z_{min} = [q_{min}\ -\dot{q}_{max}]^T$$

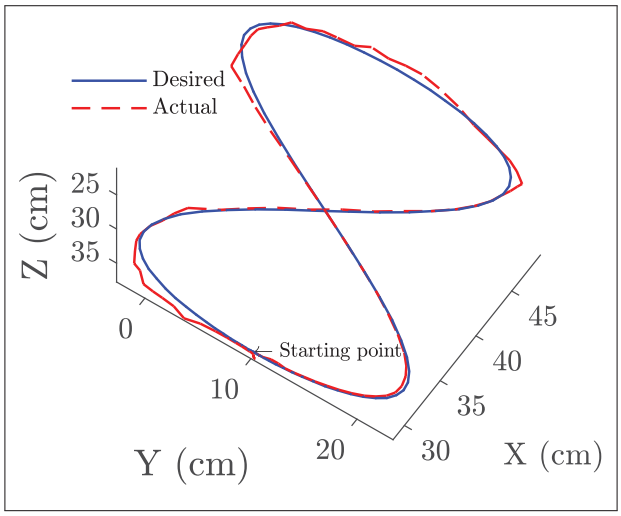

Figure 4.7 Cartesian trajectory tracking for the proposed controller

The evaluation of the proposed controller was carried out through a comparison with several controllers using the same design parameters, except for the gain k_i as shown in Table 4.3. From a first trial using the SMC-ERL, we have collected the data $\mathcal{D}$ for two reasons. First, to train the GP using $m = 30$ data points from the collected data. And second to build the set Ω_{tube} from the residual of the learning process as shown in (Aswani *et al.*, 2013). The optimization feasibility is ensured by the terminal region size set as $\epsilon = 1.2$, a prediction horizon of 10 steps with a sampling time of $t_s = 1ms$, and we use as optimizer the gradient descent method. The matrices

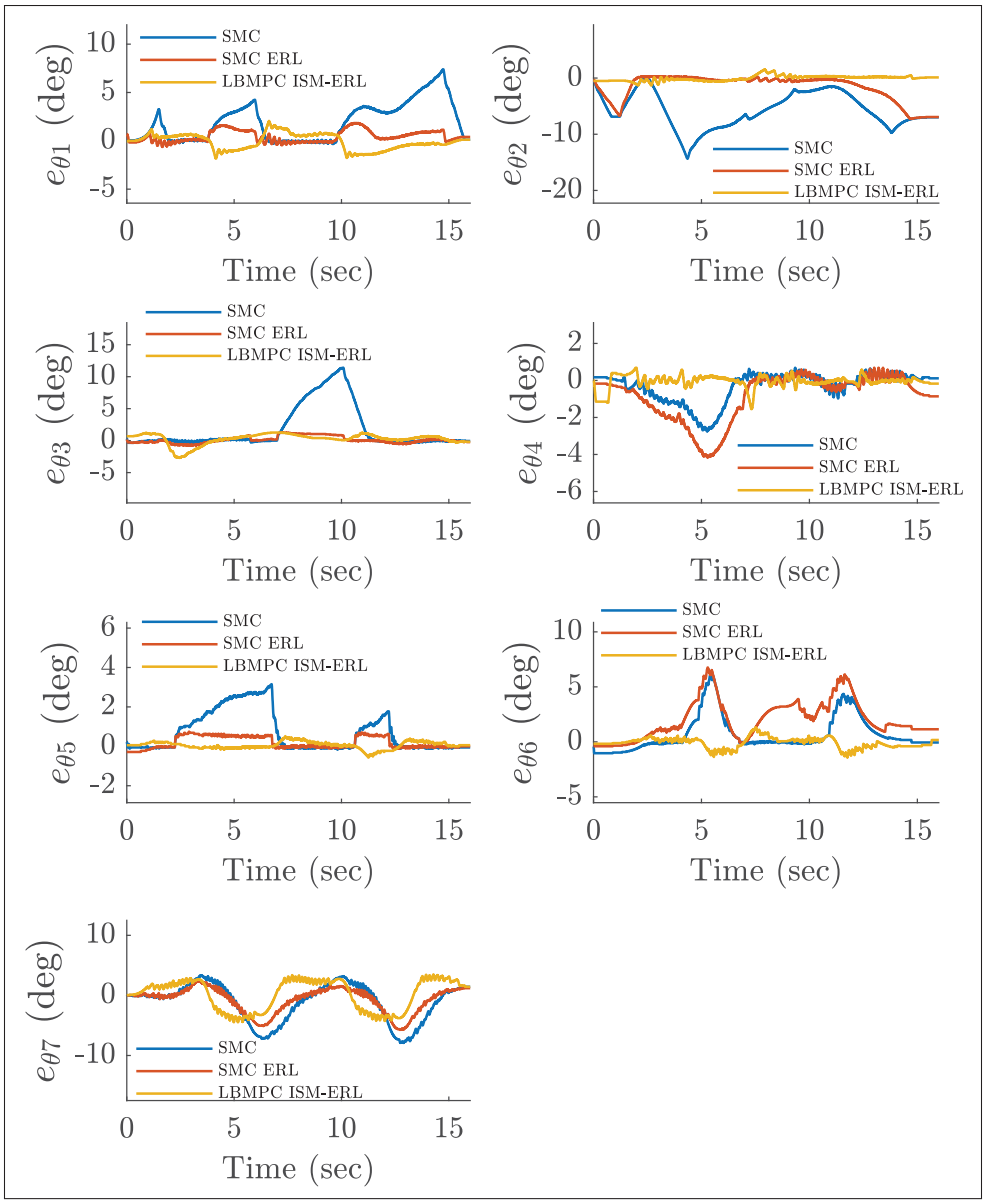

Figure 4.8 Joint tracking error comparison

Q and R for the cost function were chosen heuristically as:

$$Q = diag(660, 3100, 100, 2500, 650, 50, 10, 1, 1, 1, 1, 1, 1, 1)$$

$$R = 0.01 I_{n \times n}$$

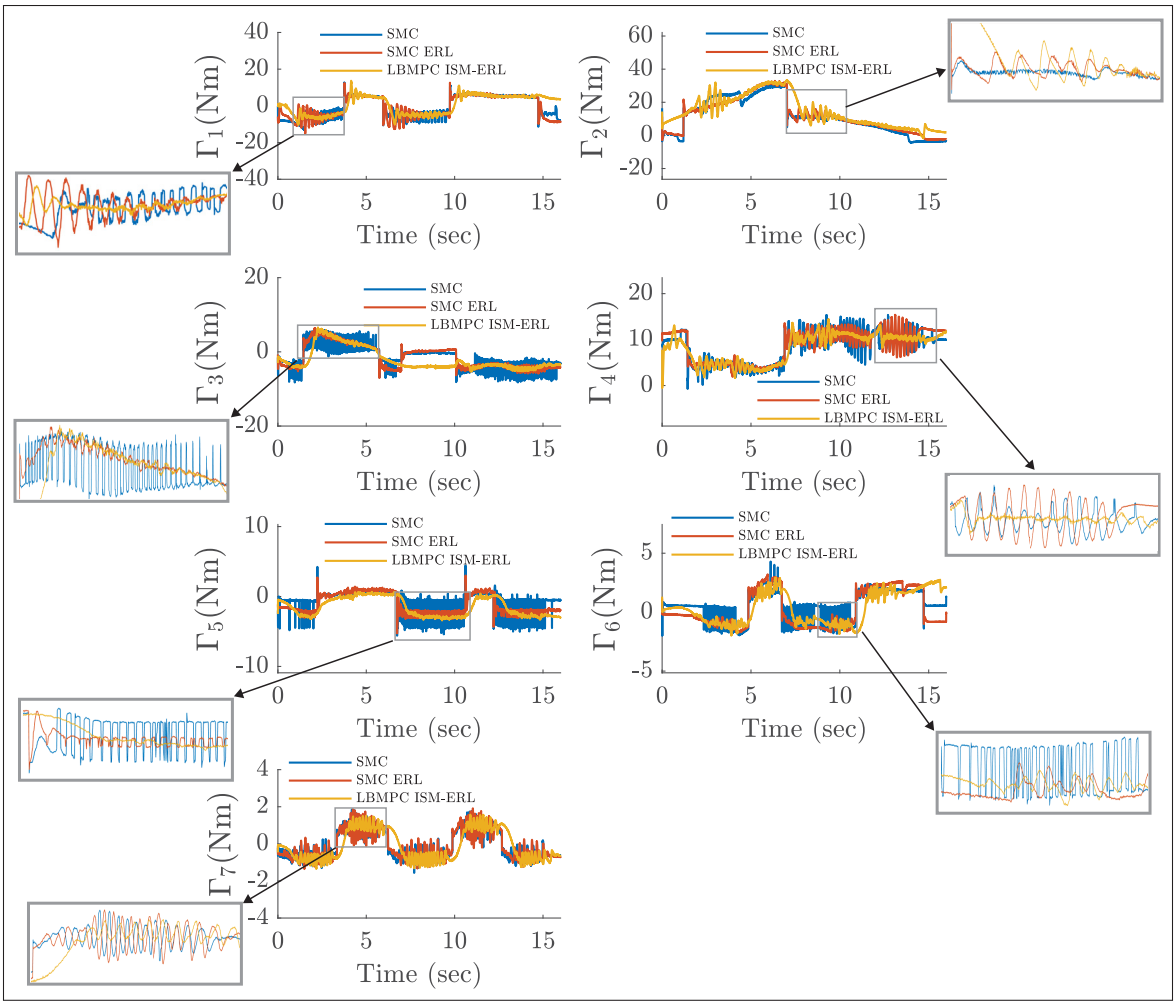

Figure 4.9 Control signal comparison

while the K_{mpc} is chosen to fulfill the condition below Equation (4.42).

$$K_{mpc} = [-K_{mpc1}, -K_{mpc2}]$$

$$K_{mpc1} = diag(253, 547, 99, 492, 252, 70, 31)$$

$$K_{mpc2} = diag(30, 33, 14, 32, 23, 12, 8)$$

By choosing the constant $\eta = 18.61$ that fulfils the condition $\eta < -\lambda_{max}(A + BK_{mpc})$, we solve equation (4.43) obtaining the following unique matrix P such as its elements are:

$$P_{11} = diag(662, 879, 600, 831, 661, 594, 589)$$
$$P_{12} = P_{21} = diag(15.9, 18.9, 14.9, 18.2, 15.9, 14.8, 14.8)$$
$$P_{22} = diag(0.78, 0.83, 0.77, 0.82, 0.78, 0.77, 0.77)$$

4.4.3 Controller results

The proposed approach shows a smaller tracking error as shown in Fig. 4.8 and the improvement with respect to existing controllers is shown in Table 4.4, where the computational time is below $1ms$ for the proposed controller. The control activity was reduced due to the Learning-based MPC approach that enables the choice of a smaller gain k_i without losing stability (See Fig. 4.9). Additionally in this paper, to show the Learning-based improvement from trial to trial, three trials were performed; Trial 1 (T1): MPC with ISM (without the Learning-based approach), Trial 2 (T2): Using the collected data from trial 1, the GP disturbance model was trained and then the Learning-based MPC with ISM was performed, and Trial 3 (T3): Using the collected data from trials 1 and 2, the GP was trained then the Learning-based MPC with ISM was performed, the improvement from trial to trial is shown in Fig. 4.10.

Table 4.3 SMC design parameters

	SMC (Fallaha *et al.*, 2011)	SMC-ERL (Fallaha *et al.*, 2011)	Proposed controller
k_i	$diag([6\ 6\ 6\ 2\ 8\ 2\ 1]) * 100$	$diag([145\ 155\ 160\ 45\ 180\ 55\ 30])$	$diag([30\ 30\ 30\ 10\ 40\ 10\ 5])$
λ_i	$diag([32\ 26\ 13\ 38\ 13\ 5\ 4])$	$diag([32\ 26\ 13\ 38\ 13\ 5\ 4])$	-
p_i	-	1	1
γ_i	-	20	20
χ_i	-	0.1	0.1
T_Ξ	-	-	$[0_{7\times7}, diag(1_{1\times n})]$

4.5 Discussion

The feedback linearization converts the coupled system into a decoupled uncertain linear system. The ISM reduces the model uncertainty by attenuating the unknown disturbance $\omega_0(t)$ into the smaller $\omega_r(t)$. Considering the uncertain linear system (4.21), the Learning-based MPC ensures input and state constraints are fulfilled with high accuracy and low control effort. Although high accuracy in trajectory tracking is not strongly necessary in rehabilitation tasks, the high precision is a result of the unknown disturbance attenuation due to the ISM and the optimal performance achieved by the Learning-based MPC. Therefore, it is an important result for highly nonlinear systems subject to unexpected dynamics. Moreover, the choice of a smaller ISM gain for only disturbance attenuation reduces the chattering in the control signal.

Table 4.4 Controllers comparison in terms of error RMS values

Joint (i)	$\|e\|_{RMS}$ **(rad)**		
	Proposed controller	**SMC ERL**	**SMC std**
1	0.0143	0.0130	0.0488
2	0.0082	0.0500	0.1148
3	0.0155	0.0096	0.0658
4	0.0061	0.0261	0.0152
5	0.0033	0.0063	0.0214
6	0.0085	0.0467	0.0300
7	0.0445	0.0385	0.0588
Mean	0.0143	0.0271	0.0507
Normalized	1.0000	1.8940	3.5361
Computation time (ms)	0.687	0.281	0.273

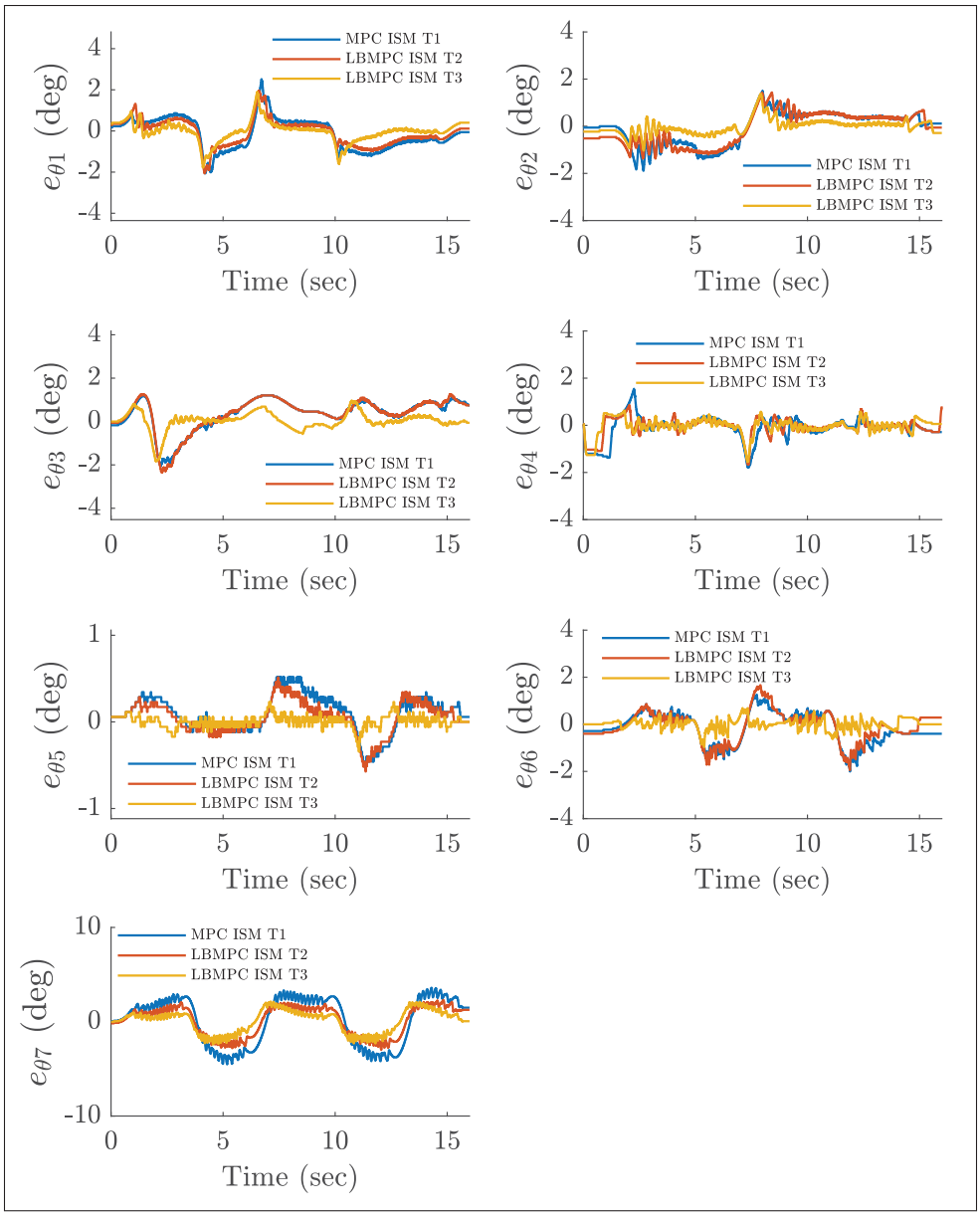

Figure 4.10 Learning-based improvement from trial to trial

4.6 Conclusion

Through this paper, we propose a robust Learning-based MPC with ISM to provide safe robotic rehabilitation in the presence of unknown patient efforts and unmodeled physiological characteristics. The experimental results, with a healthy subject, prove the approach's superior

effectiveness compared to conventional SMC and SMC with ERL. Additionally, the Learning-based improvement from trial to trial was proven. Considering the results obtained, the proposed controller showed better performance (tracking error and control chattering) compared to existing controllers. This article shows how to implement this technique in highly nonlinear robotic exoskeletons. As future work a Cautious MPC approach will be considered to improve safety in rehabilitation tasks.

Acknowledgments

The first author would like to thank to CONAHCYT (Mexican National Council of Humanities, Science and Technology) for the support under grant 739833. CONAHCYT was previously named CONACYT (Mexican National Council of Science and Technology).

CHAPTER 5

MIRROR REHABILITATION SYSTEM FOR UPPER-LIMB MOTOR SKILL IMPROVEMENT IN STROKE SURVIVORS

David Bedolla-Martinez[1] , Yassine Kali[1] , Maarouf Saad[1] , Cristobal Ochoa-Luna[2] , Mohammad Habibur Rahman[3]

[1] Département de Génie Mécanique, École de Technologie Supérieure, 1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

[1] Tecnologico de Monterrey, School of Engineering and Sciences, Ave. Eugenio Garza Sada 2501, Monterrey, N.L. México, PC: 64849

[1] Mechanical and Biomedical Engineering Department, University of Wisconsin-Milwaukee, Milwaukee, WI, USA, PC: 53211

Paper submitted to the *Biocybernetics and Biomedical Engineering*, November 2023.

5.1 Introduction

Stroke is a leading cause of upper-limb disability worldwide (G. Broeks *et al.*, 1999). While advancements in quick-response protocols have reduced stroke-related mortality, the growing number of survivors has increased the demand for post-stroke rehabilitation services (Johnson *et al.*, 2019). Conventional rehabilitation approaches focus on restoring motor skills through repetitive exercises, targeting muscle weakness, and stimulating brain neuroplasticity (Johansson, 2000). Neuroplasticity refers to the brain's remarkable capacity to reorganize itself by forming new neural pathways or transferring cognitive functions from damaged to undamaged areas. This transformation of brain sensory representation is known as mirror mechanism. Studies have demonstrated that conventional mirror rehabilitation, which utilizes visual illusions alone, can induce neuroplasticity in the brain through mirror neurons in the premotor cortex (Park *et al.*, 2015). Furthermore, the use of a 2 DoF exoskeleton robot has demonstrated improvements in patient kinesthesia (Nam *et al.*, 2017) by inducing neuroplasticity due to the direct visual feedback and proprioceptive stimulation of both arms moving symmetrically. However, despite the evident benefits, the integration of robotic exoskeletons and mirror rehabilitation remains limited in the existing literature.

In this sense, few robotic systems are found in the literature for single and multi-joint mirror rehabilitation applications. In the single joint rehabilitation case, a force sensor is used for wrist flexion and extension movement (Kim & Kim, 2017). In (Yang *et al.*, 2021b), a combination of surface EMG and force sensor is used to control the force in the impaired elbow for flexion and extension movements. Similar works controlling a single joint can be found in the literature using superficial EMG for wrist (Zhang *et al.*, 2016), and elbow (Hajian *et al.*, 2021) (Zhang *et al.*, 2019a). For multi-joint mirror rehabilitation, the feasibility of only upper-arm and forearm joint estimation using IMU (Inertial Measurement Units) and EMG sensors were studied for an 8 DoF exoskeleton in (Blana *et al.*, 2016). In (Wang & Fu, 2011), a mirror rehabilitation system for upper-limb was presented using only IMU sensors, later the system underwent an upgrade by integrating a motion prediction module based on surface EMG signals (Ren *et al.*, 2019) for an 11 DoF exoskeleton.

It is worth noting that the previous multi-joint systems lack a wrist orientation estimator from EMG signals. To that end, different EMG classifier techniques have been proposed with good results such as Support Vector Machine (SVM) and Linear Discriminant Analysis (LDA) (Khushaba *et al.*, 2016), optimized artificial neural networks (Lima *et al.*, 2018), K-Nearest Neighborhood (KNN) (Liu & Zhou, 2012) and Fuzzy c-means clustering (Momen *et al.*, 2007). For further information on EMG pattern recognition for wrist orientation, a recent review can be found in (Parajuli *et al.*, 2019). In contrast to previous mirror robotic systems that focused solely on the joint estimation of the upper-arm and forearm, a wrist orientation estimator is incorporated in this paper. To achieve this, a deep convolutional neural network model is employed. This approach was chosen because there are few applications of EMG recognition in real-time using this technique (Yang & Liu, 2021).

The previous works have not addressed the problems of uncertainties, external disturbances, and unmodeled dynamics related to the wearer. To address the above-mentioned shortcomings, in this paper, to validate the mirror rehabilitation system, a novel feedback controller is proposed to overcome external disturbances and a special type of model uncertainties caused by the wearer, as well as to obtain fixed-time convergence rates (Basin, 2019) and prescribed performance

(Bu, 2023). In the literature, for finite-time convergence the terminal sliding mode control (Venkataraman & Gulati, 1993) guarantees that the tracking error converges globally to the origin. For fixed-time convergence, the twisting (Polyakov & Poznyak, 2009a) and super-twisting (Polyakov & Poznyak, 2009b) have been proposed to ensure convergence within a pre-defined time. An example of the fixed-time approach designed for a robotic manipulator can be found in (Zhang *et al.*, 2019b).

Regarding the prescribed performance control approach, it allows us to predefine the convergence rate of the tracking error, as well as the maximum overshoot (Qin & Sun, 2020). An example can be found in (Zhang & Yu, 2023) where an observer-based prescribed performance was designed for an autonomous surface vessel.

In summary, a mirror rehabilitation system incorporating a wrist orientation estimator is proposed. In addition, to track the mirrored trajectory, a novel fixed-time super-twisting-like algorithm with prescribed performance is developed to overcome the external disturbances and model uncertainties caused by the wearer on the robotic exoskeleton. Thus the main contributions of this paper are as follows:

1. Propose a mirror upper-limb rehabilitation scheme to anthropomorphically locate the elbow and forearm, achieving comfortable and natural postures.
2. Incorporate into the proposed system, a wrist orientation estimator using EMG signals from the forearm muscles.
3. Propose a fixed-time super-twisting-like algorithm with prescribed performance to overcome disturbances and uncertainties while tracking a mirrored trajectory.

This paper is organized as follows: Section 5.2 is dedicated to the preliminaries (forward kinematics and motion capture system). Section 5.3 introduces the Mirror rehabilitation system by developing the human forward kinematics, the wrist orientation estimator, a human-type inverse kinematic solution, and the solution to the inverse orientation problem. In Section 5.4, the feedback linearization of the robotic exoskeleton is performed, and the design of a fixed-time super-twisting-like algorithm and its stability are presented. In Section 5.5, the mirror

rehabilitation is implemented, where the wrist orientation estimator is evaluated, as well as results of simulation and real-time experiments of the proposed controller are reported. Finally, in Section VI the conclusions are summarized.

5.2 Preliminaries

This section recalls the forward kinematics of the robotic exoskeleton, as well as the motion capture system (MOCAP) that measures human upper-limb movement.

5.2.1 Robotic exoskeleton forward kinematics

The ETS-MARSE (Motion Assistive Robotic Exoskeleton for Superior Extremity) is a seven degrees of freedom (DoF) redundant exoskeleton robot, designed for upper-limb rehabilitation. For kinematic modeling, the ETS-MARSE's frames were located as shown in Fig. 5.1, then the modified Denavit-Hartenberg parameters were derived as shown in Table 5.1, where d_s is the distance from the origin frame to the shoulder, $d_e \in \mathbb{R}$ is the upper arm length and $d_w \in \mathbb{R}$ is the forearm length. From Table 5.1, the homogeneous transformation matrix between the base frame to the end-effector is obtained as follows:

$$ {}^{0}_{7}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T\,{}^{4}_{5}T\,{}^{5}_{6}T\,{}^{6}_{7}T \tag{5.1} $$

This result is known as forward kinematics, where the homogeneous transformation ${}^{i-1}_{i}T \in \mathbb{R}^{4\times4}$ between the frame $i-1$ and i is defined as:

$$ {}^{i-1}_{i}T = \begin{bmatrix} R_x(\alpha_{i-1}) & 0_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \Delta_{xyz}(a_{i-1}, 0, 0) \begin{bmatrix} R_z(\theta_i) & 0_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \Delta_{xyz}(0, 0, d_i) \tag{5.2} $$

where $R_x \in \mathbb{R}^{3\times3}$, $R_z \in \mathbb{R}^{3\times3}$ are rotation matrices (Goldstein, 1980) and $\Delta_{xyz}(x_\delta, y_\delta, z_\delta) \in \mathbb{R}^{4\times4}$ is a translation matrix given the displacements x_δ, y_δ and z_δ along the x, y and z axis respectively,

defined as follows:

$$\Delta_{xyz}(x_\delta, y_\delta, z_\delta) = \begin{bmatrix} & & x_\delta \\ & I_{3\times3} & y_\delta \\ & & z_\delta \\ 0_{1\times3} & & 1 \end{bmatrix} \tag{5.3}$$

Where $I_{3\times3}$ is the identity matrix. By using each ${}^{i-1}_{i}T$, the end-effector position w is obtained, as well as the elbow position e and shoulder s position, as follows:

$$\begin{bmatrix} s^T & 1 \end{bmatrix}^T = {}^{0}_{1}T \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T \tag{5.4}$$

$$\begin{bmatrix} e^T & 1 \end{bmatrix}^T = {}^{0}_{1}T^{1}_{2}T^{2}_{3}T^{3}_{4}T \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T \tag{5.5}$$

$$\begin{bmatrix} w^T & 1 \end{bmatrix}^T = {}^{0}_{1}T^{1}_{2}T^{2}_{3}T^{3}_{4}T^{4}_{5}T \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T \tag{5.6}$$

where

$$s = \begin{bmatrix} 0 & 0 & d_s \end{bmatrix}^T \tag{5.7}$$

$$e = \begin{bmatrix} d_e c_1 s_2 & d_e s_1 s_2 & d_s + d_e c_2 \end{bmatrix}^T \tag{5.8}$$

$$w = \begin{bmatrix} -d_w(s_4(-c_1c_2c_3 + s_1s_3) - c_1s_2c_4) + d_e c_1 s_2 \\ d_w(s_4(c_1s_3 + s_1c_2c_3) + s_1s_2c_4) + d_e s_1 s_2 \\ d_s + d_w(c_2c_4 - s_2c_3s_4) + d_e c_2 \end{bmatrix} \tag{5.9}$$

here s_i and c_i for $i \in [1, 7]$ are the sine and cosine functions of θ_i.

5.2.2 Motion capture system

The motion capture system (MOCAP) is based on the Myo armband which is a wireless device that includes 8 surface EMG (Electromyography) channels and a 9-axis IMU (Inertial Measurement Unit). The measured information is sent to a host computer via Bluetooth at a 50 ms rate. In this work, the position of the elbow and wrist, and orientation of the forearm and wrist are calculated by using two Myo armbands as shown in Fig. 5.2, obtaining the forward

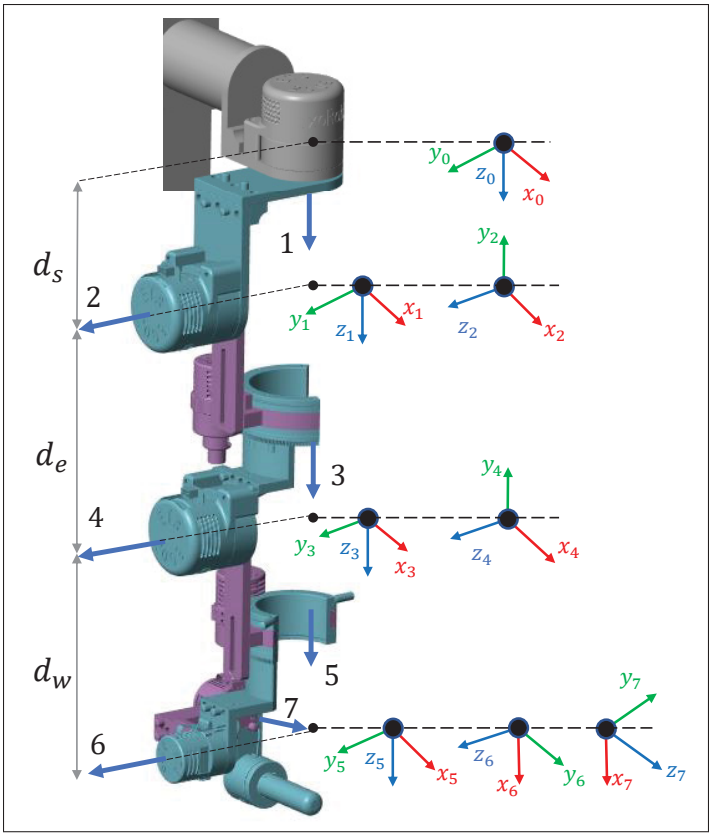

Figure 5.1 Frames location

Table 5.1 Modified D-H parameters

θ_i	a_{i-1}	α_{i-1}	d_i
θ_1	0	0	d_s
θ_2	0	$-\frac{\pi}{2}$	0
θ_3	0	$\frac{\pi}{2}$	d_e
θ_4	0	$-\frac{\pi}{2}$	0
θ_5	0	$\frac{\pi}{2}$	d_w
$\theta_6 - \frac{\pi}{2}$	0	$-\frac{\pi}{2}$	0
θ_7	0	$-\frac{\pi}{2}$	0

human kinematics. The Myo number 1 (MYO 1) armband is located in the upper arm about 5 cm from the olecranon bone to the biceps brachii, and the MYO number 2 (MYO 2) is located

in the forearm about 5 cm from the olecranon bone to the supinator muscle (Ren *et al.*, 2019). When extending the left arm to the front, the position and orientation of both Myo armbands should be as shown in Fig. 5.3.

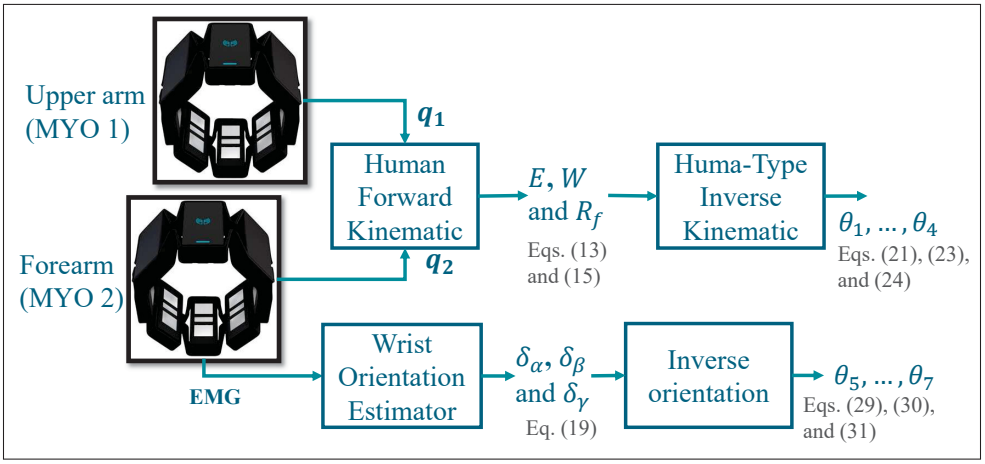

Figure 5.2 Mirror rehabilitation system

5.3 Mirror rehabilitation system

In this section, a mirror rehabilitation system for hemiplegic stroke survivors is presented. The rehabilitation system is able to measure the healthy arm's joint angles and replicate them on the impaired arm.

5.3.1 Human forward kinematics

The MOCAP system provides the following two measured quaternions $\mathbf{q_1}$ and $\mathbf{q_2}$ obtained from the upper arm and forearm respectively.

$$\begin{aligned} \mathbf{q_1} &= [q_{11}, q_{12}, q_{13}, q_{14}]^T \\ \mathbf{q_2} &= [q_{21}, q_{22}, q_{23}, q_{24}]^T \end{aligned} \tag{5.10}$$

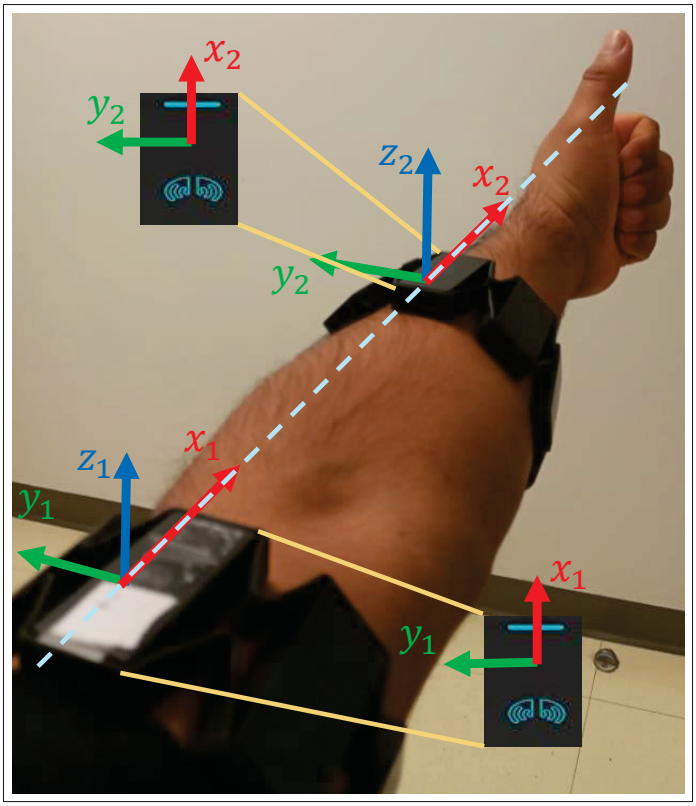

Figure 5.3 Myo armband position and orientation

The homogeneous transformation for the elbow is defined as follows:

$$T_e = \Delta_{xyz}(0, 0, d_s)\, T_q(\mathbf{q_1})\, \Delta_{xyz}(d_e, 0, 0) \tag{5.11}$$

where $T_q(\mathbf{q})$ is the resulting matrix given a unit quaternion $\mathbf{q} = [q_1, q_2, q_3, q_4]^T$ and it is defined as (Kuipers, 1999):

$$T_q(\mathbf{q}) = \begin{bmatrix} 1 - 2q_3^2 - 2q_4^2 & 2q_2q_3 - 2q_1q_4 & 2q_2q_4 + 2q_1q_3 & 0 \\ 2q_2q_3 + 2q_1q_4 & 1 - 2q_2^2 - 2q_4^2 & 2q_3q_4 - 2q_1q_2 & 0 \\ 2q_2q_4 - 2q_1q_3 & 2q_3q_4 + 2q_1q_2 & 1 - 2q_2^2 - 2q_3^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.12}$$

Then T_e is written as:

$$T_e = \begin{bmatrix} R_e & E \\ 0_{1\times3} & 1 \end{bmatrix} \tag{5.13}$$

where $E \in \mathbb{R}^{3\times1}$ is the elbow measured position and $R_e \in \mathbb{R}^{3\times3}$ defines the elbow measured orientation. Following the same procedure, the homogeneous transformation of the wrist is obtained as follows:

$$T_w = \Delta_{xyz}(E_x, E_y, E_z)\, T_q(\mathbf{q_2})\, \Delta_{xyz}(d_w, 0, 0) \tag{5.14}$$

and $T_w \in \mathbb{R}^{4\times4}$ is written as follows:

$$T_w = \begin{bmatrix} R_f & W \\ 0_{1\times3} & 1 \end{bmatrix} \tag{5.15}$$

where $W \in \mathbb{R}^{3\times1}$ is the wrist measured position and $R_f \in \mathbb{R}^{3\times3}$ is the forearm measured orientation. Now, the xyz components of the measured position of the shoulder, elbow, and wrist are defined as follows:

$$\begin{aligned} S &= [0, 0, d_s]^T \\ E &= [E_x, E_y, E_z]^T \\ W &= [W_x, W_y, W_z]^T \end{aligned} \tag{5.16}$$

5.3.2 Wrist orientation estimator

Due to the lack of wrist orientation information in the previous subsection, in this work, a deep convolutional neural network model is used to estimate the wrist orientation from forearm surface EMG signals as shown in Fig. 5.4. To that end, two EMG features called the mean absolute value (*MAV*) (Shi *et al.*, 2018) and the waveform length (*WL*) (Phinyomark *et al.*, 2012) are considered, which are the most popular features for EMG pattern recognition in the literature, both expressed as:

$$MAV_i = \frac{1}{N} \sum_{i=1}^{N} |x_i| \tag{5.17}$$

$$WL_i = \frac{1}{N}\sum_{i=1}^{N} |x_i - x_{i-1}| \tag{5.18}$$

Where $x_i \in \mathbb{R}^8$ is the measured raw EMG signal and N is the number of samples. Seven wrist movements for classification are defined as shown in Fig. 5.5. The input of the deep

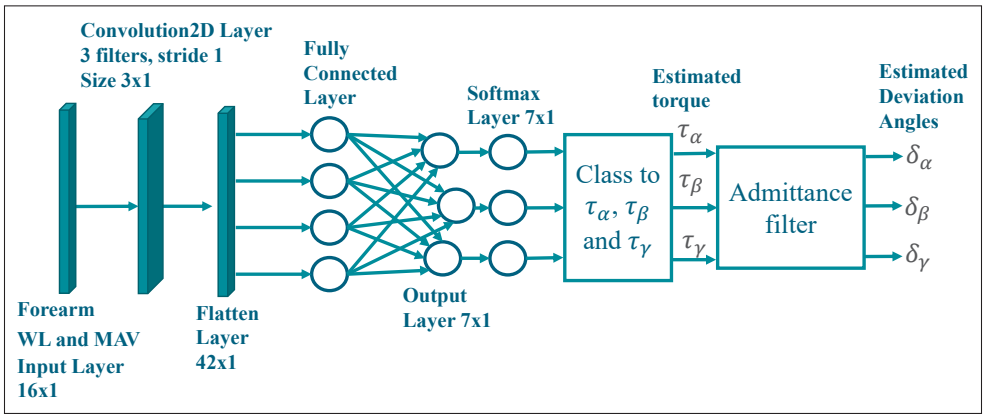

Figure 5.4 Wrist orientation estimator

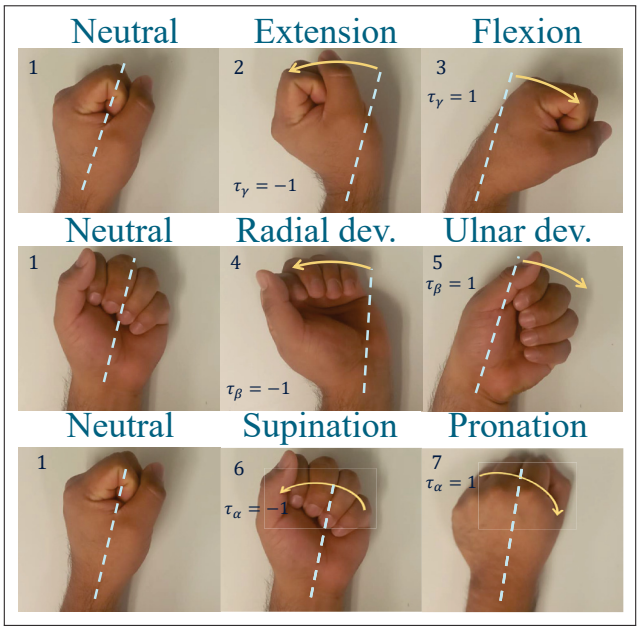

Figure 5.5 Wrist movements for classification

convolutional neural network model is composed of both EMG features *WL* and *MAV* (8 forearm signals each), resulting in a 16-signal input layer, then a 2D convolutional layer extracts the most significant features from all the EMG features using 3 filters with size 3x1. A flattening layer, as shown in Fig. 5.4, converts the information from the filters into a single vector, which is fed into a fully connected layer, then the softmax output layer classifies the movement which is converted in the estimated $\pm\tau_\alpha \in \mathbb{R}$ (forearm pronation/supination), $\pm\tau_\beta \in \mathbb{R}$ (wrist radial/ulnar deviation) and $\pm\tau_\gamma \in \mathbb{R}$ (wrist flexion/extension). In total 313 hyperparameters (9 weights and 3 Bias from the convolution layer and 7x42 weights and 7 bias from the fully connected layer) are tuned during the training. By using an admittance filter (Ochoa Luna *et al.*, 2015) in its differential equation form as shown in (5.19), the following orientation angles are obtained $\delta_\alpha \in \mathbb{R}$, $\delta_\beta \in \mathbb{R}$ and $\delta_\gamma \in \mathbb{R}$ which define the deviation from the forearm orientation.

$$\begin{bmatrix} \dot{\delta}_\alpha \\ \dot{\delta}_\beta \\ \dot{\delta}_\gamma \end{bmatrix} = A_1^{-1} \begin{bmatrix} \tau_\alpha \\ \tau_\beta \\ \tau_\gamma \end{bmatrix} - A_1^{-2} A_2 \begin{bmatrix} \delta_\alpha \\ \delta_\beta \\ \delta_\gamma \end{bmatrix} \tag{5.19}$$

where A_1 and A_2 are diagonal matrices. Since R_f in (5.15) denotes the measured forearm orientation only, the measured wrist orientation is computed using the angles from (5.19).

$$O = T_w \begin{bmatrix} R_z(\delta_\gamma) & 0_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} R_y(\delta_\beta) & 0_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} R_x(\delta_\alpha) & 0_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \tag{5.20}$$

5.3.3 Human-type inverse kinematics

Once the measured positions and orientations are available (E, W, and O), the joint angle of each DoF of the ETS-MARSE are computed. Now, given the measured elbow position E, and using (5.8), the joint angles θ_1 and θ_2 are found as follows:

$$\begin{aligned} \theta_1 &= atan2(E_y, E_x) \\ \theta_2 &= atan2(\pm\sqrt{E_x^2 + E_y^2}, E_z - ds) \end{aligned} \tag{5.21}$$

Note that the elbow position for the ETS-MARSE only depends on θ_1 and θ_2. The value of θ_3 is also found analytically, substituting w by W in (5.6) and multiplying both sides by $({}^0_1T\,{}^1_2T)^{-1}$ yields:

$$\begin{bmatrix} d_w c_3 s_4 \\ -d_e - d_w c_4 \\ d_w s_3 s_4 \\ 1 \end{bmatrix} = \begin{bmatrix} W_x c_1 c_2 + W_y s_1 c_2 - W_z s_2 + d_s s_2 \\ -W_x c_1 s_2 - W_y s_1 s_2 - W_z c_2 + d_s c_2 \\ -W_x s_1 + W_y c_1 \\ 1 \end{bmatrix} \tag{5.22}$$

Then

$$\theta_3 = atan2(-W_x s_1 + W_y c_1, W_x c_1 c_2 + W_y s_1 c_2 - W_z s_2 + d_s s_2) \tag{5.23}$$

The value of θ_4 is obtained by:

$$\theta_4 = \pi \pm acos\left(\frac{d_w^2 + d_e^2 - ||W - S||^2}{2 d_e d_w}\right) \tag{5.24}$$

5.3.4 Inverse orientation

By using the measured desired orientation O 5.20, the joint angle for the last three joints θ_5, θ_6 and θ_7 are computed. It can be noted that the matrix ${}^4_7T$ contains the orientation of the end-effector.

$${}^4_7T = {}^4_5T\,{}^5_6T\,{}^6_7T \tag{5.25}$$

The matrix ${}^4_7T$ yields:

$${}^4_7T = \begin{bmatrix} s_5 s_7 + c_5 s_6 c_7 & s_5 c_7 - c_5 s_6 s_7 & c_5 c_6 & 0 \\ -c_6 c_7 & c_6 s_7 & s_6 & -d_w \\ -c_5 s_7 + s_5 s_6 c_7 & -c_5 c_7 - s_5 s_6 s_7 & s_5 c_6 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.26}$$

Using the matrices O and ${}^{4}_{7}T$ the following relation is built

$$ {}^{4}_{7}T = {}^{0}_{4}T^{-1} O \tag{5.27} $$

Developing the right-hand part of (5.27), one obtains:

$$ {}^{4}_{7}T = \begin{bmatrix} \underline{o}_{11} & \underline{o}_{12} & \underline{o}_{13} & \underline{o}_{14} \\ \underline{o}_{21} & \underline{o}_{22} & \underline{o}_{23} & \underline{o}_{24} \\ \underline{o}_{31} & \underline{o}_{32} & \underline{o}_{33} & \underline{o}_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.28} $$

Considering (5.26) and (5.28), the solution for θ_6 is:

$$ \theta_6 = atan2\left(\underline{o}_{23}, c_6\right) \tag{5.29} $$

where $c_6 = \pm\sqrt{\underline{o}_{13}^2 + \underline{o}_{33}^2}$. The solution for θ_5 is:

$$ \theta_5 = atan2\left(\underline{o}_{33}, \underline{o}_{13}\right) \tag{5.30} $$

The solution of θ_7 is:

$$ \theta_7 = atan2\left(\underline{o}_{22}, -\underline{o}_{21}\right) \tag{5.31} $$

Remark 1. *Note that the previously calculated solutions remain valid when* $|\theta_6| < \frac{\pi}{2}$*, otherwise the wrist ulnar or radial deviation will be highly uncomfortable.*

5.4 Trajectory tracking controller design

In this section, a fixed-time super-twisting-like algorithm with prescribed performance (see Fig. 5.6) is developed for the ETS-MARSE exoskeleton to follow the measured trajectory from the healthy arm mirrored to the impaired arm.

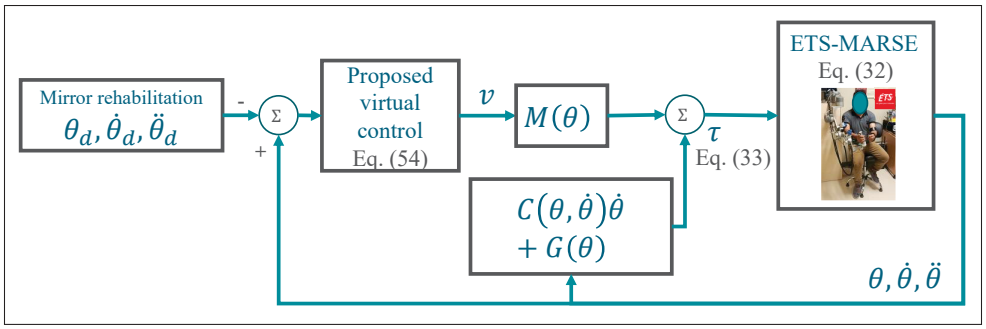

Figure 5.6 Overall system's block diagram

5.4.1 ETS-MARSE dynamics

The dynamics of the studied exoskeleton is given in the following standard form:

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) = \tau + \tau_u \tag{5.32}$$

where $\theta \in \mathbb{R}^n$, $\dot{\theta} \in \mathbb{R}^n$ and $\ddot{\theta} \in \mathbb{R}^n$ are the joint angular position, velocity, and acceleration respectively, $M(\theta) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $C(\theta,\dot{\theta}) \in \mathbb{R}^{n\times n}$ is the Coriolis and centrifugal matrix, $G(\theta) \in \mathbb{R}^n$ is the gravitational force vector, τ is the torque applied by the motors and τ_u is the unknown dynamics e.g. external disturbances, model uncertainties and human-robot interaction forces. Notice that n is the number of DoF and is equal to 7 in our case. By using the known dynamics of (5.32), a nominal control law is obtained as follows:

$$\tau = M(\theta)v + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) \tag{5.33}$$

where v is a virtual control input that will be designed in the subsequent part. Substituting (5.33) into (5.32) yields:

$$\ddot{\theta} = v + w_u \tag{5.34}$$

where $w_u = M^{-1}(\theta)\tau_u$ denotes the uncertain vector with bounded initial functions such as:

$$|w_{ui}(t_0)| < W_{0i} \tag{5.35}$$

where W_{0i} is a positive constant for $i = 1 \rightarrow n$. In addition w_u verifies the following Lipschitz condition:

$$|\dot{w}_{ui}(t)| < W_i \tag{5.36}$$

where W_i is a known positive constant for $i = 1 \rightarrow n$.

5.4.2 Fixed-time super-twisting-like control with prescribed performance

For the controller design procedure, some useful results are presented:

Lemma 1. *(refer to (Chen* et al.*, 2020)) If there exists a function $V : \mathbb{R}^n \rightarrow \mathbb{R}_+ \cup \{0\}$ such that*

$$\dot{V}(x) \leq -\left(a_1 V^{\kappa_1}(x) + a_2 V^{\kappa_2}(x)\right)^{\kappa_3}, \forall x \neq 0 \tag{5.37}$$

where a_1, a_2, a_3, κ_1, κ_2 and κ_3 are positive constants with $\kappa_1\kappa_3 > 1$ and $0 < \kappa_2\kappa_3 < 1$. Then a fixed-time stability is achieved, and its maximum settling time is estimated by:

$$T_{max} = \frac{1}{a_1^{\kappa_3}(\kappa_1\kappa_3 - 1)} - \frac{1}{a_2^{\kappa_3}(\kappa_2\kappa_3 - 1)} \tag{5.38}$$

Lemma 2. *(refer to (Basin* et al.*, 2019)) If there is a scalar system that can be written in the following differential inclusion form:*

$$\begin{aligned} \dot{y} &= -a_1|y|^{0.5}\text{sign}(y) - a_2|y|^{\alpha}\text{sign}(y) + z \\ \dot{z} &= -a_3\text{sign}(y) + \dot{x} \end{aligned} \tag{5.39}$$

where x is a Lipschitz function such as $\dot{x} \leq \Delta X$, $a_1, a_2 > 0$, $a_3 > \Delta X$, and $\alpha > 1$. Then, y converges to the equilibrium point in fixed-time smaller than:

$$T \leq \left(\frac{1}{a_2(\alpha - 1)\xi^{\alpha-1}} + \frac{2\xi^{0.5}}{a_1} + \frac{\Delta X_0}{M}\right)\left(1 + \frac{1}{\left(\frac{m}{M} - \frac{mh(a_1)}{a_1}\right)}\right) + \frac{\Delta X_0}{m} \tag{5.40}$$

where ΔX_0 is the upper bound of the initial value of the Lipschitz function x at the initial time $t = t_0$, $M = a_3 + \Delta X$, $m = a_3 - \Delta X$, $h(a_1) = \frac{1}{a_1} + \left(\frac{2\varepsilon}{ma_1}\right)^{1/3}$, with ε as the base of natural logarithms, $\xi = \left(\frac{a_1}{a_2}\right)^{\frac{2}{1+2\alpha}}$.

A novel controller is proposed for the ETS-MARSE robot using a fixed-time sliding surface with prescribed performance and a super-twisting-like algorithm. The chosen prescribed performance function (Bu, 2023) is shown below:

$$\eta(t) = \eta_\infty + (\eta_0 - \eta_\infty)\exp(-lt) \tag{5.41}$$

where η_0 and η_∞ are the values of $\eta(t)$ at time $t = 0$ and $t \to \infty$, respectively. Let us define the i^{th} constrained error as follows:

$$-\eta(t) < e_i = \theta_i - \theta_{id} < \eta(t) \tag{5.42}$$

where θ_{id} is the i^{th} desired reference. The constrained error is transformed into an unconstrained one $z = [z_1, .., z_n]^T$ with first and second derivatives defined as:

$$\begin{aligned} z_i &= \frac{1}{2}\ln\left(\frac{\frac{e_i}{\eta(t)} + 1}{1 - \frac{e_i}{\eta(t)}}\right) \\ \dot{z}_i &= r_i H_{1i} \\ \ddot{z}_i &= \dot{r}_i H_{1i} + r_i\left(\ddot{e}_i + H_{2i}\right) \end{aligned} \tag{5.43}$$

where H_{1i} and H_{2i} are defined below:

$$\begin{aligned} H_{1i} &= \dot{e}_i - \frac{e_i\dot{\eta}(t)}{\eta(t)} \\ H_{2i} &= \frac{e_i\dot{\eta}(t)^2}{\eta(t)^2} - \frac{e_i\ddot{\eta}(t)}{\eta(t)} - \frac{\dot{e}_i\dot{\eta}(t)}{\eta(t)} \end{aligned} \tag{5.44}$$

And $r \in \mathbb{R}^{n\times n}$ and its derivative are given by:

$$\begin{aligned} r_i &= \frac{1}{2\eta(t)\left(1+\frac{e_i}{\eta(t)}\right)} + \frac{1}{2\eta(t)\left(1-\frac{e_i}{\eta(t)}\right)} \\ \dot{r}_i &= -\frac{\left(\eta(t)^2+e_i^2\right)\dot{\eta}(t) - 2\eta(t)e_i\dot{e}_i}{(e+\eta(t))^2(e-\eta(t))^2} \end{aligned} \tag{5.45}$$

The following fixed-time switching sliding surface is proposed.

$$s = \dot{z} + D(z) \tag{5.46}$$

And its derivative yields:

$$\dot{s} = \ddot{z} + \dot{D}(z) \tag{5.47}$$

where $D(z) = [D_1(z_1), \cdots, D_n(z_n)]^T$ and $\dot{D}(z) = [\dot{D}_1(z_1), \cdots, \dot{D}_n(z_n)]^T$ with:

$$\begin{aligned} D_i(z_i) =& \lambda_{1i}|z_i|^{\alpha_1}\text{sign}(z_i) + \lambda_{2i}|z_i|^{\alpha_2}\text{sign}(z_i) \\ \dot{D}_i(z_i) =& \lambda_{1i}\alpha_1|z_i|^{\alpha_1-1}\dot{z}_i + \lambda_{2i}\alpha_2|z_i|^{\alpha_2-1}\dot{z}_i \end{aligned} \tag{5.48}$$

The design parameters λ_{1i}, λ_{2i}, α_1 and α_2 are positive constants with $\alpha_1 > 1$ and $0.5 < \alpha_2 < 1$ and:

$$\text{sign}(z_i) = \begin{cases} -1, & \text{if } z_i < 0, \\ 1, & \text{if } z_i > 0, \\ 0, & \text{otherwise} \end{cases} \tag{5.49}$$

The proposed nonlinear sliding surface ensures fixed-time convergence during the reaching phase when $s = 0$ (i.e., $\dot{z} = -D(z)$). The maximal convergence time is settled based on the

following Lyapunov stability study where:

$$\begin{aligned} V_s =& \frac{1}{2} z^T z \\ \dot{V}_s =& z^T \dot{z} = -z^T D(z) \\ \dot{V}_{si} =& -\lambda_{1i}|z_i|^{\alpha_1+1} - \lambda_{2i}|z_i|^{\alpha_2+1} \\ =& -\lambda_{1i}(2V_{si})^{\frac{\alpha_1+1}{2}} - \lambda_{2i}(2V_{si})^{\frac{\alpha_2+1}{2}} \leq 0 \end{aligned} \tag{5.50}$$

According to Lemma 1, each transformed error (z_i for $i = 1 \rightarrow n$) identically each tracking error (e_i for $i = 1 \rightarrow n$) converge to zero within fixed-time of which the maximum is estimated as follows:

$$T_{max}^{sl} = T_{max}^{cn} + \frac{1}{\lambda_{1i} 2^{\frac{\alpha_1-1}{2}} (\alpha_1 - 1)} - \frac{1}{\lambda_{2i} 2^{\frac{\alpha_2-1}{2}} (\alpha_2 - 1)} \tag{5.51}$$

where T_{max}^{cn} is the time required to reach $s = 0$. Otherwise, using the nominal model in (5.34) and setting $\dot{s} = 0$, the equivalent control is computed as follows:

$$\nu_0 = \ddot{\theta}_d - H_2 - r^{-1} \left(\dot{D}(z) + \dot{r} H_1 \right) \tag{5.52}$$

where $H_1 = [H_{11}, \cdots, H_{1n}]^T, H_2 = [H_{21}, \cdots, H_{2n}]^T, r = \text{diag}(r_1, \cdots, r_n)$ and $\dot{r} = \text{diag}(\dot{r}_1, \cdots, \dot{r}_n)$.

Moreover, to ensure a fixed-time convergence of the states to the fixed-time sliding surface, the super-twisting-like control algorithm is used (Basin *et al.*, 2019).

$$\nu_1 = -r^{-1} \left(K_1 \text{sig}^{\frac{1}{2}}(s) + K_2 \text{sig}^p(s) + \int_0^t K_3 \text{sig}^0(s) dt \right) \tag{5.53}$$

where $\text{sig}^*(s) = [|s_1|^* \text{sign}(s_1), ..., |s_n|^* \text{sign}(s_n)]^T$, the design parameters are $p \in \mathbb{R}^+$ and K_1, K_2, K_3 as $(n \times n)$ diagonal positive matrices. Finally, the total virtual control law is given as:

$$\nu = \nu_0 + \nu_1 \tag{5.54}$$

By substituting (5.54) in (5.34) results in the following closed-loop control system:

$$\begin{aligned} \dot{s} &= -K_1 \text{sig}^{\frac{1}{2}}(s) - K_2 \text{sig}^{p}(s) + y \\ \dot{y} &= -K_3 \text{sig}^{0}(s) + \dot{w}_u \end{aligned} \tag{5.55}$$

By means of *Lemma* 2, considering $K_{3i} > W_i$, each sliding surface achieves a fixed-time stability with a time smaller than:

$$T_{max}^{cn} = \left(\frac{1}{K_{2i}(p-1)\xi^{p-1}} + \frac{2\xi^{0.5}}{K_{1i}} + \frac{W_{0i}}{M} \right) \left(1 + \frac{1}{\left(\frac{m}{M} - \frac{m f(K_{1i})}{K_{1i}} \right)} \right) + \frac{W_{0i}}{m} \tag{5.56}$$

where $M = K_{3i} + W_i$, $m = K_{3i} - W_i$, $\xi = \left(\frac{K_{1i}}{K_{2i}} \right)^{\frac{2}{1+2p}}$ and $f(K_{1i}) = \frac{1}{K_{1i}} + \left(\frac{2\varepsilon}{mK_{1i}} \right)^{1/3}$, with ε as the base of natural logarithms.

Remark 2. *A singularity might occur during the sliding phase when $z_i = 0$ and $z_i \neq 0$. To avoid this problem, the term $A_i = |z_i|^{\alpha_2 - 1}\dot{z}_i$ is replaced by* $\text{sat}\,(A_i, \delta_i)$ *such as:*

$$\text{sat}(A_i, \delta_i) = \begin{cases} |z_i|^{\alpha_2 - 1}\dot{z}_i & \text{if } ||z_i|^{\alpha_2 - 1}\dot{z}_i| < \delta_i \\ \delta_i \text{sign}(|z_i|^{\alpha_2 - 1}\dot{z}_i) & \text{if } ||z_i|^{\alpha_2 - 1}\dot{z}_i| \geq \delta_i \end{cases} \tag{5.57}$$

where $\delta_i < 1$ is a very small positive constant.

5.5 Mirror rehabilitation implementation

5.5.1 Wrist orientation estimator setup and results

For the wrist orientation estimator, the matrices $A_1 = \text{diag}(5, 5, 5)$ and $A_2 = \text{diag}(0.1, 0.1, 0.1)$ are considered. Five hundred raw EMG measurements were recorded for each wrist movement (3500 EMG measurements in total), and several trials using the *MAV* and *WL* features were performed using 70% for training, 15% for validation and 15% for test as shown in Table 5.2. For the deep convolutional neural network, a total of 100,000 training iterations were chosen,

Table 5.2 EMG pattern recognition trials

	Feature	Flexion/ Extension	Ulnar/ Radial Deviation	Forearm Pronation/ Supination
Deep	$[WL_i\ WL_{i-1}]$	100%	99.71%	99.13%
neural	$[MAV_i\ MAV_{i-1}]$	99.71%	99.71%	99.42%
model	$[WL_i\ MAV_i]$	98.85%	99.42%	99.14%
Gaussian	$[WL_i\ WL_{i-1}]$	97.14%	98.32%	93.10%
processes	$[MAV_i\ MAV_{i-1}]$	99.10%	98.81%	85.62%
(Rasmussen & Nickisch, 2010)	$[WL_i\ MAV_i]$	97.45%	98.97%	89.22%

and for the Gaussian Processes a model size of 50 points (Rasmussen & Nickisch, 2010) and 120,000 training iterations were chosen.

5.5.2 Controller simulation setup

For simulation, the Simscape Multibody Environment (formerly SimMechanics) (MathWorks) was used. A demanding scenario has been configured for the controller, to that end the physical parameters of the Simscape model were chosen with 160% more than in Table 5.3 to serve as unknown dynamics. Additionally, using the MOCAP system described in **Section 5.2**, a challenging trajectory generated by a series of movements was recorded within 20 seconds. The sequence starts with the initial position, followed by a forearm pronation/supination, and concludes by taking a highlighter pen and depositing it into a container (Fig. 5.7). The controller design parameters are: For the prescribed function $\eta_0 = 1.7279$, $\eta_\infty = 0.09$, $l = 1.5$. For the fixed-time sliding surface $\lambda_1 = 20$, $\lambda_2 = 20$, $\lambda_3 = 20$, $\alpha_1 = 1.2$, $\alpha_2 = 1.3$, $\alpha_3 = 0.7$, $\delta = 0.1$, $p = 2$, $K_1 = [0.2,\ 0.2,\ 1,\ 0.2,\ 1,\ 1,\ 5]^T$, $K_2 = K_1$, and $K_3 = 10K_1$.

5.5.3 Controller simulation results

The trajectory tracking results are presented in Fig. 5.8 where several forearm pronation/supination movements were detected. It is observed that all the joints, even when starting from a

Table 5.3 ETS-MARSE physical parameters

Joint	Link Length (m)	Mass (kg)	Centre of mass (m)
1	0.145	3.4	0.0968
2	0	0	0.193
3	0.25	3.4	0
4	0	0	0.164
5	0.267	1.98	0
6	0	0.78	0.0417
7	0	0.5	0.0622

Table 5.4 Simulation results

i	1	2	3	4	5	6	7
rms(e_i) (rad)	0.1306	0.1304	0.0799	0.1372	0.0792	0.0714	0.0563
T^{sl} (sec)	5.24	5.38	0.72	11.05	0.32	0.196	0.102
T^{sl}_{max} (sec)	15.2	15.2	3.1	15.2	2.88	2.88	0.3958
OS%	0	0	2.03	0	2.09	5.24	3.93

90-degree offset, reach a small region around the desired reference position in fixed-time as shown in Table 5.4, this remains true even when tracking a challenging trajectory, with a minimal overshoot and a short 5% settling time. To provide a more detailed view of the results, a zoomed tracking error is displayed in Fig. 5.10. It's worth noting that all these errors remain within the specified constraints outlined in (5.42), even in the presence of unknown dynamics. The control signals exhibited in Fig. 5.9, present chattering free behavior. During the initial stage of the simulation, there is a notable increase in torque due to the initial offset, which is necessary to facilitate fixed-time convergence. In a real-time implementation with a human subject, the constraints detailed in (5.42) should be designed to maintain errors within an acceptable small range. A summary of the simulation results is presented in Table 5.4, where these values represent the mean of the RMS error for each DoF.

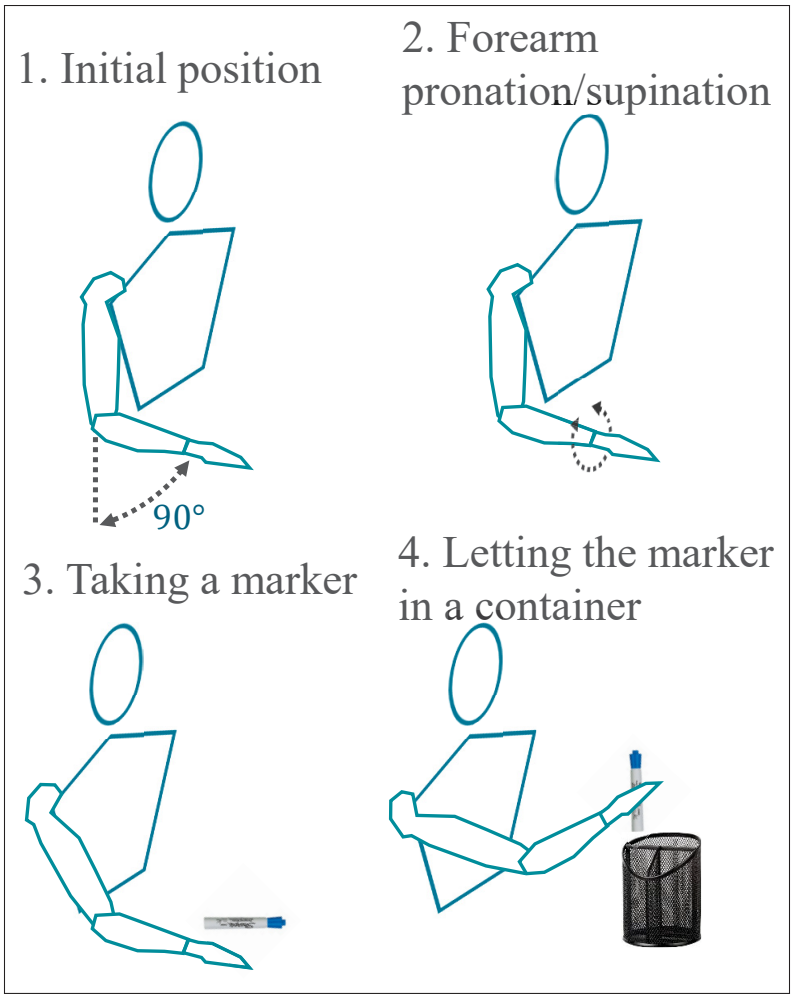

Figure 5.7 Recorded movements for simulation

5.5.4 Controller experimental setup

The real-time system is composed of several processing units as shown in Fig. 5.11. The host PC executes a user interface and stores the data from completed trials. The host PC also receives the measured human movement from the MYO armbands at a 50 ms rate and sends this information to the NI PXI-8108 where the mirror system and the proposed controller are executed at 1 ms rate. Then the control signal is sent to the NI PXI-7813R which regulates the motor current at 50 μs rate and reads the joint's angle position from the hall sensors.

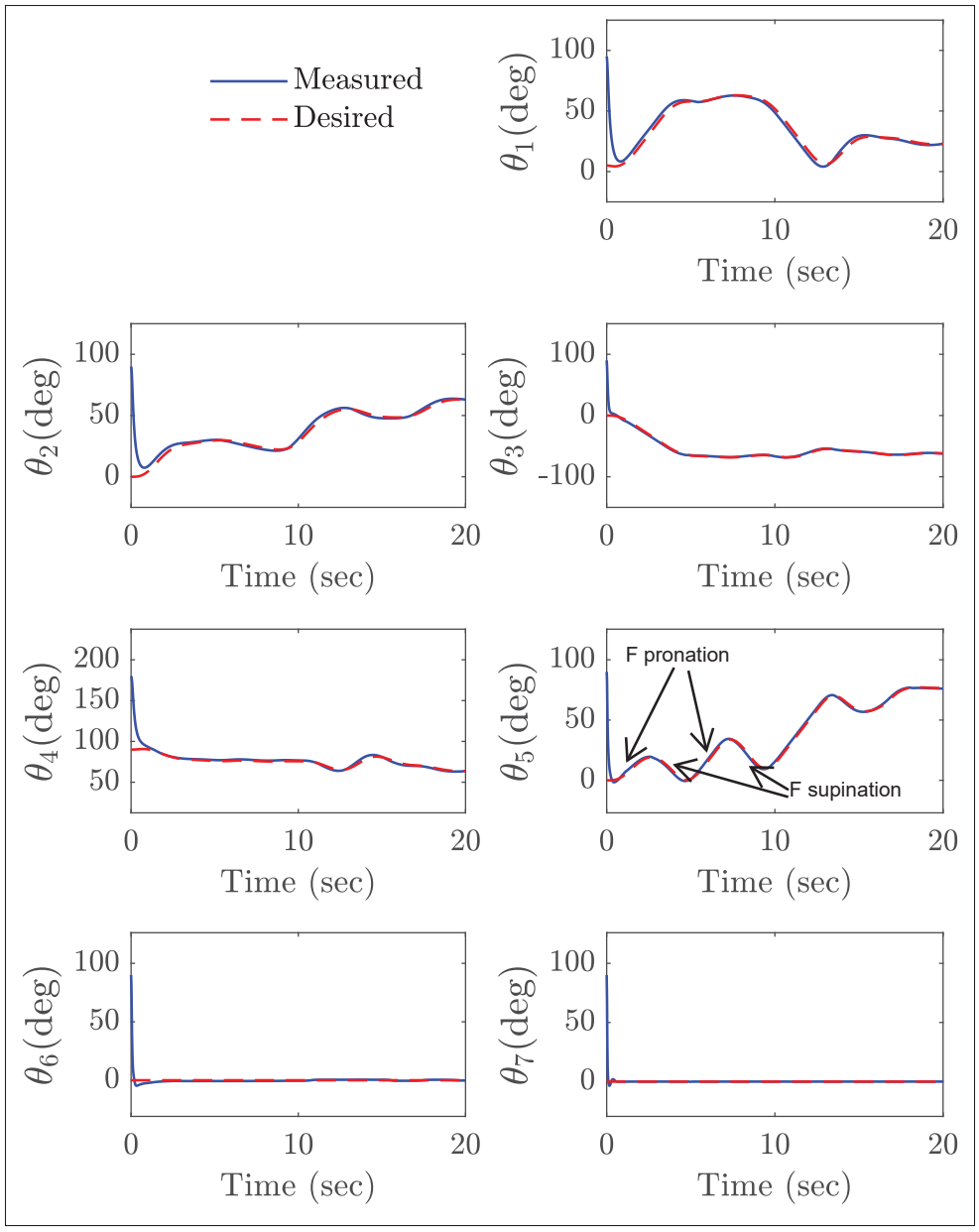

Figure 5.8 Simulation results: trajectory tracking

For the controller real-time test, a healthy subject (32 years old, 75 kg, 1.6 m height) performed a sequence of movements as in Fig. 5.12. The movements begin with the arm fully extended to the front. Then the elbow and shoulder simultaneously move to the second pose, where a wrist flexion/extension movement is estimated from the EMG signals, and finally, a shoulder

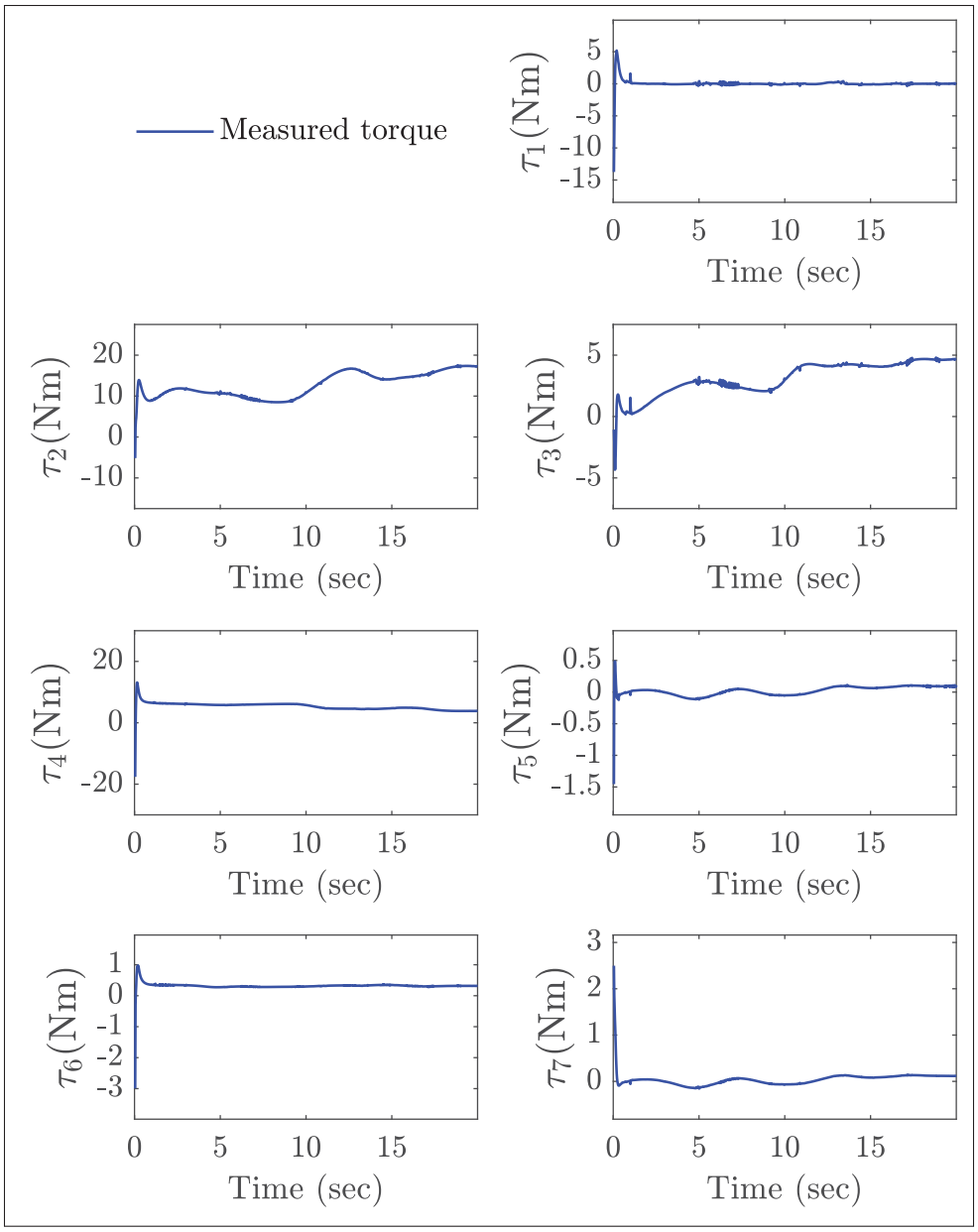

Figure 5.9 Simulation results: control signal

abduction is performed. The controller design parameters are: For the prescribed performance function $\eta_0 = 0.1$, $\eta_\infty = 0.03$, and $l = 1.1$. For the fixed-time sliding surface $\lambda_1 = 20$, $\lambda_2 = 20$, $\lambda_3 = 20$, $\alpha_1 = 1.2$, $\alpha_2 = 1.3$, $\alpha_3 = 0.7$, $\delta = 0.1$, $p = 2$, $K_1 = [10,\ 10,\ 10,\ 10,\ 10,\ 10,\ 30]^T$, $K_2 = K_1$, and $K_3 = [1.5,\ 1,\ 1,\ 1,\ 1,\ 1,\ 2.5]^T$.

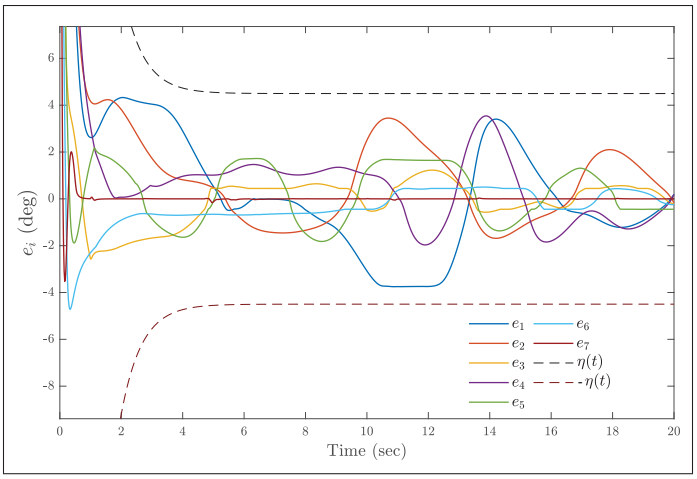

Figure 5.10 Simulation results: zoomed tracking error

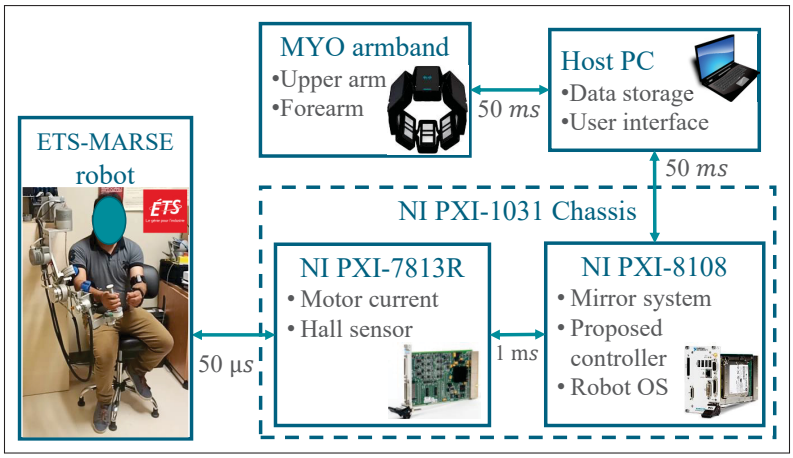

Figure 5.11 Real-time hardware setup

5.5.5 Controller experimental results

The trajectory tracking is shown in Fig. 5.13 where a continuous and smooth desired trajectory was generated thanks to the human-type inverse kinematics solution. The wrist flexion/extension movement is measured by the wrist orientation estimator as shown in joint 7. Despite multiple joints being active during the exercises while following a sharply changing trajectory, the

Table 5.5 Real-time results

DoF	1	2	3	4	5	6	7
rms_{e_i} (rad)	0.0099	0.0083	0.0063	0.0135	0.0125	0.0085	0.0079
rms_{τ_i} (Nm)	6.4374	21.5517	5.1764	9.3096	2.2960	1.5568	0.9007

tracking error is inside the constraints (5.42) as shown in Fig. 5.14. The control activity in Fig. 5.15 presents a low chattering on the joints which is imperceptible to the user as shown in https://youtu.be/sCrBGMI05y8. Finally, the real-time results are summarized in Table 5.5.

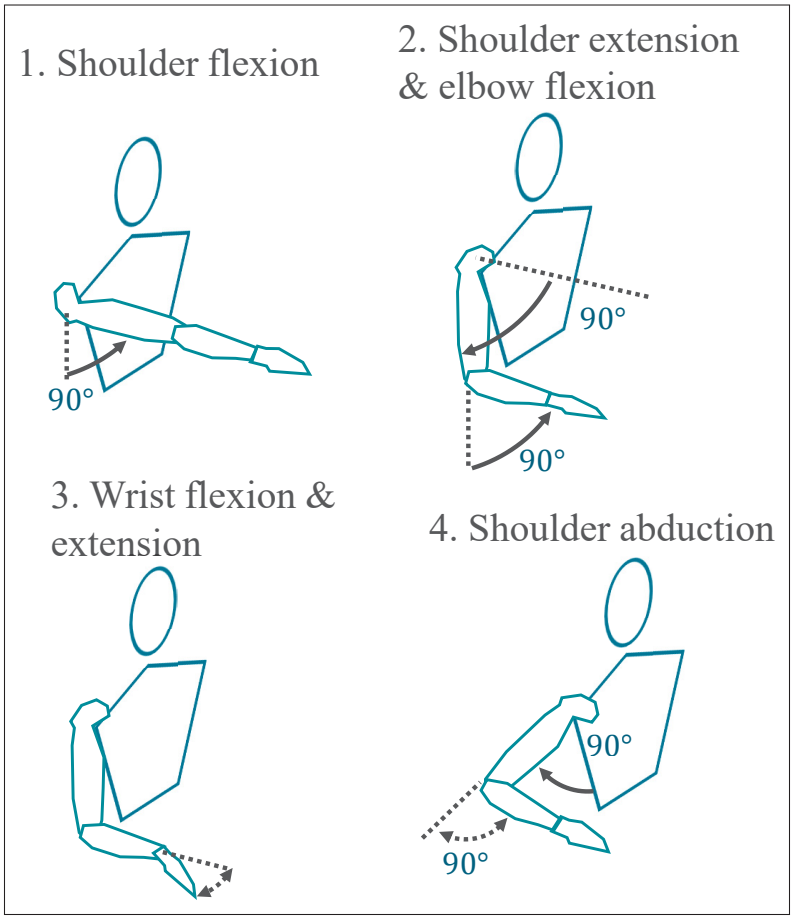

Figure 5.12 Movements for real-time test

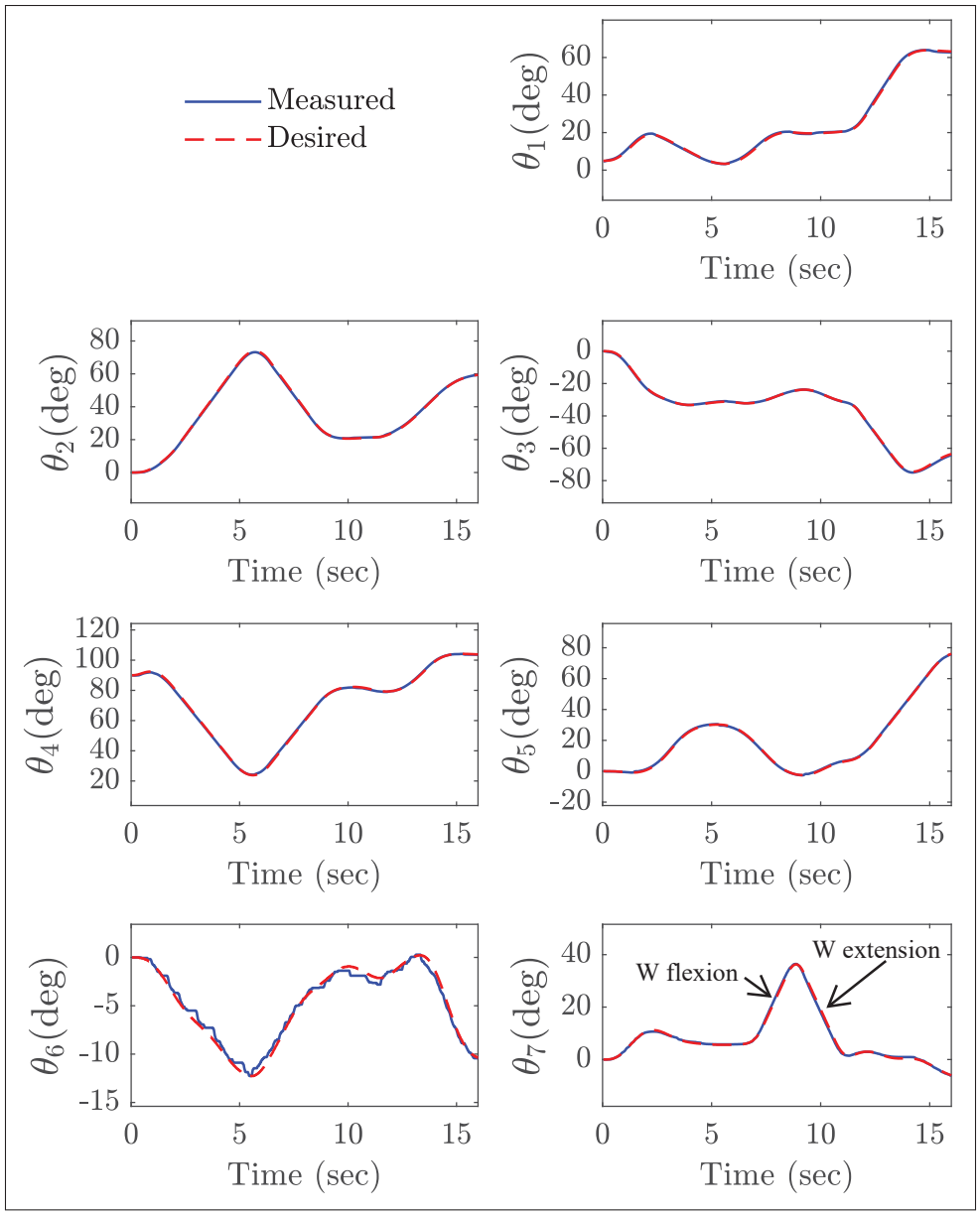

Figure 5.13 Real-time results: trajectory tracking

5.6 Conclusion

The resultant mirror rehabilitation system assists the impaired arm for hemiplegic patients and it can be easily modified for correction or resistance exercises. The rehabilitation system was designed for the ETS-MARSE robot but it is implementable on any redundant robot that has the

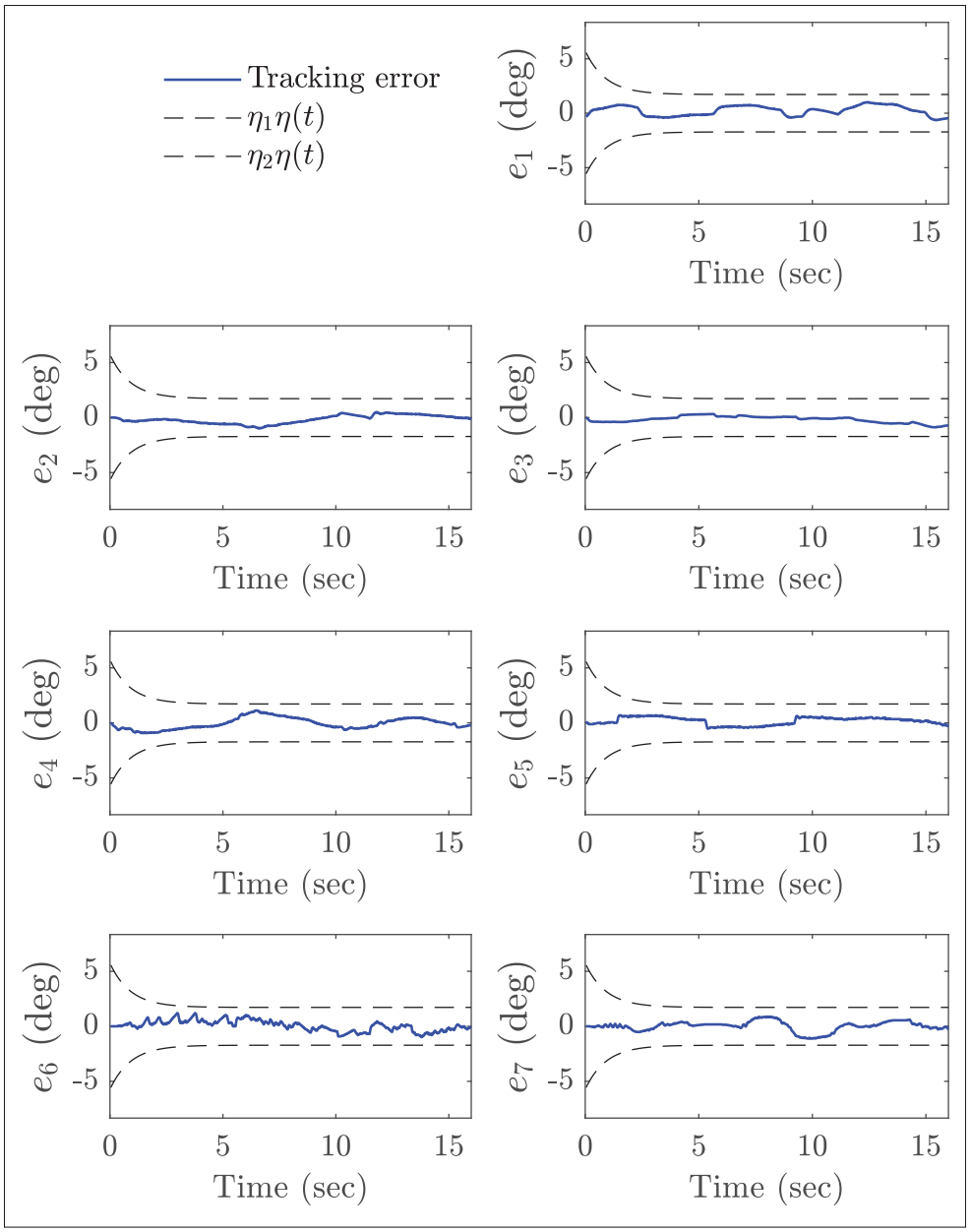

Figure 5.14 Real-time results: tracking error

dexterity and the capability to anthropomorphically situate the elbow and forearm, as well as, the wrist orientation. The wrist orientation estimator, which relies solely on a deep neural model or the Gaussian Process technique, demonstrates its efficacy in accurately classifying movements based on EMG signals. The proposed controller showed good results in both simulation and

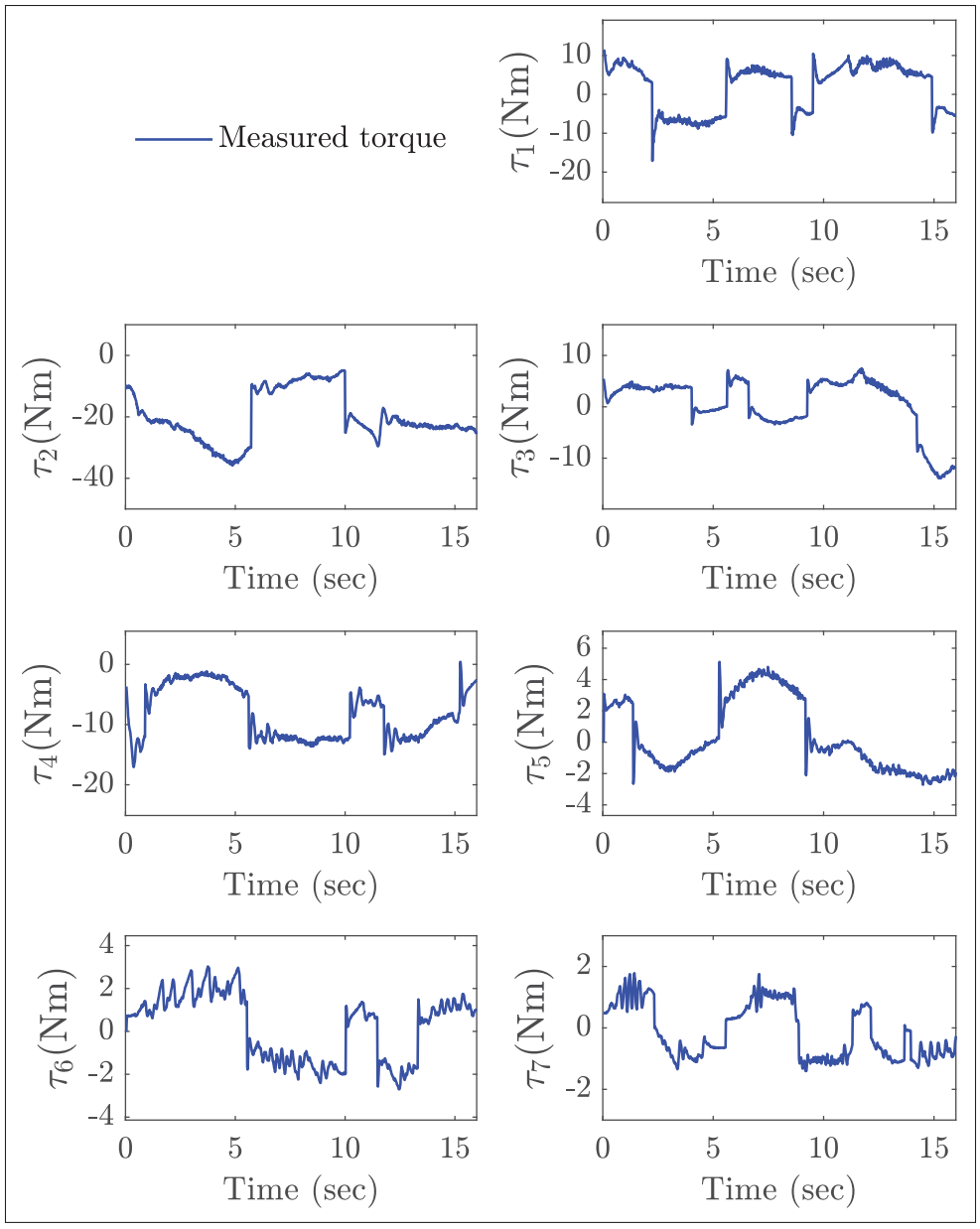

Figure 5.15 Real-time results: control signal

real-time implementation while following two sharply changing trajectories. The fixed-time reaching was demonstrated in the simulation. In future research, a teleoperation system utilizing IMU sensors will be developed to facilitate remote passive upper-limb rehabilitation, thereby establishing a connection between patients and rehabilitation experts.

Acknowledgments

The first author would like to thank to CONAHCYT (Mexican National Council of Humanities, Science and Technology) for the support under grant 739833. CONAHCYT was previously named CONACYT (Mexican National Council of Science and Technology).

CONCLUSION AND RECOMMENDATIONS

This thesis focused on the design and implementation of several improvements based on learning for upper-limb rehabilitation using the ETS-MARSE exoskeleton robot, which was designed and built in the GREPCI robotics laboratory. The results of this project are summarized as follows:

- In Chapter 3, a novel approach to inverse kinematics, based on a machine learning technique, has been introduced and compared with two conventional methods. The comparison between the Jacobian approach, the discomfort index approach, and the Gaussian Process for swivel angle estimation reveals the superior performance of the Gaussian Process method. It excels in estimating swivel angles for upper-limb motions, even when confronted with data not included in the training dataset. The proposed swivel angle estimation approach using Gaussian Process not only exhibits high accuracy but also boasts low computational complexity, making it a pragmatic choice for real-time applications, particularly in the context of upper-limb rehabilitation employing redundant robotic exoskeletons.

  Chapter 3 sets the stage for future work by proposing the utilization of the uncertainty computed in the Gaussian Process output to create safer trajectories for upper-limb rehabilitation. This approach holds promise for enhancing the safety and efficacy of rehabilitation exercises, considering the variability in human movement.

  The introduction of a robust Model Predictive Control (MPC) incorporating an Integral Sliding Mode (ISM) fusion for trajectory tracking during rehabilitation exercises represents a significant contribution. This approach ensures the satisfaction of constraints with high accuracy while striking a balance between tracking performance and control effort. The empirical validation in Chapter 3, including tests on three different upper-limb motions, both with and without subjects, reaffirms the effectiveness of the proposed controller, highlighting the practical applicability of the presented methods.

  Recommendations include further refinement of the Gaussian Process approach, including the expansion of the training dataset to encompass a broader spectrum of upper-limb motions

and scenarios. This expansion can bolster generalization and robustness in swivel angle estimation across diverse rehabilitation exercises.

In the pursuit of generating safer trajectories for upper-limb rehabilitation, a comprehensive exploration of safety measures is imperative. It is essential to develop methods that account for unforeseen movements or perturbations during rehabilitation exercises.

As part of future work, addressing the high dimensionality challenge in the 7-DoF exoskeleton ETS-MARSE is of paramount importance. Strategies for optimizing the performance of the Gaussian Process technique in such complex systems should be thoroughly investigated.

- In Chapter 4, we introduce a novel approach that combines Learning-based Model Predictive Control (MPC) with Integral Sliding Mode (ISM) to enable safe and effective robotic rehabilitation for individuals with upper limb impairment. This innovative method is designed to tackle the complex issues posed by unknown patient efforts and unmodeled physiological characteristics, offering a promising solution for rehabilitation.

  Our experimental results, featuring a healthy subject, convincingly demonstrate the superior effectiveness of the proposed controller when compared to traditional Sliding Mode Control (SMC) and SMC with Exponential Reaching Law (ERL). This approach stands out in terms of its exceptional tracking precision and reduction of control chattering, underscoring its potential to elevate the quality of rehabilitation.

  Chapter 4 emphasizes the adaptive nature of the MPC, showcasing its ability to learn and improve from trial to trial. This adaptability is a pivotal advantage, allowing the controller to refine its performance over time, potentially customizing the rehabilitation process to cater to individual needs and progress.

  The chapter also sheds light on the implementation of this technique in highly nonlinear robotic exoskeletons, affirming its versatility and relevance in addressing the intricate challenges often encountered in rehabilitation scenarios.

As a forward-looking recommendation, we encourage the exploration of a Cautious MPC approach in future research endeavors. This approach holds the promise of further enhancing the safety aspects of rehabilitation tasks, particularly in the context of dealing with unknown patient efforts and unmodeled physiological characteristics. Future research should aim to develop methods that proactively anticipate and mitigate potential risks.

In conclusion, Chapter 4 introduces a promising approach to robotic rehabilitation, effectively addressing the complexities arising from unknown patient efforts and unmodeled physiological characteristics. Its superior performance and adaptability position it as a valuable contribution to the field of rehabilitation robotics. The recommendations presented here are geared towards amplifying the safety, applicability, and real-world integration of this innovative approach within clinical rehabilitation practices.

- Chapter 5 introduces an innovative mirror rehabilitation system tailored for individuals experiencing hemiplegia following a stroke. This system intricately utilizes the movement of the unaffected arm, captured by a Motion Capture (MoCap) system, to guide the impaired arm with the assistance of an upper-limb robotic exoskeleton. The result is a comfortable arm configuration customized to address the distinct needs of the targeted patient.

  Distinguishing itself from prior approaches, this system incorporates a wrist orientation estimator grounded in surface EMG signals derived from forearm muscles. This breakthrough significantly enhances the system's ability to faithfully track and mirror the intended arm movements.

  Empirical evaluations involving a 7 DoF exoskeleton robot provide solid evidence of the real-time effectiveness and unwavering resilience of the proposed approach. This substantiates the practical viability of the system within the rehabilitation domain.

  While initially developed for the ETS-MARSE robot, the system's adaptability allows it to seamlessly integrate with any redundant robot capable of replicating elbow, forearm, and

wrist, position and orientations. This adaptability broadens the system's potential application, making it more versatile.

The proposed controller has consistently demonstrated commendable performance in both simulation and real-time implementation, even when faced with changing trajectories. This consistent excellence underscores the system's suitability and application in rehabilitation.

It is recommended to consider expanding the system's functionality beyond guiding arm movements to include correction and resistance exercises. This diversification will enhance the system's versatility and its value across a broader range of rehabilitation needs.

Continue to advance the wrist orientation estimator by exploring advanced learning techniques to improve its precision and reliability. This will result in more accurate movement classifications based on EMG signals.

As outlined in the future research plan, the creation of a teleoperation system using Inertial Measurement Unit (IMU) sensors holds substantial promise. This initiative can act as a crucial link between patients and rehabilitation experts, facilitating remote assistance and monitoring. Pay special attention to ensuring that the system is user-friendly and accessible to both patients and rehabilitation experts. Simple interfaces and clear instructions are imperative to maximize adoption.

In conclusion, the presented mirror rehabilitation system offers a promising solution for individuals with hemiplegia following a stroke. Its potential to adapt to various rehabilitation exercises and its consistent ability to track movements establish it as a valuable asset in the field of rehabilitation. The recommendations provided here are intended to further enhance the system's functionality, reach, and effectiveness in clinical practice.

BIBLIOGRAPHY

Alibeigi, M., Rabiee, S. & Ahmadabadi, M. N. (2017). Inverse kinematics based human mimicking system using skeletal tracking technology. *Journal of Intelligent & Robotic Systems*, 85(1), 27–45.

Almasri, B. & Ouezdou, F. B. (2008). Human-like motion based on a geometrical inverse kinematics and energetic optimization. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, (1), 640–646.

Anam, K. & Al-Jumaily, A. A. (2012). Active exoskeleton control systems: State of the art. *Procedia Engineering*, 41(Iris), 988–994. Retrieved from: http://dx.doi.org/10.1016/j.proeng.2012.07.273.

Aswani, A., Gonzalez, H., Sastry, S. S. & Tomlin, C. (2013). Provably safe and robust learning-based model predictive control. *Automatica*, 49(5), 1216–1226.

Bai, S., Virk, G. & Sugar, T. (2018). *Wearable Exoskeleton Systems: Design, Control and Applications*. London UK: Institution of Engineering and Technology.

Bao, X., Sheng, Z., Dicianno, B. E. & Sharma, N. (2020). A Tube-Based Model Predictive Control Method to Regulate a Knee Joint With Functional Electrical Stimulation and Electric Motor Assist. *IEEE Transactions on Control Systems Technology*, 1–12.

Barrios-Muriel, J., Romero-Sánchez, F., Alonso-Sánchez, F. J. & Salgado, D. R. (2020). Advances in orthotic and prosthetic manufacturing: A technology review. *Materials*, 13(2), 295.

Basin, M. (2019). Finite-and fixed-time convergent algorithms: Design and convergence time estimation. *Annual reviews in control*, 48, 209–221.

Basin, M., Rodriguez-Ramirez, P. & Garza-Alonso, A. (2019). Continuous fixed-time convergent super-twisting algorithm in case of unknown state and disturbance initial conditions. *Asian Journal of Control*, 21(1), 323–338.

Bedolla-Martinez, D., Kali, Y., Saad, M., Ochoa-Luna, C. & Rahman, M. H. (2023). Learning human inverse kinematics solutions for redundant robotic upper-limb rehabilitation. *Engineering Applications of Artificial Intelligence*, 126, 106966. Retrieved from: https://www.sciencedirect.com/science/article/pii/S0952197623011508.

Bellman, R. (1966). Dynamic programming. *Science*, 153(3731), 34–37.

Benati, M., Gaglio, S., Morasso, P., Tagliassco, V. & Zaccaria, R. (1980). Anthropomorphic Robotics: Representing Mechanical Complexity. *Biological Cybernetics*, 38, 125–140.

Bhatt, P. M., Malhan, R. K., Shembekar, A. V., Yoon, Y. J. & Gupta, S. K. (2020). Expanding capabilities of additive manufacturing through use of robotics technologies: A survey. *Additive Manufacturing*, 31, 100933.

Biagiotti, L. & Melchiorri, C. (2008). *Trajectory planning for automatic machines and robots*. Springer Science & Business Media.

Billard, A. G., Calinon, S. & Guenter, F. (2006). Discriminative and adaptive imitation in uni-manual and bi-manual tasks. *Robotics and Autonomous Systems*, 54, 370–384.

Bin, Z., Rong, X. & Jun, W. (2011). Kinematics analysis of a novel 7-DOF humanoid manipulator for table tennis. *in Proc. Int. Conf. Electron. Commun. Control (ICECC)*, pp. 1524–1528.

Bitzer, S. & Van Der Smagt, P. (2006). Learning EMG control of a robotic hand: Towards Active Prostheses. *Proceedings - IEEE International Conference on Robotics and Automation*, 2006(May), 2819–2823.

Blana, D., Kyriacou, T., Lambrecht, J. M. & Chadwick, E. K. (2016). Feasibility of using combined EMG and kinematic signals for prosthesis control: A simulation study using a virtual reality environment. *Journal of Electromyography and Kinesiology*, 29, 21–27.

Brackbill, E. A., Mao, Y., Agrawal, S. K., Annapragada, M. & Dubey, V. N. (2009). Dynamics and control of a 4-dof wearable cable-driven upper arm exoskeleton. *2009 IEEE International Conference on Robotics and Automation*, pp. 2300–2305.

Brahmi, B. (2019). *Nonlinear Control of an Exoskeleton Seven Degrees of Freedom Robot to Realize an Active and Passive Rehabilitation Tasks*. Montreal: ÉCOLE DE TECHNOLOGIE SUPÉRIEURE.

Brahmi, B., Rahman, M. H., Saad, M. & Luna, C. O. (2016). Iterative estimator-based nonlinear backstepping control of a robotic exoskeleton. *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, 10(8), 1375–1381.

Brahmi, B., Saad, M., Ochoa-Luna, C. & Rahman, M. (2017a). Adaptive control of an exoskeleton robot with uncertainties on kinematics and dynamics. *2017 International conference on rehabilitation robotics (ICORR*, pp. 1369–1374.

Brahmi, B., Driscoll, M., Laraki, M. H. & Brahmi, A. (2020). Adaptive high-order sliding mode control based on quasi-time delay estimation for uncertain robot manipulator. *Control Theory and Technology*, 18(3), 279–292.

Brahmi, B., Saad, M., Rahman, M. H. & Ochoa-Luna, C. (2017b). Cartesian trajectory tracking of a 7-DOF exoskeleton robot based on human inverse kinematics. *IEEE Transaction on Systems, Man, and Cybernetics: Systems*, 49(3), 600–611.

Brahmi, B., Saad, M., Ochoa-Luna, C., Di Gennero, S. & Rahman, M. H. (2018). A new integral second-order terminal sliding mode control with time delay estimation for an exoskeleton robot with dynamics uncertainties. *Proceedings of the 5th International Conference of Control, Dynamic Systems, and Robotics (CDSR'18)*, 113, 1–10.

Bu, X. (2023). Prescribed performance control approaches, applications and challenges: A comprehensive survey. *Asian Journal of Control*, 25(1), 241–261.

Campos, F. & Calado, J. (2009). Approaches to human arm movement control—A review. *Annual reviews in control*, 33(1), 69–77.

Cavallaro, E. & Rosen, J. (2006). Real-time myoprocessors for a neural controlled powered exoskeleton arm. *Ieee Trans. Bio.-Med. Eng.*, 53(11), 2387–2396.

Chan, F. H., Yang, Y. S., Lam, F. K., Zhang, Y. T. & Parker, P. A. (2000). Fuzzy EMG classification for prosthesis control. *IEEE Transactions on Rehabilitation Engineering*, 8(3), 305–311.

Chen, C., Li, L., Peng, H., Yang, Y., Mi, L. & Zhao, H. (2020). A new fixed-time stability theorem and its application to the fixed-time synchronization of neural networks. *Neural networks*, 123, 412–419.

Chen, H. & Allgöwer, F. (1998). A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10), 1205–1217.

Craig, J. (2005). *Introduction to robotics*. Mechanics and Control.

Dai, L., Yu, Y., Zhai, D.-H., Huang, T. & Xia, Y. (2020). Robust model predictive tracking control for robot manipulators with disturbances. *IEEE Transactions on industrial electronics*, 68(5), 4288–4297.

Dalla Gasperina, S., Ghonasgi, K., de Oliveira, A. C., Gandolla, M., Pedrocchi, A. & Deshpande, A. (2020). A Novel Inverse Kinematics Method for Upper-Limb Exoskeleton under Joint Coordination Constraints. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3404–3409.

Date, N., Yoshimoto, H., Arita, D. & Taniguchi, R. I. (2004). Real-time human motion sensing based on vision-based inverse kinematics for interactive applications. *In Proceedings of the 17th International Conference on Pattern Recognition*, pp. 318–321.

Duka, A.-V. (2014). Neural Network based Inverse Kinematics Solution for Trajectory Tracking of a Robotic Arm. *Procedia Technology*, 12, 20-27. The 7th International Conference Interdisciplinarity in Engineering, INTER-ENG 2013, 10-11 October 2013, Petru Maior University of Tirgu Mures, Romania.

Ekambaram, V., Seshadri, P. V., Bache, V. K. & Sadacharam, S. (2020). Efficient discovery of survivors in disaster area using robotics and mobile networks. U.S.

Fallaha, C., Saad, M., Ghommam, J. & Kali, Y. (2020). Sliding mode control with model-based switching functions applied on a 7-dof exoskeleton arm. *IEEE/ASME Transactions on Mechatronics*, 26(1), 539–550.

Fallaha, C. J., Saad, M., Kanaan, H. Y. & Al-Haddad, K. (2011). Sliding-mode robot control with exponential reaching law. *IEEE Transactions on Industrial Electronics*, 58(2), 600–610.

Fei, F., Wang, H. & Tian, Y. (2017). Robust Time Delay Estimation Based Intelligent PID Control of a 6DOF Upper-Limb Exoskeleton Robot. *2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, 2, 386–389.

Ficuciello, F., Romano, A., Lippiello, V., Villani, L. & Siciliano, B. (2014). Human motion mapping to a robot arm with redundancy resolution. In *In Advances in Robot Kinematics* (pp. 193–201). Springer, Cham.

Fridman, L., Poznyak, A., Bejarano, F. J. et al. (2014). *Robust output LQ optimal control via integral sliding modes*. Springer.

G. Broeks, J., Lankhorst, G., Rumping, K. & Prevo, A. (1999). The long-term outcome of arm function after stroke: results of a follow-up study. *Disability and rehabilitation*, 21(8), 357–364.

Gams, A. & Lenarcic, J. (2006). Humanoid arm kinematic modeling and trajectory generation. *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006.*, pp. 301–305.

Gao, Z., Wanyama, T., Singh, I., Gadhrri, A. & Schmidt, R. (2020). From industry 4.0 to robotics 4.0-a conceptual framework for collaborative and intelligent robotic systems. *Procedia manufacturing*, 46, 591–599.

Goldstein, H. (1980). Classical Mechanics Addison-Wesley Series in Physics. Addison-Wesley USA. *second editon*.

Grüne, L., Pannek, J., Grüne, L. & Pannek, J. (2017). *Nonlinear model predictive control*. Springer.

Gunasekara, J. M., Gopura, R. A., Jayawardane, T. S. & Lalitharathne, S. W. (2012). Control methodologies for upper limb exoskeleton robots. *2012 IEEE/SICE International Symposium on System Integration, SII 2012*, 19–24.

Gunasekara, M. P., Gopura, R. A. R. C., Jayawardena, T. S. S. & Mann, G. K. I. (2013). Dexterity measure of upper limb exoskeleton robot with improved redundancy. *2013 IEEE 8th International Conference on Industrial and Information Systems*, pp. 548-553.

Hajian, G., Etemad, A. & Morin, E. (2021). Generalized EMG-based isometric contact force estimation using a deep learning approach. *Biomedical Signal Processing and Control*, 70, 103012.

Han, S., Wang, H., Tian, Y. & Christov, N. (2020). Time-delay estimation based computed torque control with robust adaptive RBF neural network compensator for a rehabilitation exoskeleton. In *ISA transactions* (pp. 171–181).

Hedjar, R., Toumi, R., Boucher, P. & Dumur, D. (2005). Finite horizon nonlinear predictive control by the Taylor approximation: application to robot tracking trajectory. *International Journal of Applied Mathematics and Computer Science*.

Hewing, L., Wabersich, K. P., Menner, M. & Zeilinger, M. N. (2020a). Learning-based model predictive control: toward safe learning in control. *Annual Review of Control*, 3, 269–296.

Hewing, L., Kabzan, J. & Zeilinger, M. N. (2020b). Cautious Model Predictive Control Using Gaussian Process Regression. *IEEE Transactions on Control Systems Technology*, 28(6), 2736–2743.

Huang, B., Ye, Z., Li, Z., Yuan, W. & Yang, C. (2018). Admittance control of a robotic exoskeleton for physical human robot interaction. *2017 2nd International Conference on Advanced Robotics and Mechatronics, ICARM 2017*, 2018-Janua(12), 245–250.

Incremona, G. P., Ferrara, A. & Magni, L. (2017). MPC for robot manipulators with integral sliding modes generation. *IEEE/ASME Transactions on Mechatronics*, 22(3), 1299–1307.

Islam, M. R., Brahmi, B., Ahmed, T., Assad-Uz-Zaman, M. & Rahman, M. H. (2020). Exoskeletons in upper limb rehabilitation: A review to find key challenges to improve functionality. In *Control Theory in Biomedical Engineering* (pp. 235–265). Academic Press.

Jadbabaie, A., Yu, J. & Hauser, J. (2001). Unconstrained receding-horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 46(5), 776–783.

Jafari Fesharaki, S., Sheikholeslam, F., Kamali, M. & Talebi, A. (2020). Tractable robust model predictive control with adaptive sliding mode for uncertain nonlinear systems. *International Journal of Systems Science*, 51(12), 2204–2216.

Jebri, A., Madani, T., Djouani, K. & Benallegue, A. (2020). Robust adaptive neuronal controller for exoskeletons with sliding-mode. *Neurocomputing*, 399, 317–330.

Jesus, R. C., Molina, L., Carvalho, E. A. & Freire, E. O. (2022). Singularity-Free Inverse Kinematics with Joint Prioritization for Manipulators. *Journal of Control, Automation and Electrical Systems*, 33(3), 1022–1031.

Johansson, B. B. (2000). Brain plasticity and stroke rehabilitation: the Willis lecture. *Stroke*, 31(1), 223–230.

Johnson, C. O., Nguyen, M., Roth, G. A., Nichols, E., Alam, T., Abate, D., Abd-Allah, F., Abdelalim, A., Abraha, H. N., Abu-Rmeileh, N. M. et al. (2019). Global, regional, and national burden of stroke, 1990–2016: a systematic analysis for the Global Burden of Disease Study 2016. *The Lancet Neurology*, 18(5), 439–458.

Jung, E., Choe, J. & Kim, S. (1994). Psychophysical cost function of joint movement for arm reach posture prediction. *Proceedings of the 38th Annual Meeting of the Human Factors and Ergonomics Society*, pp. 636–640.

Jung, E., Kee, D. & Chung, M. (1995). Upper body reach posture prediction for ergonomic evaluation models. *Int. J. Ind. Ergon.*, 16(2), 95–107.

Kabzan, J., Hewing, L., Liniger, A. & Zeilinger, M. N. (2019). Learning-Based Model Predictive Control for Autonomous Racing. *IEEE Robotics and Automation Letters*, 4(4), 3363–3370.

Kali, Y., Saad, M., Benjelloun, K. & Khairallah, C. (2018). Super-twisting algorithm with time delay estimation for uncertain robot manipulators. *Nonlinear Dynamics*, 93, 557–569.

Kelly, R., Davila, V. S. & Perez, J. A. L. (2005). *Control of Robot Manipulators in Joint Space (Advanced Textbooks in Control and Signal Processing)*. London UK: Springer-Verlag.

Khairuddin, I. M., Sidek, S. N. I., Majeed, A. P. & Puzi, A. A. (2019). Classifying Motion Intention from EMG signal: A k-NN Approach. *2019 7th International Conference on Mechatronics Engineering, ICOM 2019*, 2019–2022.

Khushaba, R. N., Al-Timemy, A., Kodagoda, S. & Nazarpour, K. (2016). Combined influence of forearm orientation and muscular contraction on EMG pattern recognition. *Expert Systems with Applications*, 61, 154–161.

Kim, H. & Rosen, J. (2015). Predicting redundancy of a 7 dof upper limb exoskeleton toward improved transparency between human and robot. *Journal of Intelligent & Robotic Systems*, 80(1), 99–119.

Kim, H., Miller, L. M., Al-Refai, A., Brand, M. & Rosen, J. (2011). Redundancy resolution of a human arm for controlling a seven DOF wearable robotic system. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 3471–3474.

Kim, H., Miller, L. M., Byl, N., Abrams, G. M. & Rosen, J. (2012a). Redundancy resolution of the human arm and an upper limb exoskeleton. *IEEE Transactions on Biomedical Engineering*, 59(6), 1770–1779.

Kim, H., Miller, L. M., Li, Z., Roldan, J. R. & Rosen, J. (2012b). Admittance control of an upper limb exoskeleton - Reduction of energy exchange. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 6467–6470.

Kim, J. & Kim, J. (2017). Robot-assisted mirroring exercise as a physical therapy for hemiparesis rehabilitation. *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 4243–4246.

Klein, C. A. & Huang, C.-H. (1983). Review of pseudoinverse control for use with kinematically redundant manipulators. *IEEE Trans. Syst., Man, Cybern*, 13(2), 245–250.

Kuipers, J. B. (1999). *Quaternions and rotation sequences: a primer with applications to orbits, aerospace, and virtual reality*. Princeton university press.

Langson, W., Chryssochoos, I., Raković, S. V. & Mayne, D. Q. (2004). Robust model predictive control using tubes. *Automatica*, 40(1), 125–133.

Lauretti, C., Cordella, F., Ciancio, A. L., Trigili, E., Catalan, J. M., Badesa, F. J., Crea, S., Pagliara, S. M., Sterzi, S., Vitiello, N. et al. (2018). Learning by demonstration for motion planning of upper-limb exoskeletons. *Frontiers in neurorobotics*, 12, 5.

Lee, H. D., Lee, B. K., Kim, W. S., Han, J. S., Shin, K. S. & Han, C. S. (2014). Human–robot cooperation control based on a dynamic model of an upper limb exoskeleton for human power amplification. *Mechatronics*, 24(2), 168–176.

Lei, Z. (2019). An upper limb movement estimation from electromyography by using BP neural network. *Biomedical Signal Processing and Control*, 49, 434–439. Retrieved from: https://doi.org/10.1016/j.bspc.2018.12.020.

Lenzi, T., De Rossi, S. M. M., Vitiello, N. & Carrozza, M. C. (2012). Intention-based EMG control for powered exoskeletons. *IEEE Transactions on Biomedical Engineering*, 59(8), 2180–2190.

Levant, A. (2007). Principles of 2-sliding mode design. *Automatica*, 43(4), 576–586.

Lima, A. A. M., Araujo, R. M., dos Santos, F. A. G., Yoshizumi, V. H., de Barros, F. K., Spatti, D. H., Liboni, L. H. & Dajer, M. E. (2018). Classification of hand movements from emg signals using optimized mlp. *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7.

Limon, D., Alamo, T., Raimondo, D. M., De La Peña, D. M., Bravo, J. M., Ferramosca, A. & Camacho, E. F. (2009). Input-to-state stability: a unifying framework for robust model predictive control. In *Nonlinear model predictive control* (pp. 1–26). Berlin, Heidelberg.: Springer.

Limón, D., Alamo, T., Salas, F. & Camacho, E. F. (2006). On the stability of constrained MPC without terminal constraint. *IEEE transactions on automatic control*, 51(5), 832–836.

Liu, J. & Zhou, P. (2012). A novel myoelectric pattern recognition strategy for hand function restoration after incomplete cervical spinal cord injury. *IEEE transactions on neural systems and rehabilitation engineering*, 21(1), 96–103.

Long, Y., jiang Du, Z., dong Wang, W. & Dong, W. (2018). Human motion intent learning based motion assistance control for a wearable exoskeleton. *Robotics and Computer-Integrated Manufacturing*, 49(August 2017), 317–327.

Maiworm, M., Limon, D., Manzano, J. M. & Findeisen, R. (2018). Stability of gaussian process learning based output feedback model predictive control. *IFAC-PapersOnLine*, 51(20), 455–461.

Marquez, H. J. (2003). *Nonlinear control systems: analysis and design*. Hoboken, John Wiley.

MathWorks. Simscape Multibody Environment. Retrieved from: https://www.mathworks.com/products/simscape-multibody.html,Accessed:2022-11-10.

Matsubara, T. & Morimoto, J. (2013). Bilinear modeling of EMG signals to extract user-independent features for multiuser myoelectric interface. *IEEE Transactions on Biomedical Engineering*, 60(8), 2205–2213.

Mayne, D. Q., Rawlings, J. B., Rao, C. V. & Scokaert, P. O. M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814.

Mayne, D. Q., Seron, M. M. & Raković, S. (2005). Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2), 219–224.

Mayne, D. Q., Kerrigan, E. C., Van Wyk, E. & Falugi, P. (2011). Tube-based robust nonlinear model predictive control. *International Journal of Robust and Nonlinear Control*, 21(11), 1341–1353.

Meng, M., She, Q., Gao, Y. & Luo, Z. (2010). EMG signals based gait phases recognition using hidden Markov models. *In The 2010 IEEE International Conference on Information and Automation*, pp. 852–856.

Mesbah, A. (2016). Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems*, 36(6), 30–44.

Mesbah, A. (2018). Stochastic model predictive control with active uncertainty learning: A survey on dual control. *Annual Reviews in Control*, 45, 107–117.

Microsoft. (2015). Kinect for Windows features. Retrieved on 2022-04-01 from: http://www.microsoft.com/en-us/kinectforwindows/meetkinect/features.aspx.

Momen, K., Krishnan, S. & Chau, T. (2007). Real-time classification of forearm electromyographic signals corresponding to user-selected intentional movements for multifunction prosthesis control. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 15(4), 535–542.

Morand, A. (2014). *Pratique de la rééducation neurologique*. Elsevier Masson.

Nam, H. S., Koh, S., Beom, J., Kim, Y. J., Park, J. W., Koh, E.-s., Chung, S. G. & Kim, S. (2017). Recovery of proprioception in the upper extremity by robotic mirror therapy: A clinical pilot study for proof of concept. *Journal of Korean medical science*, 32(10), 1568–1575.

Nguyen-Tuong, D. & Peters, J. (2011). Model learning for robot control: a survey. *Cognitive processing*, 12, 319–340.

Ochoa-Luna, C. (2016). *Nonlinear Control of a Seven Degrees-of-Freedom Exoskeleton Robot Arm*. Montreal: ÉCOLE DE TECHNOLOGIE SUPÉRIEURE.

Ochoa Luna, C., Habibur Rahman, M., Saad, M., Archambault, P. S. & Bruce Ferrer, S. (2015). Admittance-based upper limb robotic active and active-assistive movements. *International Journal of Advanced Robotic Systems*, 12(9), 117.

Ong, S., Yew, A., Thanigaivel, N. & Nee, A. (2020). Augmented reality-assisted robot programming system for industrial applications. *Robotics and Computer-Integrated Manufacturing*, 61, 101820. Retrieved from: https://www.sciencedirect.com/science/article/pii/S0736584519300250.

Ostafew, C. J. (2010). *Learning-based Control for Autonomous Mobile Robots*. (Ph.D. thesis).

Ostafew, C. J., Schoellig, A. P. & Barfoot, T. D. (2016a). Robust Constrained Learning-based NMPC enabling reliable mobile robot path tracking. *International Journal of Robotics Research*, 35(13), 1547–1563.

Ostafew, C. J., Schoellig, A. P., Barfoot, T. D. & Collier, J. (2016b). Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking. *Journal of Field Robotics*, 33(1), 133–152.

Oztemel, E. & Gursev, S. (2020). Literature review of Industry 4.0 and related technologies. *Journal of Intelligent Manufacturing*, 31(1), 127–182.

Parajuli, N., Sreenivasan, N., Bifulco, P., Cesarelli, M., Savino, S., Niola, V., Esposito, D., Hamilton, T. J., Naik, G. R., Gunawardana, U. et al. (2019). Real-time EMG based pattern recognition control for hand prostheses: a review on existing methods, challenges and future implementation. *Sensors*, 19(20), 4596.

Park, B. G., Lee, J. W. & Kwon, W. H. (1999). Robust one-step receding horizon control for constrained systems. *International Journal of Robust and Nonlinear Control*, 9(7), 381–395.

Park, J.-Y., Chang, M., Kim, K.-M. & Kim, H.-J. (2015). The effect of mirror therapy on upper-extremity function and activities of daily living in stroke patients. *Journal of physical therapy science*, 27(6), 1681–1683.

Pereida, K. & Schoellig, A. P. (2018). Adaptive model predictive control for high-accuracy trajectory tracking in changing conditions. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS*, pp. 7831–7837.

Perrusquía, A., Yu, W. & Li, X. (2020). Multi-agent reinforcement learning for redundant robot control in task-space. *International Journal of Machine Learning and Cybernetics*, 1–11.

Phinyomark, A., Phukpattaranont, P. & Limsakul, C. (2012). Feature reduction and selection for EMG signal classification. *Expert systems with applications*, 39(8), 7420–7431.

Polyakov, A. & Poznyak, A. (2009a). Lyapunov function design for finite-time convergence analysis:"Twisting" controller for second-order sliding mode realization. *Automatica*, 45(2), 444–448.

Polyakov, A. & Poznyak, A. (2009b). Reaching time estimation for "super-twisting" second order sliding mode controller via Lyapunov function designing. *IEEE Transactions on Automatic Control*, 54(8), 1951–1955.

Poppe, R. (2007). Vision-based human motion analysis: An overview. *Computer vision and image understanding*, 108(1-2), 4–18.

Qin, H. & Sun, Y. (2020). Chapter 5 - Autonomous control of underwater offshore vehicles. In Karimi, H. R. (Ed.), *Fundamental Design and Automation Technologies in Offshore Robotics* (pp. 115-160). Academic Press.

Rahman, M. H. (2012). *Development of an exoskeleton robot for upper-limb rehabilitation*. Montréal: École de Technologie Supérieure.

Rahman, M. H., Rahman, M. J., Cristobal, O., Saad, M., Kenné, J.-P. & Archambault, P. S. (2015). Development of a whole arm wearable robotic exoskeleton for rehabilitation and to assist upper limb movements. *Robotica*, 33(1), 19–39.

Rahmani, M. & Rahman, M. H. (2019). An upper-limb exoskeleton robot control using a novel fast fuzzy sliding mode control. *Journal of Intelligent & Fuzzy Systems*, 36(3), 2581–2592.

Rasmussen, C. E. & Nickisch, H. (2010). Gaussian processes for machine learning (GPML) toolbox. *Journal of Machine Learning Research*, 11, 3011-3015. Retrieved from: www.scopus.com. Cited By :681.

Rawlings, J. B., Mayne, D. Q. & Diehl, M. (2017). *Model predictive control: theory, computation, and design* (ed. 2). Madison, WI: Nob Hill Publishing.

Ren, J.-L., Chien, Y.-H., Chia, E.-Y., Fu, L.-C. & Lai, J.-S. (2019). Deep learning based motion prediction for exoskeleton robot control in upper limb rehabilitation. *2019 International Conference on Robotics and Automation (ICRA)*, pp. 5076–5082.

Rubagotti, M., Raimondo, D. M., Ferrara, A. & Magni, L. (2009). Robust model predictive control of continuous-time sampled-data nonlinear systems with integral sliding mode. *2009 European Control Conference (ECC)*, pp. 2247–2252.

Rubagotti, M., Raimondo, D. M., Ferrara, A. & Magni, L. (2010). Robust model predictive control with integral sliding mode in continuous-time sampled-data nonlinear systems. *IEEE Transactions on Automatic Control*, 56(3), 556–570.

Scokaert, P. O. M. & Mayne, D. Q. (1998). Min Max feedback model predictive control for Constrained Linear Systems. *IEEE Transaction on automatic control*, 43(8), 1136–1142.

Shi, W.-T., Lyu, Z.-J., Tang, S.-T., Chia, T.-L. & Yang, C.-Y. (2018). A bionic hand controlled by hand gesture recognition based on surface EMG signals: A preliminary study. *Biocybernetics and Biomedical Engineering*, 38(1), 126–135.

Shtessel, Y., Edwards, C., Fridman, L., Levant, A. et al. (2014). *Sliding mode control and observation*. Springer.

Slotine, J. J. E. & Li, W. (1991). *Applied nonlinear control*. Englewood Cliffs: Prentice hall.

Soltani-Zarrin, R., Zeiaee, A., Langari, R. & Tafreshi, R. (2017). A computational approach for human-like motion generation in upper limb exoskeletons supporting scapulohumeral rhythms. *arXiv*, (1), 1–2.

Spasic, M., Hovd, M., Mitic, D. & Antic, D. (2016). Tube Model Predictive Control with an Auxiliary Sliding Mode Controller. *Modeling, Identification and Control*, 37(3), 181–193.

Stinear, C. M., Lang, C. E., Zeiler, S. & Byblow, W. D. (2020). Advances and challenges in stroke rehabilitation. *The Lancet Neurology*, 19(4), 348–360. Retrieved from: http://dx.doi.org/10.1016/S1474-4422(19)30415-6.

Su, H., Enayati, N., Vantadori, L., Spinoglio, A., Ferrigno, G. & Momi, E. D. (2018). Online human-like redundancy optimization for tele-operated anthropomorphic manipulators. *International Journal of Advanced Robotic Systems*, 15(6), 1-13.

Su, H., Qi, W., Gao, H., Hu, Y., Shi, Y., Ferrigno, G. & De Momi, E. (2020). Machine learning driven human skill transferring for control of anthropomorphic manipulators. *2020 5th International Conference on Advanced Robotics and Mechatronics (ICARM)*, pp. 107–112.

Tang, W., Cavazza, M., Mountain, D. & Earnshaw, R. (1998). Real-time inverse kinematics through constrained dynamics. In *Modelling and motion capture techniques for virtual enviroments* (pp. 159–170). Springer.

Teasell, R., Meyer, M. J., Foley, N., Salter, K. & Willems, D. (2009). Stroke rehabilitation in Canada: a work in progress. *Topics in stroke rehabilitation*, 1(16), 11–19.

Teasell, R. W., Foley, N. C., Salter, K. L. & Jutai, J. W. (2008). A Blueprint for Transforming Stroke Rehabilitation Care in Canada: The Case for Change. *Archives of Physical Medicine and Rehabilitation*, 89(3), 575-578.

Terry, J. S., Rupert, L. & Killpack, M. D. (2017). Comparison of linearized dynamic robot manipulator models for model predictive control. *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pp. 205–212.

Thies, M., Zäch, J.-N., Gao, C., Taylor, R., Navab, N., Maier, A. & Unberath, M. (2020). A learning-based method for online adjustment of C-arm Cone-beam CT source trajectories for artifact avoidance. *International journal of computer assisted radiology and surgery*, 15, 1787–1796.

Tolani, D., Goswami, A. & Badler, N. I. (2000). Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical Models*, 62(5), 353–388.

Tolani, D. & Badler, N. I. (1996). Real-time inverse kinematics of the human arm. *Presence: Teleoperators and Virtual Environments*, 5(4), 393–401.

Torossian, B., Bekkers, F., Sweijs, T., Roelen, M., Hristov, A. & Atalla, S. (2020). *The Military Applicability of Robotic and Autonomous Systems*. Hague Centre for Strategic Studies.

Treussart, B., Geffard, F., Vignais, N. & Marin, F. (2020). Controlling an upper-limb exoskeleton by EMG signal while carrying unknown load. *Proceedings - IEEE International Conference on Robotics and Automation*, 9107–9113.

Triwiyanto, T., Pawana, I. P. A., Irianto, B. G., Indrato, T. B. & Wisana, I. D. G. H. (2019). Embedded system for upper-limb exoskeleton based on electromyography control. *Telkomnika (Telecommunication Computing Electronics and Control)*, 17(6), 2992–3002.

Venkataraman, S. T. & Gulati, S. (1993). Control of Nonlinear Systems Using Terminal Sliding Modes. *Journal of Dynamic Systems, Measurement, and Control*, 115(3), 554-560.

Wang, C., Peng, L., Hou, Z.-G., Li, J., Luo, L., Chen, S. & Wang, W. (2019). Kinematic redundancy analysis during goal-directed motion for trajectory planning of an upper-limb exoskeleton robot. *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 5251–5255.

Wang, J., Wang, J. & Han, Q.-L. (2021). Neurodynamics-Based Model Predictive Control of Continuous-Time Under-Actuated Mechatronic Systems. *IEEE/ASME Transactions on Mechatronics*, 26(1), 311-322.

Wang, W.-W. & Fu, L.-C. (2011). Mirror therapy with an exoskeleton upper-limb robot based on IMU measurement system. *2011 IEEE International Symposium on Medical Measurements and Applications*, pp. 370–375.

Williams, C. K. & Rasmussen, C. E. (2006). *Gaussian processes for machine learning*. MIT press Cambridge, MA.

Xie, J., Zhao, X. & Dong, H. (2021). Learning-based nonlinear model predictive control with accurate uncertainty compensation. *Nonlinear Dynamics*, 104(4), 3827–3843.

Xie, S. (2016). Advanced robotics for medical rehabilitation. *Springer tracts in advanced robotics*, 108(1), 357.

Xu, B., Zhang, R., Li, S., He, W. & Shi, Z. (2020). Composite Neural Learning-Based Nonsingular Terminal Sliding Mode Control of MEMS Gyroscopes. *IEEE Transactions on Neural Networks and Learning Systems*, 31(4), 1375–1386.

Xu, J. & Jian-Xin, J. (2013). Iterative learning control for output-constrained systems with both parametric and nonparametric uncertainties. *Automatica*, 49(8), 2508–2516.

Xu, J., Wang, W. & Sun, Y. (2010). Two optimization algorithms for solving robotics inverse kinematics with redundancy. *Journal of Control Theory and Applications*, 8(2), 166–175.

Yang, D. & Liu, H. (2021). An EMG-based deep learning approach for multi-DOF wrist movement decoding. *IEEE Transactions on Industrial Electronics*, 69(7), 7099–7108.

Yang, J., Marler, R. T., Kim, H., Arora, J. & Abdel-Malek, K. (2004). Multi-objective optimization for upper body posture prediction. *10th AIAA/ISSMO multidisciplinary analysis and optimization conference*, pp. 4506.

Yang, W., Xu, D., Jiang, B. & Shi, P. (2021a). A novel dual-mode robust model predictive control approach via alternating optimizations. *Automatica*, 133, 109857.

Yang, Z., Guo, S., Hirata, H. & Kawanishi, M. (2021b). A mirror bilateral neuro-rehabilitation robot system with the sEMG-based real-time patient active participant assessment. *Life*, 11(12), 1290.

Youcef-Toumi, K. & Ito, O. (1990). A time delay controller for systems with unknown dynamics. *ASME. J. Dyn. Sys., Meas., Control*, 112(1), 133–142.

Yu, W. & Rosen, J. (2013). Neural PID control of robot manipulators with application to an upper limb exoskeleton. *IEEE Transactions on cybernetics*, 43(2), 673–684.

Yu, Y., Dai, L., Sun, Z. & Xia, Y. (2021). Robust Model Predictive Tracking Control for Robot Manipulators With Disturbances. *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, 68(5), 4288–4297.

Zanchettin, A. M., Bascetta, L. & Rocco, P. (2013). Achieving humanlike motion: Resolving redundancy for anthropomorphic industrial manipulators. *IEEE Robotics and Automation Magazine*, 20(4), 131–138.

Zhang, C. & Yu, S. (2023). Disturbance observer-based prescribed performance super-twisting sliding mode control for autonomous surface vessels. *ISA transactions*, 135, 13–22.

Zhang, C., Chen, X., Cao, S., Zhang, X. & Chen, X. (2019a). A novel HD-sEMG preprocessing method integrating muscle activation heterogeneity analysis and kurtosis-guided filtering for high-accuracy joint force estimation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(9), 1920–1930.

Zhang, L., Wang, Y., Hou, Y. & Li, H. (2019b). Fixed-Time Sliding Mode Control for Uncertain Robot Manipulators. *IEEE Access*, 7, 149750-149763.

Zhang, S., Guo, S., Gao, B., Huang, Q., Pang, M., Hirata, H. & Ishihara, H. (2016). Muscle strength assessment system using sEMG-based force prediction method for wrist joint. *Journal of Medical and Biological Engineering*, 36(1), 121–131.

Zhang, Y. & Lu, M. (2020). A review of recent advancements in soft and flexible robots for medical applications. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 16(3), 2096.

Zhao, J., Xie, B. & Song, C. (2014). Generating human-like movements for robotic arms. *Mechanism and Machine Theory*, 81, 107–128.

Zhou, J.-S., Liu, Z.-Y. & Pei, R. (2001). A new nonlinear model predictive control scheme for discrete-time system based on sliding mode control. *Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148)*, 4, 3079–3084.