Imperial College London
Department of Mechanical Engineering

Model Predictive Control for Intelligent Lower Limb Robotic Assistance

Christopher Caulcrick

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Mechanical Engineering, August 2021

Statement of Originality

I declare that the work of this thesis is original and that all material in this thesis is my own unless duly acknowledged. The work has not been submitted to any other institution for any degree or diploma.

Copyright Notice

Abstract

Loss of mobility or balance resulting from neural trauma is a critical consideration in public health. Robotic exoskeletons hold great potential for rehabilitation and assisted movement, yet optimal assist-as-needed control remains unresolved given pathological variation among patients. This thesis develops, simulates and experimentally evaluates a Model Predictive Control (MPC) architecture for lower limb exoskeletons. It uses a Fuzzy Logic Algorithm (FLA) to bridge the gap in human-robot synergy by identifying modes of assistance (passive, active-assist, and active-resist) based on human involvement.

Muscle activity, mapped through Electromyography (EMG) or Mechanomyography (MMG), is known to precede the onset of human joint torque for modelling and prediction of human-exoskeleton movement. This thesis investigates the complementary and competing benefits of MMG and EMG as a means of human joint torque prediction. A qualitative and quantitative comparison is presented using three biomechanics agnostic machine learning approaches: linear regression, polynomial regression, and neural networks. At the expense of training and implementation complexity, the neural network models performed best, achieving a normalised mean absolute error of 0.063 with MMG and 0.048 with EMG.

The controller is evaluated in hardware with three subjects on a seated 1-Degree of Freedom (DOF) knee exoskeleton tracking a sinusoidal trajectory with human relaxed, assistive, and resistive. The controller is also demonstrated with one subject assisting the swing phase of walking. Human joint torque is predicted using a linear regression model with EMG signals to inform the MPC and assistance mode selection by the FLA. Experimental results show quick and appropriate transfers among the assistance modes and satisfactory assistive performance in each mode. Results illustrate an objective approach to lower limb robotic assistance through on-the-fly transition between modes of movement. This provides a new level of human-robot synergy for robotic rehabilitation, mobility assistance, and human-robot interaction more broadly.

Acknowledgements

The contributions of this thesis were jointly funded by the Engineering and Physical Sciences Research Council (EPSRC) and McLaren Applied through an Industrial Cooperative Award in Science & Technology (CASE). The research was conducted under supervision of Dr. Ravi Vaidyanathan, Reader in Biomechatronics at Imperial College London. Industrial supervision was provided by Dr. Caleb Sawade and Dr. Will Hoult at McLaren Applied. I would like to thank all of my supervisors for their excellent supervision. I am also grateful to Dr. Samer Mohamed for making me feel welcome on my visits to the LISSI Lab in Paris.

Throughout my master's degree and PhD I have been fortunate to work surrounded by good friends in the Biomechatronics Lab. Thank you for the laughs and memorable conversations. I am particularly grateful to Weiguang Huo for his mentorship and generosity. I would also like to thank the volunteers who participated in testing and were very patient in the process.

Finally, I would like to express my deepest gratitude to my family, my friends, and Sophie.

Contents

List of Tables

List of Figures

Abbreviations

AAN Assist-as-Needed.

ADC Analog-to-Digital Converter.

ADL Activities of Daily Living.

API Application Programming Interface.

BF Biceps Femoris.

BWS Body Weight Supported.

DOF Degree of Freedom.

EMG Electromyography.

FES Functional Electrical Stimulation.

FLA Fuzzy Logic Algorithm.

FMV Fuzzy Membership Value.

GUI Graphical User Interface.

HTE Human Torque Estimation.

IMU Inertial Measurement Unit.

IQR Interquartile Range.

MAE Mean Absolute Error.

MMG Mechanomyography.

MPC Model Predictive Control.

MSE Mean Square Error.

nMAE Normalised Mean Absolute Error.

NN Neural Network.

OCP Optimal Control Problem.

OLS Ordinary Least Squares.

PD Proportional Derivative.

PI Proportional Integral.

PID Proportional Integral Derivative.

QOL Quality of Life.

RBF Radial Basis Function.

RF Rectus Femoris.

RMS Root Mean Square.

RMSE Root Mean Square Error.

SCI Spinal Cord Injury.

SD Standard Deviation.

SE Semitendinosus.

SEA Series Elastic Actuator.

VM Vastus Medialis.

Nomenclature

$\ddot{\theta}$	Angular Acceleration
$\ddot{y}(s)$	Acceleration
$\dot{\theta}$	Angular Velocity
$\dot{\theta}_r$	Reference Angular Velocity
$\dot{y}(s)$	Velocity
$\hat{\tau}_h$	Estimated Human Torque
τ	Torque
τ_e	Exoskeleton Torque
τ_g	Gravitational Torque
τ_h	Human Torque
θ	Joint Angle
θ_d	Desired Joint Angle
θ_e	Joint Angle Error
θ_l	Lumped Torque
θ_r	Reference Joint Angle
θ_t	Hip Joint Angle
A	Static Friction
B	Viscous Friction
$B(s)$	Damping
d_s	Shank Length to Centre of Mass
$F(s)$	Force
$I(s)$	Inertia
J	Inertia
$K(s)$	Stiffness

l_t	Thigh Length
m	Assistance Mode Variable
m_s	Shank Mass
$y(s)$	Position
$Z(s)$	Impedance

Chapter 1

Introduction

This thesis is an investigation into applying Model Predictive Control (MPC) to robotic assistance while using muscle activity information from the subject to inform the controller and adapt robot behaviour. Motivating this work is the need for robotic assistance and rehabilitation devices to adjust to patient-specific requirements, which vary significantly between patients and over the course of recovery. MPC is used to meet this need because of the method's ability to optimise for various control objectives simultaneously, dictated by the configuration of a cost function. These objectives can include trajectory tracking, robot joint torques, and enforcing of constraints on system states. The MPC cost function is not limited to using internal robotic system information, and it is the inclusion of external information measured from the human subject that is the focus of this work.

1.1 Robotic Assistance and Rehabilitation

Balance dysfunction resulting from ageing and/or neural trauma is reaching epidemic proportions. In the USA, there are 800,000 new cases of stroke each year [1]; 40% of patients suffer serious falls within a year

of stroke [2]. Indeed, the number of falls per year due to stroke is expected to reach 500,000 in the USA alone by 2040, representing a total annual cost of $16 billion per year [3]. Population-based studies have shown a 35% prevalence of gait disorders among persons over age 70, and 80% over 85 years of age [4]. The UK societal cost of stroke exceeds £8 billion a year [5]. Following acute medical response, long-term patient outcomes are almost entirely dependent on rehabilitation and early initiation is critical. Outcome can be impacted by beginning activity in as little as 24 hours after stroke trauma [6] and there is broad agreement that regular activity within the first 6 months of incidence is vital for best possible recovery [7].

Early activity post-stroke is often impossible without some form of supported movement, yet limited resources mean that most patients receive inadequate rehabilitation; even in-clinic stroke patients are only functionally active 13% of the day [8]. Ageing population projections indicate a 44% increase in strokes in the UK in the next 20 years [9]. Even partial muscle dysfunction impacts simple independence and Activities of Daily Living (ADL). Age-related changes in muscle and joint forces have also been demonstrated in movements such as sit-to-stand [10] which demands up to 97% of available muscle strength in the elderly [11]. Stroke patients consequently take significantly longer to execute simple movements, entailing greater risk of falls and poorer capacity to function [12]. The importance of lower limb rehabilitation for mobility and its role in increased risk of social isolation have led to a range of rehabilitation therapies; however, exploiting the full potential of assistive technologies in clinical settings remains challenging. Telemedicine in rehabilitation is an area of vital future importance [13], [14]. Neural trauma and elderly patients are in firm need of tools to support their balance, mobility and rehabilitation out-of-clinic.

Wearable robots – i.e. robotic exoskeletons – offer the potential for supported movement and immediate therapy post trauma which is critical for neural rehabilitation as well as general mobility. Compared with conventional therapy, robotic rehabilitation may deliver highly controlled repetitive and intensive training, reduce the burden of clinical staff as well as providing a quantitative assessment of motion and forces

[15]–[17]. Stationary devices such as GaitTrainer [18] and HapticWalker [19] report both patient cognitive engagement and compliance during locomotor training. Furthermore, studies using wearable exoskeletons such as Hybrid Assistive Limb [20], Ekso [21] and H2 [22] have argued that relevant sensory inputs and central neuronal circuits become activated under physiological conditions (i.e. overground walking) leading to neural regeneration. Significant effort has been focused on synergising exoskeleton action for patients with the capacity to provide partial muscular efforts (e.g. elderly or stroke patients), for whom the exoskeleton complements the wearer's strength to complete a successful movement. This demands the combination of augmenting mobility, lowering risk of injury and, potentially, supporting rehabilitation. The majority of patient beneficiaries are at serious risk of falls from loss of control [23] as well as sustained damage due to unhealthy compensation [10]. However, this group also has the potential of rehabilitation if exoskeleton control can synergise its action with user muscle activity. Achieving this demands an exoskeleton to be intelligent enough to coordinate its activity with human muscle action, recognise and reward human effort to maximise rehabilitation, and be compliant such that it does not inhibit user movements.

As a serious life-threatening disease, stroke can lead to long-term problems affecting patients' daily living ability, which places heavy burdens on families, communities and health services. There are over 13.7 million new stroke cases worldwide each year since 1990 [24], and 5 million of them result in permanently disability [25]. The annual cost of strokes has been estimated at 51 billion dollars in the USA and 38 billion euros in Europe [26], [27].

Research has shown that intensive rehabilitation training and exercises can help stroke patients relearn patterns of motion, enabling them to move again [28], meanwhile exercises must be appropriately adjusted according to patients' rehabilitation stages. There are three common types of these exercises: passive, active-assist, and active, which are respectively used in the early, middle and late stages of stroke reha-

bilitation. Passive exercises completely rely on the assistance of therapists; active-assist exercises are used when patients have limited mobility but still need therapists' assistance for performing a full range of motion; active exercises can take place without help from therapists once voluntary muscle control is recovered [29]. Such subject-specific rehabilitation processes are largely labour-intensive and costly. Understandably, development of robots for assisting post-stroke patients in daily living activities and rehabilitation training is recently attracting great attention [30].

Robotic assistive devices, such as powered exoskeletons and orthoses, can assist patients with passive and active-assist exercises [31], [32], [33]. Position control has been popularly used to carry out passive exercise. For active-assist, robotic devices are required to provide appropriate assistance by sensing patient participation and/or understanding the patient's physical limitations, much like a trained therapist. Control approaches delivering active-assist have been termed Assist-as-Needed (AAN) [34], i.e., allowing patients to complete movement without suppressing effort contribution [35]. Although studies have shown AAN control can support the recovery of stroke patients, finding an optimal general AAN algorithm is challenging because the pathology can vary greatly with each case, and the rate of recovery is highly unpredictable [36].

1.2 Problem Statement

A common solution for providing active-assist exercise to an impaired limb during AAN robotic rehabilitation is to increase compliance of servo-type position controllers. This can be achieved either using mechanically compliant joints [37] or by reducing control gains [38]. The main advantage of such approaches is that they are relatively simple to implement as they do not require knowledge of the underlying system model. However, they are not adaptive to individual condition and have limited effectiveness at stimu-

lating neuroplasticity as patient effort is not actively encouraged [39]. Meanwhile, impedance control is employed for many assistive exoskeletons, aiming to exhibit human-like mechanical properties and implement a good compromise between tracking ability and compliance [40]. Conventional impedance control may require trial-by-trial adaptation to obtain suitable impedance which varies from subject to subject and with muscular fatigue. Applying such methods in clinical situations is therefore not straightforward, particularly for early stages of recovery. A number of adaptive impedance approaches have been developed to promote and maintain patient participation [41], [42]. Pérez-Ibarra et al. [43] proposed an assistive-resistive strategy, which can determine appropriate task difficulty by estimating the dynamic contribution of the patient based on an exoskeleton dynamic model. However, drawbacks of such model-based methods include significant latency in human effort estimation and sensitivity to external disturbances. A small body of work has explored offline adaptive control to periodically change task difficulty, i.e., adapting trial-by-trial to modulate control parameters based on predesigned performance indexes. For example, Balasubramanian et al. [44] designed a wearable upper-extremity robot for functional therapy in repetitive ADL. An iterative learning control was used to tune the assistive torque trial-by-trial. Such approaches also face a latency problem, not responding quickly to changes of the wearer's motion intention and rehabilitation requirements.

Existing AAN control approaches have a common problem - many do not adapt to changes in the subject's rehabilitation requirements in time to prevent periods of unsuitable interaction condition. Human-robot interaction should be suitable in terms of joint compliance and encouraged human effort contribution to the exercise. Patient ability can vary spatially, with greater mobility displayed in particular areas of joint motion. These requirements can vary greatly between patients and over the course of rehabilitation, therefore trial-by-trial parameter modulation approaches have no guarantee of adapting in time to keep up with the user's changing needs, which may vary within the time frame of a trial elapsing. Previous AAN control approaches that have relied on system models have suffered from latency when estimating

dynamic contribution of the user, and inability to separate these contributions from external disturbances.

1.3 Aim and Objectives

The aim of this work is to develop an AAN control approach for robotic assistance which can adapt to changes in the user's requirements in real-time and distinguish human effort contribution from external disturbance. The objectives of this research are to address the most prominent issues present in existing AAN control approaches: a) adaptation latency, b) singular modality, c) unconstrained behaviour, d) disturbance identification, e) human torque estimation latency, and f) flawed human torque estimation assumptions.

1.4 Scope of this Research

Limits to the research scope and application of the AAN control approach are as follows.

- The targeted conditions for robotic rehabilitation and assistance in this research are limited to stroke and loss of mobility in the elderly. Other types of pathology for which AAN robotic assistance may be relevant include Parkinson's and Spinal Cord Injury (SCI). The highly dynamic and rapidly changing human-robot interaction demands in assistance of Parkinson's patients with tremor are not taken into account. The control approach may be applicable to mobility problems found with spinal cord injury patients, but development and evaluation of this research focuses on the application and efficacy of the controller with respect to stroke patients in all stages of recovery and elderly patients suffering muscular weakness.

- This investigation is limited to actuation of a single robotic joint. Though this limits the applicability of specific controller configurations developed in this study, it allows the research to focus on user requirements and human-robot interaction at the joint level. By focusing on analysis at joint level, the investigation can progress efficiently, without conflation of findings from multiple joints made simultaneously. The control strategy can be applied individually for robotic assistance at multiple joints if necessary.

- The example exercises considered in this study are limited to movements in the sagittal plane. This reduces the degrees of freedom, and the mathematical modelling complexity of the human-exoskeleton system during exercise tasks.

- The exercises investigated in this work are limited to lower limb examples. This narrows the research scope, reducing the number of muscle groups and human-exoskeleton system model parameters involved in controller development. Although the applications in this work are limited to lower limb single joint actuation, the principles and findings from this work are applicable to upper limb robotic assistance. The background literature review therefore includes control approaches applied to upper limb and lower limb robotic assistance.

1.5 Contributions

This thesis reports a number of key developments of interest to the area of AAN control for robotic assistance. The contributions are summarised in the following points.

- An AAN-MPC control architecture is presented which can adapt to changes in the wearer's requirements using information from muscle activity sensor inputs. The tools and methodology for designing AAN control algorithms can apply to general robotic assistance cases involving human-robot in-

teraction. The method addresses issues prevalent within AAN control: human dynamic contribution estimation latency, and external disturbance sensitivity.

- A novel Fuzzy Logic Algorithm (FLA) for identifying the suitable robotic assistance mode based on estimated human dynamic contribution and reference exercise task.
- A novel experimental procedure for recording and processing representative human isometric and dynamic muscle contraction signals during various human-exoskeleton interactions.
- A human joint torque estimation approach for predicting human joint torque based on muscle activity signals and system states.
- A qualitative and quantitative comparison of novel microphone-based Mechanomyography (MMG) and Electromyography (EMG) for human joint torque prediction
- A simulation framework for human-robot systems using real sensor data collected from subjects, useful for designing and testing the controller, fuzzy logic algorithm, and human torque estimation approach offline.
- Validation of the control architecture with the design and construction of single joint exoskeleton robot configurations for testing the controller with subjects.
- Real-time robot control software combining controller code with a Graphical User Interface (GUI) and Application Programming Interface (API)s for sensor data acquisition and motor control. The code also has simulation functionality for testing controller and system behaviour with sensor data collected either offline or on-the-fly.

1.6 Publications

Journals Papers

C. Caulcrick, W. Huo, and R. Vaidyanathan, "Human Joint Torque Modelling with MMG and EMG during Lower Limb Human-Exoskeleton Interaction," *IEEE Robotics and Automation Letters*, 2021.

C. Caulcrick, W. Huo, E. Franco, S. Mohammed, W. Hoult, and R. Vaidyanathan, "Model Predictive Control for Human-Centred Lower Limb Robotic Assistance," *IEEE Transactions on Medical Robotics and Bionics*, 2021.

Conferences

C. Caulcrick, W. Huo, and R. Vaidyanathan, "Human Joint Torque Modelling with MMG and EMG during Lower Limb Human-Exoskeleton Interaction," *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE), Lyon, France*, Online Presentation.

C. Caulcrick, W. Huo, E. Franco, R. Vaidyanathan, "MMG-based MPC for Robotic Rehabilitation," *2019 14th Dynamic Walking Conference, Canmore, Alberta, Canada*, Podium Presentation.

C. Caulcrick, F. Russell, S. Wilson, C. Sawade and R. Vaidyanathan, "Unilateral Inertial and Muscle Activity Sensor Fusion for Gait Cycle Progress Estimation," *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob), Enschede, Netherlands*, pp. 1151-1156, Poster Presentation.

1.7 Structure

This thesis has been organised in 7 chapters in addition to this one, with a synopsis of the remaining chapters given as follows.

Chapter 2 provides a state of the art review of robotic assistance then focuses on post-stroke rehabilitation. The main application areas of strength augmentation, mobility assistance, and robotic rehabilitation are explored. Within the latter, topics of neurological disorders and neurorehabilitation requirements, including commonly used lower limb rehabilitation exercises, exercise modes, and robot control challenges are discussed. The chapter provides context for the study and understanding of current applications that are significant in the field of wearable robotics for rehabilitation and assistance.

Chapter 3 provides background on control methods for robotic rehabilitation and assistance, and details of existing state of the art AAN control approaches. Training modalities used in rehabilitation are discussed including where these modes are useful in the course of physical recovery. The most common AAN control approaches are presented and examples of each are critically reviewed. Following comparison and evaluation of the approaches, adaptation latency is identified as a common weakness, which leads to the research proposition of intelligent model predictive control using muscle activity signals for human joint torque prediction.

Chapter 4 provides the outline for the control approach proposed in this thesis. Following development of the control framework, the human-exoskeleton systems are presented, including details of the knee joint actuator, seated and standing exoskeleton configurations, and accompanying models.

The presented models are used for design and simulation of the intelligent nonlinear MPC strategy that follows. An example of the MPC approach is provided with simulation results, comparing the performance of the proposed controller to Proportional Integral Derivative (PID) and impedance control methods with respect to constrained assistance, step response, and sudden human torque response. Comparison of simulated performance for seated tracking and the influence of modelling error is also presented.

Chapter 5 presents the FLA for assistance mode detection. Background of fuzzy logic applied to robotic assistance is provided. The method of the algorithm is developed, which explains how likelihood values for active-assist, passive, safety modes of operation are determined from the inputs of reference joint velocity and estimated human joint torque. The purpose of the assistance penalty parameter is discussed and a method of trajectory adaptation using high level human interaction condition information is proposed. Finally, a comparison study exploring the effect of using the assistance mode variable to adjust gain values of PID, impedance, and intelligent MPC controllers is presented.

Chapter 6 explores the important challenge of predicting human joint torque using muscle activity signals and joint states. Background of existing approaches to estimating human muscle force and joint torque using muscle activity signals is provided. MMG and EMG muscle activity signals are both tested to compare their ability to predict human joint behaviour. Various modelling approaches are investigated, and the resulting predictions are compared and contrasted. It is concluded that EMG signals provide the most accurate human joint torque prediction, so this signal is proposed for validation of the proposed control framework.

Chapter 7 presents an evaluation of the proposed AAN control framework for seated and standing exercises. Evaluation metrics including human torque ratio and robot compliance are proposed for quantitative evaluation and comparison of controller behaviour. Several healthy subjects are involved in the study which evaluates the ability of the controller to deliver suitable assistance to subjects across various human interaction conditions. The assistance modes of passive, active-assist, and safety are demonstrated for seated leg extension and flexion, and overground walking. The proposed controller is shown to respond rapidly to changes in human-robot interaction, and adjust the nature of robotic assistance, addressing the major shortfall of the existing AAN controllers discussed in Chapter 3.

Chapter 8 summaries key findings, compares the proposed AAN control approach to existing methods, and, finally, describes possible areas for future work.

Chapter 2

Wearable Robots for Lower Limb Assistance

As discussed in the introduction, wearable robots have an important part to play in providing assistance for individuals with very specific physical needs. This chapter provides context for this study and understanding of current applications that are significant in the field of wearable robotics for assistance. The first part of this chapter describes the approach taken for the literature review. The scope of the survey is provided, limitations are stated, and key definitions are provided for terms used in the review and throughout the rest of the thesis. The second part of this chapter provides an overview of lower limb wearable robots for assistance and summarises the main application areas of strength augmentation, mobility assistance, and robotic rehabilitation. The third part of this chapter provides detailed background on the application of robotic rehabilitation, which is the area of focus for the work in this thesis. This covers neurological disorders, neurorehabilitation requirements including commonly used lower limb rehabilitation exercises and exercise modes, and robot control challenges. This will provide the necessary background for the following chapter which includes a literature survey focusing on control methods for robotic rehabilitation.

2.1 Approach to Literature Survey

The scope of the literature survey in this thesis (Chapters 2 & 3) is illustrated using the map in Fig. 2.1. In this chapter, notable wearable robots for assistance are explored and grouped according to type of end user application: strength augmentation, mobility assistance, or rehabilitation. The survey then focuses in detail on the application of robotic rehabilitation following neurological disorder, looking at various patient requirements, recommended exercise tasks, and the associated control challenges. In the following chapter, the survey focuses on approaches to the control challenges of robotic rehabilitation, grouped by fundamental control strategy types. The control strategies are discussed and a commonly found problems with the existing strategies are identified.

The part of the survey focusing on wearable robots for assistance is limited to lower limb devices because this is the application area for experimental work in this thesis. The survey is also limited to wearable devices, so robotic systems which rely on external actuation systems and do not bear their own weight are not considered. The study is also limited to systems with direct joint actuation, so devices actuating using externally anchored pulleys and Bowden cables are not included. Multi joint and single joint devices are within the scope of this survey.

Assistance in the context of wearable robots describes additional force or torque applied to the human body which helpfully contributes towards achieving a desired task. A wearable robot is defined as a mechanical device that is designed around the form and function of the human body and can be worn by the operator, with segments and joints corresponding to those of the person it is externally coupled with [45]. Wearable robots can effectively integrate the cognitive ability of the human being and the mechanical capability of robotic devices to assist the user in accomplishing a desired activity [16]. We define exoskeletons as wearable electromechanical devices which augment the user's body either to enhance physical ability, provide support or perform rehabilitation exercise.

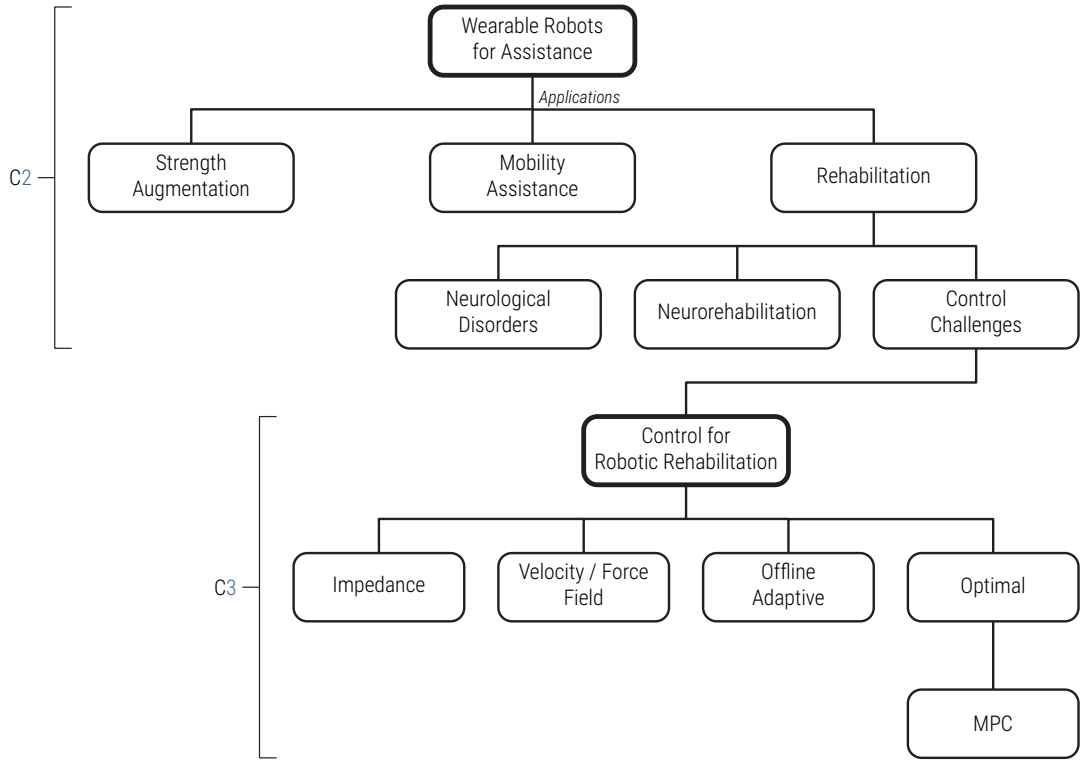


Figure 2.1: Map of Literature Review Areas. The root of the literature survey is the topic of wearable robots for assistance. From there, the literature is grouped into key application areas. The review then focuses on the application of robotic rehabilitation for neurological disorders. Approaches dealing with the control challenges for this area are the focus of the literature survey in the following chapter. The chapter concludes by identifying the major problems with existing approaches in order to identify a potential contribution to the research area of assist-as-needed control for robotic rehabilitation and assistance.

Rehabilitation is defined as the process of helping a person who has suffered an illness or injury restore lost skills and so regain maximum self-sufficiency. This survey focuses on the rehabilitation of motor skills specifically. The robotic rehabilitation applications considered are limited to neurological conditions only. This is because the bulk of current research efforts in the field focus on this type of rehabilitation. The secondary effects of conditions like incomplete Spinal Cord Injury (SCI) and stroke cause significant burdens on families, health services, and clinical rehabilitation services [24]. Findings and developments made in this research area often translate to other types of rehabilitation, such as following sports injury.

2.2 Overview of Robotic Assistance Applications

Wearable robots have an important role in providing physical assistance to humans in three main application areas. These are strength augmentation, mobility assistance, and rehabilitation. Strength augmentation applications aim to enhance the physical ability of able-bodied humans. Mobility assistance applications target wearers with physical impairments that either partially or entirely restrict mobility. Where patients have lost motor control, these applications aid mobility through direct actuation of the wearer's limbs. These applications intend to enable patients to stand up, sit down, ascend/descend stairs and inclines, and walk on level ground. Robotic rehabilitation applications also target wearers with mobility impairments, but here the aim is to restore strength, motor control, and independent mobility through gait training exercises. For these applications, patient participation should be encouraged and human-robot interaction requirements vary greatly depending on a number of factors including the stage of recovery.

2.2.1 Strength Augmentation

Strength augmentation exoskeletons can be further categorised based on the two application areas of the military and the workplace.

Military exoskeletons have been developed for fit and young soldiers that are required to perform physically demanding missions when they reach a combat zone. To assist in such cases, exoskeletons can amplify the capabilities of the soldier so that they feel only a fraction of a load being carried from one location to another and long distances can be covered with reduced metabolic cost. A prominent case of research and development is the Berkeley Lower Extremity Exoskeleton (BLEEX) [46], shown in Fig. 2.2a. It was developed at the University of California, Berkeley under a DARPA-funded project to develop a ver-

(a) BLEEX: Military [46]. **(b)** Hexar: Workplace [49]. **(c)** ReWalk: Mobility [50]. **(d)** Ekso: Rehab. [51]. **(e)** Lokomat: Rehab. [52].

Figure 2.2: Lower Limb Wearable Exoskeletons. Prominent examples of lower limb exoskeletons and their primary assistance applications.

satile transport platform for mission-critical equipment. It uses linear hydraulic actuators to allow soldiers to carry 100 kg without fatigue. Other notable military exoskeletons for similar applications include Sarcos XOS2 [47] (Raytheon, USA), HULC, and ExoClimber [48] (Berkeley Robotics and Human Engineering Laboratory, USA).

All of these devices have been designed to augment capabilities of soldiers in wartime during emergency operations. The main challenge in developing load augmentation exoskeletons for soldiers is reducing the overall system weight while keeping capacity for significant augmentation over long periods. However, the high torque actuators require lots of power and energy from large, heavy batteries to achieve this. Clearly development of longer-lasting lightweight batteries is needed for exoskeletons to have widespread impact in this sector.

Workplace exoskeletons have been developed for workers in industrial environments who have to perform physically demanding tasks which can lead to stress and adverse health effects. Tasks can include lifting heavy parts on an assembly, pick and place operations, and handling heavy equipment. Exoskeletons can help workers perform these specialised motions within defined reachable spaces. Workers can be supported during 3D joint movements to avoid health being compromised as a result of physically de-

(a) Handle™ (Boston Dynamics, USA) **(b)** Pick™ (Boston Dynamics, USA)

Figure 2.3: Sophisticated Workplace Robots. Robots for automating manual warehouse tasks [59]. Innovations such as these, which leave humans out of the loop, make the research and development of workplace exoskeletons less necessary and less commercially appealing.

manding motions. Examples of powered workplace exoskeletons include the PERCRO Body Extender [53], Innophys [54], Wearable-Agri-Robot [55], and Hexar [49], shown in Fig. 2.2b. For partially supporting loads on workers during workplace tasks, some passive[1] exoskeletons have been developed, for example the Fortis Exoskeleton (Lockheed Martin, USA) and ROBO-MATE [56].

Despite research and development efforts over the years, these systems are yet to be widely adopted for manual labour in industry. Alternative types of robotic automation solutions which leave humans out of the loop have prevailed over exoskeletons in workplace environments. Such systems have proven advantageous for reasons including increased efficiency, reliability, and human safety. Automated robotic systems are already widely used in warehouses and factories, with companies such as Amazon [57] and Ocado [58] integrating robots in their distribution centres for fulfilling customer orders with greater efficiency. Boston Dynamics have recently developed and marketed fully automated robotic warehouse solutions such as their Handle™ and Pick™ robots [59], shown in Figs. 2.3a and 2.3b respectively.

[1] Passive refers to devices which use components such as springs, pulleys, and dampers instead of active actuators such as DC motors which consume electrical power.

2.2.2 Mobility Assistance

Exoskeletons for mobility assistance applications can be further categorised based on the end user. Those user categories include patients suffering from various types of paralysis, referred to here as paraplegia, and otherwise healthy people suffering from mobility problems, such as the elderly with reduced physical ability or those suffering from a sports injury, here referred to as partial mobility.

Paraplegia is an impairment in motor or sensory function of the lower extremities. It is usually caused by complete SCI or a congenital condition that affects neural elements of the spinal canal. Individuals with paraplegia can range in their levels of disability, so mobility assistance requirements can vary from case to case. In severe cases, individuals may have full or partial lower body paralysis. Paralysis is the inability in motor and sensory functionality preventing motions such as standing and walking. Paralysis cases can be categorised by the regions of the body affected: quadriplegia cases affect the whole body from the neck down, hemiplegia impairs one side of the body divided by the sagittal plane, and paraplegia affects the body from the waist down. A number of mobility assistance solutions have been used by paraplegics including braces and crutches, wheelchairs, and orthotic[2] devices. Braces and crutches do not provide full motion autonomy and so have limited use. They are also unsuitable for severe cases of quadriplegia. However, they are acceptable in less severe cases of paralysis for short-term help.

Aided locomotion through conventional means such as wheelchairs and mobility scooters can provide long-term mobility support in large open plan areas. However, major structural changes are required for these to be used in public buildings and traditional homes. Such adjustments are becoming more widespread as society advances and becomes more inclusive, but accessibility is still limited today. Wheelchairs provide efficient movement on flat surfaces but cannot be used in tight spaces or over unstructured ter-

[2] Orthotics is a medical speciality that focuses on the design and application of orthoses. An orthosis is an externally applied device used to modify the structure and functional characteristics of the neuromuscular and skeletal system [60].

rain. They result in the user sitting in the same posture for extended periods of time, which can lead to secondary health problems. They also do not permit eye-level interactions with non wheelchair users and this can contribute to social exclusion. Despite this, wheelchairs are often the only long-term mobility option available, with no economically viable alternatives, meaning people are forced to move to single storey housing, install chair lifts and ramps, change fittings that are too high or low and so on. Such changes come at a significant financial cost for individuals, even if the initial cost of a wheelchair or mobility scooter may be attractively low. The overall cost of wheel-based mobility adoption for an individual can be so significant that the prospect of investing in a body-fitting exoskeleton becomes attractive. In principle, with a mobility assistance exoskeleton, minimal changes to homes and lifestyles are needed for users to be active and independent. This option could potentially allow people suddenly in need of mobility assistance to continue living in the same home without significant structural modification.

There are several mobility assistance exoskeletons developed for upright walking of paralysed people. A notable example is the ReWalk exoskeleton [50] (ReWalk Robotics, USA) for SCI patients, shown in Fig. 2.2c. It is a patient-worn device which is self-contained and includes a backpack. It uses rechargeable batteries to drive the hip and knee joints. Tilt sensors compute the trunk angle and a wristwatch-style device is used to manually activate different motion modes such as stand-to-sit, sit-to-stand, and walking. The safety and performance of the ReWalk for restoring ambulatory function in complete SCI patients was studied in a clinical trial involving 12 subjects. It demonstrated considerable potential as a safe ambulatory powered orthosis, with most subjects achieving a level of walking proficiency close to that needed for limited community ambulation [61].

Another device for SCI patients is the Indego exoskeleton [62] (Vanderbilt University, USA) which offers a modular mechanical design, consisting of a hip brace, two thigh frames, and two shank frames. It can be assembled quickly, adjusted with a single hand, and is easy to don and doff. It is to be used with

conventional ankle-foot orthoses. It has six modes for sitting, standing, walking, and all transitions. Other devices such as Ekso [51] (Fig. 2.2d), Rex [63], and Mindwalker [64] have also been developed for SCI and paraplegic patients.

The ReWalk, Ekso, and Rex exoskeletons have been approved by medical regulators such as the United States Food and Drug Administration. Similar approval is a prerequisite for commercialisation in most territories. Although commercially available, the usefulness of these devices is still in doubt as there are various technical and social issues in the way of their widespread adoption. Technical barriers include a lack speed compared to healthy human motion, excessive weight of the devices, requirement of heavy battery packs for use outdoors, comfort issues for both short and long periods of wear, and complexities introduced with sensing and actuation components. Social issues include hesitant attitudes toward the use of assistive technologies in activities of everyday life, high capital costs, and unintuitive human-robot interaction.

Partial mobility exoskeletons are intended for users who have some physical limitations but are otherwise healthy, for example the elderly or those suffering from a sports injury. There is a rapidly growing number and proportion of older people in the global population [65]. Therefore there is a need to shift the focus of assistive technology development towards their needs. This can be achieved by creating assistive exoskeletons that can help them to stay independent and maintain a good quality of life long into old age. The physical functionality of elderly people degrades with age as they usually become less mobile and get tired more quickly. Flexible and adaptive assistive technology should be developed so that the level of assistance can adapt to meet the changing needs of an individual without having to change the device. Despite global concerns, only a few multipurpose exoskeletons have been developed which can be used by the elderly or partially injured and no commercial models are available to date.

PhaseX is a spin-out company created by University of Gälve in Sweden which aims to commercialise assistive exoskeleton technology developed within the Active and Assisted Living (AAL) Call 4 project EXO-LEGS [66]. Simplified lower limb mobility exoskeletons able to support key human motions were developed [67]. Such wearable exoskeletons have the potential to meet the mobility requirements of partially limited people during daily living activities. For such devices to succeed, the intention must be to create user-centric solutions which can become affordable mass-market consumer products, rather than high-value low-volume clinical devices.

2.2.3 Rehabilitation

Rehabilitation is necessary for individuals with gait disorders, which can result from a variety of medical conditions such as traumatic brain injury, incomplete SCI[3], and stroke. Because of the ageing population, more people are suffering from acquired neurological disorders. The recovery of gait is an integral part of rehabilitation and often influences whether a patient can return home or to work [69]. Many individuals affected by a neurological impairment fail to recover the functional use of the lower limbs, even after prolonged rehabilitative treatment. These limitations come at great economic cost and are responsible for a significant reduction in quality of life [70]. In all cases rehabilitation should be started as early as possible.

As manual therapy is physically demanding for both the patient and therapist, robotic exoskeletons have been investigated to increase therapy efficiency. Robots are capable of providing more intensive patient training, quantitative feedback, and improved functional outcomes [71]. Medical gait training exoskeletons can be either stationary treadmill-based or wearable for over-ground training.

[3]Unlike a complete SCI, which results in the complete loss of function below the point of injury, with incomplete SCI only part of the spinal cord is damaged and some motor and sensory function remains. The remaining functionality varies greatly from person to person. The distribution of incomplete and complete lesions is currently about 50/50 [68].

Developed in the late 1990s and still widely used today, one of the best examples of a robot-assisted gait training exoskeleton achieving clinical success is the Lokomat (Hocoma, CH), shown in Fig. 2.2e [72]. It is a treadmill-based Body Weight Supported (BWS) rehabilitation system. It includes audio-visual biofeedback, using a screen in front of the user and is powered at the hip and knee joints. The Lokomat has demonstrated feasibility for locomotion training with severely affected patients after stroke [73], [74], and incomplete SCI [75], [76]. The system has also demonstrated a clear decrease in therapist exertion and reduction in the number of personnel required compared with manual gait training.

Wearable lower limb exoskeletons such as the ReWalk and Ekso, already discussed for mobility assistance applications in Sec. 2.2.2, have been investigated for over-ground gait rehabilitation with incomplete SCI and stroke patients. The effects of gait training with the ReWalk exoskeleton on Quality of Life (QOL) following incomplete SCI was investigated. Patient improvements were recorded in QOL, mobility, risk of falling, motor skills, and control of bladder and bowel functions [77].

The Ekso (Ekso Bionics, USA) is designed to assist rehabilitation of patients with various levels of lower extremity weakness caused by incomplete SCI and stroke [78]. The device has 6 DOFs (3 per leg). The hip and knee joints are actively powered, providing assistance in the sagittal plane, and the ankle joints are passively sprung. It features *Smart Assist*[4] which enables adjustable assistance from the suit based on individual patient needs. In a clinical study it was shown that gait training with the Ekso GT promotes learning of the correct step pattern in patients. All subjects showed improvements in gait spatiotemporal indexes (mean velocity, cadence, step length and step width). This allowed them to take a greater number of steps at a faster pace compared to traditional rehabilitation [81].

The H2 exoskeleton [22] has six actuated joints and is designed for intensive bilateral overground training. It uses an assistive gait algorithm, developed to create a force field along a desired trajectory, only applying

[4] Also known as *Variable Assist* in the US, this technology lets clinicians augment patients' training, allowing them to isolate specific aspects of the gait cycle in each leg for independent treatment with variable power [79], [80].

torque as patients deviate from the prescribed movement pattern. In clinical trials, the H2 was shown to be safe and easy to use for walking tasks. Customisation for individual patients and optimisation for specific rehabilitation treatments are possible with this system.

The Stride Management Assist (SMA) exoskeleton (Honda, JP) is intended for function task-specific training on bilateral spatiotemporal gait parameters [82]. It aims to enhance walking performance and increase community mobility and social interaction in elderly adults and patients with gait disorders. The SMA is worn around the hips and provides independent actively powered flexion and extension at each hip joint.

Hybrid Assistive Limb (HAL) provides motion interactively according to the wearer's voluntary drive [83]. It detects Electromyography (EMG) signals generated by the patient's muscle activities and floor reaction force signals caused by intended weight shifts. Whereas most similar devices use predefined motions for users, HAL generates motion paths based on the wearer's motion intention. HAL is designed to deliver unilateral or bilateral overground gait training without a treadmill.

AlterG Bionic Leg is a powered knee orthosis. It uses multiple sensors, including pressure sensors placed in the wearer's shoes, accelerometers, and joint angle detectors to create a model of the user's activity. It can assist during sit-to-stand transfers, walking on level surfaces, stair ascent, and stair descent, providing assistive torque at the knee joint. It is programmable so a trained physiotherapist can define parameters for unilateral gait training which are updated over the course of therapy [84].

In recent years, significant progress has been made in the mechanical design of wearable gait training exoskeletons [16]. Their feasibility for the rehabilitation of gait disorders caused by incomplete SCI and stroke has been demonstrated in studies published in influential peer-reviewed journals [68], [85]. Several devices have been successfully commercialised, however there sill exists a large gap in the research with respect to personalised and adaptable human-robot interaction. Wearers of devices often report discomfort during therapy sessions. Efforts should be made to make devices feel organic for the patient [71]. Although

functional outcomes are often similar to traditional therapy, there still exists significant variability in training performance from person to person. This inconsistency is observed less with traditional rehabilitation methods: trained therapists continuously make judgements and tailor treatments for individual patients at every stage of recovery [86]. This thesis will consider the application of wearable robots for robotic assistance and rehabilitation, specifically investigating how existing control approaches can be improved to make assistance more adaptable to the wearer's condition and stage of rehabilitation. The following section focuses on the clinical processes involved in robotic rehabilitation and the control problems that arise from highly variable patient requirements.

2.3 Focus on Robotic Rehabilitation

Rehabilitation robotics is a rapidly developing field that shows great potential for aiding people with movement disorders in their recovery, leading to improved quality of life and independence. Rehabilitation robots serve as physical aids to therapists and present new opportunities for innovating the recovery process. Increased motivation can be achieved with serious games[5] and interactive virtual environments [43], [87], [88]. Integrated sensing can provide quantitative feedback about patient progress and therapy outcomes, and enables the use of machine learning algorithms [31], [36]. Innovations such as these lead to more effective and enjoyable therapy methods.

Control strategies for rehabilitation robots differ from those of conventional industrial manipulators. A big difference is the nature of environment with which the robot interacts: ideal human-robot interaction requires highly variable mechanical compliance [89]. Another difference is the need for human-centred control regimes, with robot behaviour and controller priorities adapting based on information from the

[5] A serious game is one designed for a primary purpose other than pure entertainment: in this context motor and cognitive rehabilitation.

patient.

2.3.1 Neurological Disorders

Global average life expectancy is increasing thanks to advances in medicine and healthcare, so the demographic profile is shifting towards a more elderly population. Unfortunately, this shift is not translating to more healthy life years[6] at the same rate. Morbidity and disability rates are rising, with neurological disorders being a very common cause.

Stroke Cerebrovascular accident, or stroke, is the rapid loss of brain function due to disturbance in the blood supply, resulting in cell death. There are two main types of stroke: ischaemic, caused by blockage of an artery supplying oxygen-rich blood to the brain, and hemorrhagic, caused by an artery in the brain leaking or rupturing. Ischaemic is the more common type, occurring in around 85% of cases. There have been over 13.7 million new stroke cases worldwide each year since 1990 [24], leaving 5 million people permanently disabled [25]. The annual cost of strokes has been estimated at 51 billion dollars in the USA and 38 billion euros in Europe [26], [27].

According to the American Heart Association, an estimated 7 million Americans $\geq$ 20 years of age self-report having had a stroke. The prevalence of stroke in the US increases significantly with advancing age. Over the time period of 2006 to 2010, older adults had increased self-reported rates. Stroke is a leading cause of serious long-term disability in the United States. It was among the top 18 diseases contributing to years lived with a disability in 2010. Of these 18 causes, only the age-standardised rates for stroke increased significantly between 1990 and 2010 [91].

[6] Healthy life years are defined as the number of years that a person is expected to continue to live without disability in a healthy condition [90].

The age-standardised incidence of stroke in Europe, at the beginning of the 21st century, ranged from 95 to 250/100,000 per year. Approximately 1.1 million inhabitants of Europe suffered a stroke each year. Because of the ageing population, the absolute number of stroke cases is expected to rise dramatically in the coming years: by 2025, 1.5 million European people will suffer a stroke each year [92].

The UK societal cost of stroke exceeds £8 billion a year [5]. Following acute medical response, long-term patient outcomes are almost entirely dependent on rehabilitation and early initiation is critical. Ageing population projections indicate a 44% increase in strokes in the UK in the next 20 years [9]. Even partial muscle dysfunction impacts simple independence and activities of daily living.

The effects of stroke depend on the severity, type, and location in the brain. Muscle weakness is a common consequence and can result in a decrease in physical activity [93]. Changes in gait performance can be observed, including reduction in speed, increased asymmetry, and greater energy cost. Physical impairment can range from a slight weakness of extremities on the side contralateral to the lesion (i.e. partial hemiparesis) to total paralysis (i.e. hemiplegia). This effect does not originate from actual muscle damage, but rather a reduced number of neural pathways leading to them. Prolonged disuse leads to brain and muscle atrophy [94]. Loss of other functions is also common, such as speech, spatial ability, and cognition. One may recover completely from stroke, but more than two thirds of survivors will have some form of disability [95].

Spinal Cord Injury The spinal cord is the pathway that carries electrical signals back and forth between the brain and the peripheral nervous system. A trauma involving damage to these pathways will cause loss of function depending on its severity and location. Every year, around the world, between 250,000 and 500,000 people suffer an SCI. The majority of cases are from preventable causes such as road traffic crashes, falls, and violence [96]. More than half of all SCI patients will develop complications during the

initial hospital stay such as traumatic brain injury, and cerebrovascular damage. Unlike stroke, which is typically an elderly disease (almost three quarters of cases occur in people over 65) [92], The average age at the time of SCI is 32 [97], which necessitates life-long specialised care. An increase in SCI incidence among the elderly has been reported over the last few decades [98].

The nature of disability depends on the location where the spinal cord and nerve roots are damaged, and the extent of the damage. The severity of injury can be described in levels of *completeness*. A *complete* injury is a case in which there is absolutely no spine-mediated neurological function below the level of the injury. An *incomplete* injury is one in which there is some remaining function, typically in the form of intact sensation or slight motor function [99]. Incomplete injury carries a better prognosis for recovery of some degree of neurological function. Approximately 60% of patients with a SCI suffer an incomplete lesion [100]. An injury above the first thoracic vertebra will result in quadriplegia, and below it will result in paraplegia. SCI is a lifelong condition, requiring ongoing efforts in multiple disciplines for it to stabilise or diminish. Efforts should also be made to prevent or limit secondary impairments and complications, improve and maintain social functionality, and maintain quality of life for the individual.

Remarks It is clear that neurological disorders can have devastating effects on a person's independence and quality of life. Ideally these acquired disorders would be avoided entirely through public education and medical advances, but in reality they are a common occurrence so effective rehabilitation is a necessity. The principles of motor learning have been widely researched, and it is encouraging to see growing efforts in neurological rehabilitation research.

2.3.2 Neurorehabilitation

Until recently, it was thought that cortical maps were hard-wired early on in child development. Research in the latter half of the 20th century showed that many aspects of the brain can be altered, or are *plastic*, even through adulthood [101]. The developing brain exhibits a higher degree of plasticity than the adult brain [102]. This ability of the brain to change continuously throughout an individual's life is known as *neuroplasticity*. The purpose of neuroplasticity is to optimise neural networks. For example, brain activity associated with a given function can be transferred to a different location, the proportion of *grey matter*[7] can change, and synapses may strengthen or weaken over time.

Neurorehabilitation utilises activation of experience-dependent neuroplasticity to promote functional recovery. Exposure to specific training exercises leads to improvement by activating neuroplasticity mechanisms [104]. The aim is to minimise and/or compensate for any functional impairment following neurological injury [105]. It does not reverse cell damage caused by stroke or SCI, but instead works towards restoring neural functionality [106].

Recovery factors A combination of factors such as spontaneous recovery, compensation, and relearning are important for functional improvement [107]. There is usually some spontaneous recovery in the vast majority of stroke patients. This is most pronounced over the first month and can continue over a period of several months. Unfortunately, recovery is typically partial, and patients are usually left with some disability. It is difficult to assess for true neurological recovery. With recovery, the objective is to restore lost function. Compensation, in contrast, results in functional improvements via engaging the unaffected limb, using assistive devices, or modifying the movement sequence [108]. From a clinical perspective and the patient's point of view, the distinction between physiological recovery versus substitution may not be

[7] Grey matter contains most of the brain's neuronal cell bodies. It includes regions of the brain involved in muscle control, and sensory perception such as seeing and hearing, memory, emotions, speech, decision making, and self-control [103].

important if disability is minimised and quality of life restored. However, this distinction is fundamentally important for understanding mechanisms of neurological recovery.

Motor learning is the acquisition and retention of new motor skills (or improvement of existing ones) through practice and experience. It is a practical application of neuroplasticity, and thus is a relatively permanent change compared to *motor adaptation*.[8] The cerebellum and basal ganglia play a major role in this process [109].

Voluntary movements are controlled by a combination of feedback and feedforward mechanisms. The sensory feedback from *afferent neurons*[9] exhibit long time delays. Delay feedback has been suggested as a possible mechanism for instabilities in muscle control, leading to behaviours such as tremor [110]. Our motor systems rely mostly on feedforward processes to control motion. However, feedforward systems lack the sensory feedback necessary for compensation of external changes (e.g. perturbations, musculoskeletal changes, etc.). That is where motor adaptation and sensory feedback prove very useful [111].

Motor learning guidelines Although the specific mechanisms underpinning neurorehabilitation are not fully understood, there is clinical evidence of the effectiveness of motor learning- (or relearning-) based approaches over conventional therapy methods [112]. There is strong evidence that task-oriented repetitive training can result in functional gains [113]. Task-specific activity is a critical factor in structural and functional changes in the region of the brain responsible for motor recovery [114]. It has been shown that a rehabilitation exercise should be repetitive, intensive, and task-specific:

[8] Motor adaptation is the process of acquiring and restoring locomotor patterns (e.g. leg coordination) through an error-driven learning process.

[9] Afferent neurons are sensory neurons that carry nerve impulses from sensory stimuli towards the central nervous system and brain, while efferent neurons are motor neurons that carry neural impulses away from the central nervous system and towards muscles to cause movement.

Repetitive From lived experience it is clear that in order to learn a task, it should be repeated over and over again. However, repetition alone is not enough to learn or relearn a skill and for permanent changes to be made in the motor cortex [115]. To achieve this, individuals must undertake progressively more challenging tasks. Also, repetition with some variation is shown to have better outcomes than exact repetition. Introducing small variance to repetitive tasks has been termed *repetition without repetition* [116].

Intensive refers here to uninterrupted therapy sessions that go on for a long time. Sessions considered to be intensive can last a number of hours [117]. Some researchers argue that only by increasing the intensity and total hours of rehabilitation can we achieve better clinical functional outcomes [118].

Task-specific Task-oriented therapy is important. It makes sense that the best way to learn or relearn a given task is to train for it specifically. In animals it has been observed that functional neural reorganisation is greater for tasks that are meaningful. Repetition alone, without functional meaning, is not enough to produce increased motor cortical representations[119].

Although the effectiveness of the aforementioned methods has been demonstrated in randomised clinical trials, there is still a lack of consensus about the timing, duration, and design of rehabilitation routines for patients with neurological disorders. The ideal solution for patients will vary significantly depending on their pathology and condition. Regardless of how a rehabilitation training strategy is devised, it should include the three characteristics discussed.

Exercise modes A rehabilitation exercise, robot-assisted or traditional, may be carried out in one of three modalities [120]:

Passive or externally imposed movement involves manipulation of the patient's limbs by the therapist or robot while the patient remains relaxed. It maintains the patient's range of motion at joints and flexibility in the muscle and connective tissue. It can also help the patient retain or re-establish important proprioceptive information about the achievable workspace that the impaired limb should reach as recovery progresses.

Active-assisted movement is appropriate for when the patient cannot complete a desired movement independently. External assisting forces are applied as needed during attempts by the patient to move a limb. Forces can be manual manipulation from a therapist or robot, or from constraints placed in the environment to guide movement.

Active-resisted movement is used in advanced stages of recovery and involves the patient completing movements against resistance from gravity, additional weights, elastic bands, therapist, or robot.

Rehabilitation exercises can also be classified based on their kinetics: isotonic, which refers to constant force/tension; isokinetic, which refers to constant velocity; and isometric, which involves no change in muscle length (i.e. static).

Training approaches Each patient should have a customised rehabilitation program that is iteratively refined. Important steps include: assessing needs, establishing attainable goals, developing progressive interventions matched to ability, and evaluating progress over time. Developing a task- and content-specific training program can improve engagement and achieve functionally important outcomes [121]. An example approach is constraint-induced movement therapy, which forces use of the affected limb by restraining the unaffected limb and emphasises intensive practice [122]. There are clear and sustained benefits to this

kind of intensive rehabilitation over traditional care. Early, intense, repetitive behavioural training is essential for optimal motor recovery. Large randomised clinical trials support the functional effectiveness of intensive progressive task-oriented training of affected limbs with neurological patients. However, it is shown that the anatomical and physiological consequences of injuries will not only limit the degree of recovery, but also determine the effectiveness of specific rehabilitative strategies. Training strategies should also aim to improve capacity, especially in the challenging outpatient setting [123].

SCI v Stroke A brain injury such as stroke affects the function of the cerebral cortex while the spinal cord which carries the motor signals is preserved. On the other hand, after an SCI, it is the nerve fibres which are damaged, while the cerebral cortex remains intact. Although the locations of these injuries are different, the impact on the motor system and the functional disabilities are similar. There is a case for neurological disorders leading to functional motor disabilities to be considered in terms of common denominators, and for developed rehabilitation solutions to be applied and adapted to all [124]. The gait training of stroke patients does differ in some ways from that of SCI patients. The main difference is that individuals with SCI tend to have paralysis on both sides of the body while most stroke patients suffer from hemiplegia [72]. Following a stroke the muscle weakness, muscle tonus, balance, and cardiovascular condition decrease walking velocity and endurance, increasing the level of disability [125]. Lower extremity strengthening exercises and task-specific training can be used to recover walking ability in individuals [126]. More strenuous training is usually recommended following stroke than SCI. This means higher gait speeds are set for exercises and can result in sessions being reported as exhausting by stroke patients [127]. The process of restoring walking function in patients with neurological pathology is made challenging by the inherent complexity and variability.

Clinical predictors of recovery General characteristics and detailed clinical assessment can predict the extent of recovery [107], [128]. In addition to predictors such as age, medical comorbidities, and initial stroke management, motivation and adherence can also have an impact on recovery. Intense rehabilitation requires significant motivation and dedication. In addition to requiring a great amount of resources, it necessitates meaningful participation and lifelong commitment to maintenance of function. This is especially challenging for survivors of neurological accidents with concurrent deficits in attention, learning, sensory function, and motivation (*abulia*[10] may accompany certain strokes, for example).

Robotic devices Rehabilitative robots may provide more accurate feedback and allow finer progressive changes in task practice to heighten recovery function [123], [130]. In addition, robots can relieve therapists from time-consuming supervision of repetitive, progressive practice [131]. BWS treadmill training may have limited additional benefits if not supplemented with strengthening, balance, and overground gait training [132]. Robotic devices may also be used at home and can connect to game-oriented practice with rewards and real-time feedback. Virtual reality and video game applications are rapidly emerging as a potentially motivating technology in combination with rehabilitation robots [133].

2.3.3 Control Challenges

The neurorehabilitation information and training guidelines provided in Section 2.3.2 present a number of control challenges for rehabilitation robots. Although mechanical structure and actuation are key, the most significant challenge is the design of control strategies which enable robots to deliver suitable assistance across the broad spectrum of rehabilitation training. The most difficult aspects of neurorehabilitation in terms of control are summarised:

[10] Abulia is an absence of willpower or an inability to act decisively, a symptom of schizophrenia or other mental illness [129].

Pathology variability The nature of pathology and therefore rehabilitation requirements can vary significantly from case to case. Each patient's training strategy should differ based on their severity of neurological disorder and range of motion. The training approach should be specific to their anatomical needs and progress throughout the course of their recovery. Control approaches should be applicable to a wide range of disorder severity, and identify the degree of pathology in order to adapt training accordingly.

Task-variability It is crucial for rehabilitation training approaches to achieve a balance of *repetition without repetition*. Therefore, control strategies should not be repetitive and should involve some variability, even when conducting intensive and repetitive training.

Task-specificity Training should be tailored to individual patient's functional ability and control strategies should be task-specific. Approaches that practice a task which is not useful in daily life are unlikely to lead to significant improvements in functional outcomes and quality of life. Control tasks need to have meaningful function in daily life in order for training to have the most long-term neurological impact.

Exercise modes Mode of exercise should change at different recovery stages, with passive, active-assisted, and active-resisted exercise being fit for early, intermediate, and late stages of recovery respectively. Control strategies usually target one of these exercise modes, however, an ideal control solution would identify the appropriate exercise mode at a particular time and adapt the control policy accordingly.

Patient motivation Survivors of neurological disorders often suffer concurrent deficits in attention and motivation. An effective predictor of stroke recovery is patient motivation and adherence to training task. Controller designers should provide solutions that actively engage the patient. Engagement can be achieved with novel control strategies or by combining the controller with additional technology such

as a user interface or serious game.

Patient safety A critical consideration for any control approach intended for a medical application with human subjects is patient safety. It should be embedded in every stage of the design and testing process. Safety is also a major concern in the mechanical design of any device intended for human-robot interaction.

The design of control strategies for rehabilitation robots presents significant challenges. In the following chapter, existing control approaches will be presented and explored in detail. Common problems in the existing approaches are identified, and on that basis the contribution of this thesis is proposed.

Chapter Summary

This chapter provided context for the work in this thesis in the form of background information on wearable robots for lower limb assistance. Notable examples of existing wearable robots were provided and discussed, categorised by the application areas of strength augmentation, mobility assistance, and rehabilitation. The chapter then focused on the robotic rehabilitation area, considering neurological disorders, factors involved in neurorehabilitation, and the control challenges that arise. Although design of controllers for robotic rehabilitation is challenging, it is a significant research area with great potential impact if the difficulties can be overcome. Existing approaches to this problem area are considered in the following chapter.

Chapter 3

Control for Robotic Rehabilitation

The previous chapter provided background information on lower limb wearable robots for rehabilitation to establish context for the work in this thesis. Notable examples of existing robots were categorised based on applications and discussed. The review then focused on the application of rehabilitation robots for neurological disorders and provided detail about requirements and challenges in this area. Robot control for effective neurorehabilitation was highlighted as an area of interest.

In this chapter, examples of existing approaches for the control of rehabilitation robots are provided. The field is firstly divided by exercise mode. The training modalities that exist in neurorehabilitation are passive, active-assist, and active-resist. Control approaches are usually designed for one of these training modes, intended for a particular phase of motor recovery. The modalities are further explored in this chapter and active-assist is highlighted as an area of interest.

Control approaches targeting the active-assist exercise modality are further investigated and categorised based on control strategies of impedance, velocity/force-field and offline adaptive. Upper limb and lower limb robotic rehabilitation applications are considered as control methods can be applied to both with the

same underlying motor learning principles.

Existing control strategies are compared and discussed in terms of commonly found problem areas that are recalled to establishing research aims for a novel controller.

3.1 Training Modalities

There are three training modalities used in the process of neurorehabilitation, corresponding to the three exercise modes given in Section 2.3.2: passive, active-assist, and active-resist. Traditionally a therapist delivers different treatment in each modality, using their judgement and expertise to adjust the nature of therapy as they see fit. Passive exercise is suitable for the early stages of rehabilitation, before a patient is able to move their limb independently. However, the effectiveness of passive motions for stimulating neuroplasticity is known to be limited. Active-assist exercise is for when the patient has some mobility but cannot complete a desired movement independently. Active-resisted exercise is used in latter stages of recovery, involving the patient completing movements against resistance.

For robotic rehabilitation devices, delivering the appropriate mode of treatment, and doing so efficiently, presents a significant two-part challenge. Firstly, the robot has the task of assessing patient progress, and secondly, the robot must deliver the appropriate modality of training exercise. Where therapists are able to make a hands-on assessment of patient progress and use their judgement to decide the appropriate type of exercise, this is not possible with robotic rehabilitation. Therapists also have training and experience which enables them to skilfully deliver different modes of training. It is therefore important for a robotic system to make a quantitative assessment of patient progress to achieve autonomy. It is an option to have therapist and robot working together: the therapist using their expert judgement to make a quantified assessment of patient progress, and the robot, informed with patient progress, delivering the appropriate

exercise modality. Even with patient assessment left to the therapist, rehabilitation robots still face the challenge of correctly delivering the therapy modality. This is where the control strategy plays an important part.

Developing a control strategy for the passive training modality is relatively straightforward. This type of exercise and human-robot interaction can be achieved with basic servo-type position control. A well-tuned Proportional Integral Derivative (PID) controller, for example, can track joint angles with little error and reject human dynamic contribution, treating it as an external disturbance. This condition is entirely robot dominant.

The active-assist and active-resist training modalities present a more interesting control challenge. In these conditions neither robot nor human are dominant. Instead, interaction is cooperative and falls on a spectrum, with servo-type robot dominant position control at one end, and completely transparent zero-impedance compliance at the other. Control approaches must dictate the nature of this human-robot interaction in order to encourage patient participation in rehabilitation task and promote motor recovery through neuroplasticity mechanisms. It is essential for the robot to allow shared control of movements, and devices must allow patients to express whatever movement they can without suppressing motor capacity. The training modality of active-resist is far less common, as it is only applicable at the very late stages of motor recovery, when a subject has regained significant mobility and independence. For this reason, there is relatively little research interest in the modality. The focus of this review and the contribution area of the research in this thesis is control for active-assist exercise. It has been shown through randomised clinical trials that robotic implementation of active-assist exercise can lead to patient outcomes comparable to those of traditional therapy [71]. Despite promising results, an optimal control strategy for active-assist exercise is yet to be developed, and there are numerous issues with existing methods which will be explored in this review. The delivery of active-assist exercise is commonly referred to as partial as-

sistance or Assist-as-Needed (AAN) [134], which will be used in this thesis. The following section explores AAN robotic control, comparing and contrasting existing approaches, highlighting common issues, and establishing a research question for this thesis.

3.2 Assist-as-Needed Control Approaches

AAN control approaches are divided into categories based on the type of control strategy: 1) Impedance, 2) Force and/or Velocity Field, and 3) Offline Adaptive. These categories serve to conveniently group existing methods, and so it should be noted that within each category the implementations vary and some approaches may contain elements of multiple control strategies. It should also be noted that upper and lower limb applications are grouped together, as control methodologies are considered to be applicable to upper and lower limb rehabilitation robots.

3.2.1 Impedance

Instead of rigid industrial-type position feedback controllers with high corrective gains, many AAN control approaches rely on more flexible impedance control [135]. Reduced corrective gains are used to exhibit human-like mechanical properties. These controllers enable a compromise between tracking performance and compliance of the robot. The beneficial force-tracking trade-off feature, together with simplicity of implementation, make impedance control algorithms some of the most commonly used for rehabilitation exoskeletons.

Impedance can be represented in Laplace notation by a transfer function $Z(s)$ that relates the position $y(s)$, and its derivatives with respect to time, with the force $F(s)$ of the end-effector of a robotic manipu-

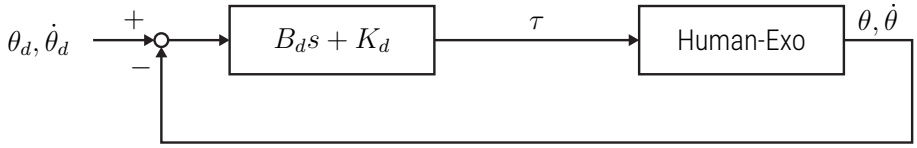


Figure 3.1: Simple Impedance Control. Schematic of a basic impedance control framework for a reference angle, θ_d, and angular velocity, $\dot{\theta}_d$, with parameters of stiffness, K_d, and damping, B_d. The parameters dictate the nature of human-like properties attributed to deflection about the reference position.

lator.

$$Z(s) = \frac{F(s)}{y(s)} = I(s)\ddot{y}(s) + B(s)\dot{y}(s) + K(s)y(s) \tag{3.1}$$

where the zero-order $K(s)$ term is the stiffness and describes the relationship between force, $F(s)$, and the position, $y(s)$; the first-order $B(s)$ term is the damping and describes the relationship between force, $F(s)$, and the velocity, $\dot{y}(s)$; and the second-order $I(s)$ term is the inertia and relates force, $F(s)$, and the acceleration, $\ddot{y}(s)$.

A commonly used impedance control law, which ignores gravitational and frictional effects and assumes low acceleration values ($\ddot{\theta} \approx 0$), meaning the second order inertial effects are negligible, can be written as

$$\tau = -B_d\dot{\theta}_e - K_d\theta_e \tag{3.2}$$

where τ is the vector of control input torques, B_d and K_d are symmetrical and non-negative matrices representing the damping and stiffness of the robot, θ denotes the angular position vector, θ_d is the desired position vector, and $\theta_e = \theta_d - \theta$. The block diagram for this scheme is shown in Figure 3.1.

This is a simplified formulation which can be implemented relatively easily and represents the most fundamental aspects of impedance control, which are the stiffness and damping parameters. There have been numerous impedance control implementations of varying complexity for robotic rehabilitation and all of them can be characterised in some part by these two parameters. The position and velocity errors are used

to apply stiffness and damping effects respectively. To successfully implement first-order impedance control, it is important to have a low inertia backdrivable system which is responsive to external interaction [40].

This basic Proportional Derivative (PD) form of the impedance control given in (3.2) is commonly implemented in robotic rehabilitation [120], [136], [137]. The first implementation in robotic devices for lower limb rehabilitation was with the Lokomat [138]. The authors used impedance control consisting of an inner Proportional Integral (PI) force control loop and an outer PD position control. A PD computed-torque form of impedance control is implemented in a parallel manipulator for ankle rehabilitation in [139].

A novel concept of performance-based progressive robot therapy that uses speed, time, or Electromyography (EMG) thresholds to initiate robot assistance was described in [140]. This pioneered the clinical application of robot-assisted therapy focusing on stroke, the largest cause of disability in the US. Task-specific, goal-directed, therapy was used to reduce motor impairments in the affected arm. The work aimed to determine the optimal therapy tailored to each stroke patient to maximise their recovery. The method used to achieve this was a performance-based impedance control algorithm which, like a line integral, specifies the initial and final conditions and the path between these conditions, quantifying each property with specific performance measures.

In the simple impedance model (3.2) gravitational and frictional effects are ignored. This can cause problems in systems where dynamics are significantly affected by gravity or friction and so additional compensation may be necessary, such as in [141]–[144]. The block diagram for this control scheme is shown in Figure 3.2. Model-based gravity compensation was present in [145] where the exoskeleton was controlled through two impedance control loops, one for assistance tangential to the trajectory, and one corrective in the normal direction. Because humans show variability in their movements, a deadband can be introduced into impedance control schemes to allow for normal variability without robot intervention. [29].

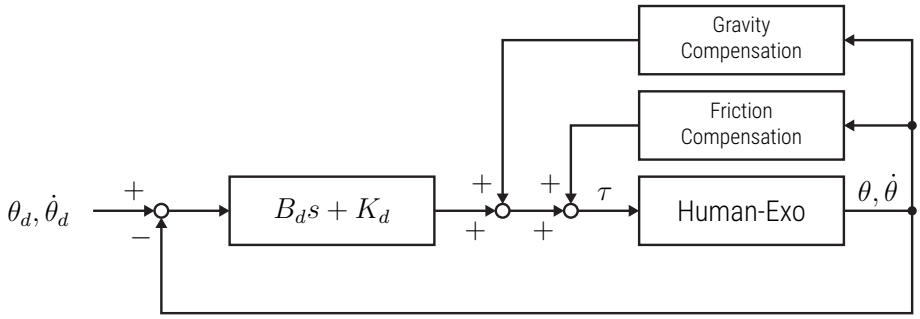


Figure 3.2: Compensated Impedance Control. Schematic of an impedance control framework which includes model-based gravity and friction compensation. The compensation enables more compliant and human-like stiffness and damping parameters to be used while tracking errors caused by gravity and friction are minimised.

Robot-supported therapy can improve rehabilitation by allowing more training. This was explored with the arm therapy robot ARMin in [146]. It is a haptic display with semi-exoskeleton kinematics, with four active and two passive DOFs. It delivers patient-cooperative arm therapy taking into account the activity of the patient, supporting and assisting as-needed. The implemented control strategy is based on an impedance controller, where the assisting force is determined based on the relative distance between a target and the human hand position. Model-based gravity and friction compensation are present in the impedance controller. The stiffness and damping parameters of the controller are constant during operation, however can be adjusted by the therapist.

The AAN control problem was addressed in [147], where Asl et al. designed a stable AAN controller with adjustable assistance level. Building on the compensated model-based impedance control method shown in Figure 3.2, the controller aims to guarantee the stability of the system and includes a bounded PD (stiffness and damping) term with dead-zone area, providing a tunnel of freedom around the trajectory tracking error. This approach includes the advantages of both the model-based and non-model-based AAN controllers. The controller also uses an adaptive Neural Network (NN) term to compensate for the unknown dynamics of the system, which has a forgetting factor to enforce AAN behaviour, by preventing

the user from overly depending on the NN contribution to complete the task.

The dual of impedance control is admittance control. This uses position control with force sensor feedback. It is best suited for exoskeletons that lack backdrivability because the forces at the interfaces with the human must be measured to move the robot. This approach is used in [148]–[150], where haptic interfaces are used to control arm movements. In [142], a model-free PID-type admittance control was used in task space to generate desired trajectories for a low level PID position control. A simplified model-based approach was used for gravity compensation.

Impedance control enables real-time variations in the amount of assistance via variation of the impedance parameters (stiffness and damping) of the controller, without significant stability problems. The parameters of an impedance controlled device can determine the level of assistance offered by the robot, such that the change of robotic assistance (or resistance) remains associated with the parameters of the controller. Impedance control can be used as a higher level control that acts on force or position control, so that the system behaves as an impedance [151]. On top of the impedance control, an additional control layer can be implemented which changes the desired impedance based on an algorithm. For adaptive impedance it is common to have three control levels: the top is where the desired impedance is determined based on the rehabilitation/assistance strategy and environment of the system; the middle is where the desired impedance is used to generate reference torques (or position for admittance control); and the bottom is the inner loop in the form of force or position control, which maintains the desired torque and acts directly on the actuators. Various methods have been used to control the top level of this heirarchy, which achieves adaptive AAN behaviour. In [152] the Lokomat researchers modified the impedance parameters so that the stiffness was varied linearly depending on the knee force, and the damping parameter was adjusted to achieve critical damping. In [153], the authors designed a set of impedance parameters and desired trajectories for each exercise mode. A number of other approaches have been developed for adapting

impedance parameters to achieve AAN control, such as in [42], [154], [155].

3.2.2 Velocity / Force Field

Impedance control approaches use a timed dependent trajectory as the reference position of the robot. This has the disadvantage of imposing a fixed timing of movement. In an effort to gain freedom in movement timing, the path control method was proposed in [156]. This allows patients to influence the timing of their leg movements along a physiologically meaningful path. Compliant virtual walls keep the patient's legs within a tunnel around the desired spatial path. The method limits timing freedom using a moving window to construct new position references, which travel with a predefined speed on the path.

An alternative approach for gaining timing freedom is the force field control method presented in [157]–[160]. Force field controllers apply a force field at the end-effector of a robotic rehabilitation device, for example the ankle is used as the reference point in [157]. The force field is broken down into tangential and normal components, with respect to the direction of the desired path. Tangential forces assist with completion of the task and normal forces correct for deviations from the desired direction of movement. The normal forces create a virtual tunnel around the desired trajectory. The equation for the normal force is

$$\mathbf{F}_n = \begin{cases} K_n \left(\frac{|d|-D_0}{D_n}\right)^2 \hat{\mathbf{n}}, & \text{if } |d| > D_0 \\ 0, & \text{otherwise} \end{cases} \tag{3.3}$$

where d is the distance from the end-effector (point $\mathbf{p}$ in Figure 3.3) to the nearest point on the desired path $\mathcal{C}$ (point $\mathbf{Q}$ in Figure 3.3), D_0 is defined as the tunnel width, D_n is a constant with length units which is used to change the shape of normal force $\mathbf{F}_n$ versus d. K_n is a constant with force units, which can be considered a stiffness parameter, and $\mathbf{n}$ is a unit vector pointing from $\mathbf{p}$ to $\mathbf{Q}$, as shown in Figure 3.3. This creates a nonlinear parabolic normal force, which works like a stiffening nonlinear spring in helping

to guide the end-effector along the desired trajectory $\mathcal{C}$.

The tangential force is defined as

$$\mathbf{F}_t = \begin{cases} K_t \left(1 - \frac{d}{D_t}\right) \hat{\mathbf{t}}, & \text{if } \frac{d}{D_t} < 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

where K_t and D_t are constants. K_t is used to change the maximum magnitude of the tangential force. $\mathbf{F}_t$ is maximum along the desired path (where $d = 0$), its magnitude decreases linearly and becomes zero at a distance of D_t from the desired path. The tangential force ramps down as the distance d increases, this is to apply assistive force only when the limb is closer to the desired path. This component of the force field is necessary to help the subject complete a task and can be scaled during the process using K_t.

A damping force is usually added which is given by $\mathbf{F}_d = -K_d \dot{\boldsymbol{x}}$ where K_d is a constant damping parameter and $\dot{\boldsymbol{x}}$ is the linear velocity of the end-effector point in the field. The total force to be applied at the point $\mathbf{p}$ is given by the vector sum $\mathbf{F} = \mathbf{F}_t + \mathbf{F}_n + \mathbf{F}_d$. The force field controller helps a stroke survivor during rehabilitation using the AAN strategy. Wall constraints are set to maximum to begin with during initial rehabilitation, so the guidance provided by normal forces generate a very narrow virtual tunnel. The tangential force required to assist the patient along the desired path is kept to the minimum necessary to complete the movement. This is controlled by parameter K_t. This heavily controlled dynamic is useful for initially providing the patient with proprioceptive feedback, and as the patient's performance improves, the tangential force can be further removed to a small value. This causes the width of this tunnel to increase over time in training and the patient is able to have more control over movement. While this control strategy provides significant timing freedom, the generated guidance has the potential to fall short in some applications, not adhering to the intended task-specific trajectory due to a lack of feedback-driven adaptation.

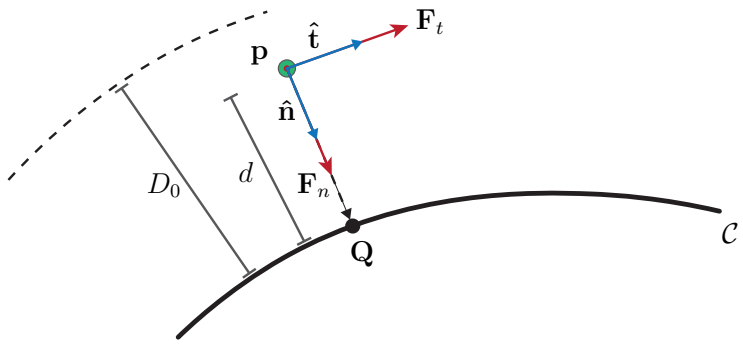


Figure 3.3: Force Field Construction. Illustration of the force field control field construction showing the end-effector point in space, $\mathbf{p}$, normal unit vector, $\hat{\mathbf{n}}$, normal force vector, $\mathbf{F}_n$, tangential unit vector, $\hat{\mathbf{t}}$, tangential force vector, $\mathbf{F}_t$, distance, d, to the nearest point ($\mathbf{Q}$) on the desired path, $\mathcal{C}$, and tunnel width, D_0.

The authors in [160] presented an AAN controller for a hand exoskeleton called Maestro, designed to provide accurate torque assistance to subjects for hand rehabilitation. A novel technique called learned force field control was proposed, in which a neural network-based model of the required torque is learned offline for a specific subject then used to render a force field to assist the finger motion to follow a target trajectory. Although the force field control is non-adaptive and there is limited control over speed of execution of the task, it did demonstrate safety benefits for subjects, as excess torque is not applied to joints in the case of motion being restricted.

Control in the velocity-domain is another method proposed, aiming to achieve freedom in timing of movement. This approach is called velocity field control and involves desired velocity values shaping the desired path, rather than position or force references [161], [162]. Desired velocities are a function of mechanical configuration rather than time, enabling the approach to deliver timing freedom under the condition that the controller is forgiving and does not compel for precise velocity tracking.

To design a velocity field controller, the desired path to be travelled should be encoded by velocity vectors in the configuration space of the robot. The velocities can be constructed either in Cartesian space [161] or joint space [163]. The velocity field vector, $\mathbf{v}_d$, can be constructed having the mathematical expression of the desired path [164]. For free form paths, the field construction can be carried out by determining the

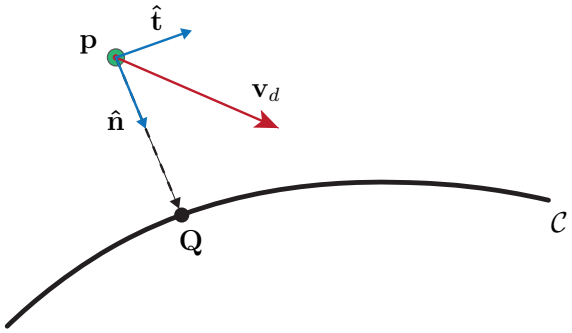


Figure 3.4: Velocity Field Construction. Illustration of the velocity field control field construction showing the end-effector point in space, $\mathbf{p}$, nearest point ($\mathbf{Q}$) on the desired path, $\mathcal{C}$, normal unit vector, $\hat{\mathbf{n}}$, tangential unit vector, $\hat{\mathbf{t}}$, and velocity field vector, $\mathbf{v}_d$.

normal vector to the path, and the tangent vector of the path, shown in Figure 3.4. This can be written as

$$\mathbf{v}_d(\mathbf{p}) = l(\alpha\hat{\mathbf{t}} + \|\mathbf{Q} - \mathbf{p}\|\hat{\mathbf{n}}) \tag{3.5}$$

where $l, \alpha \in \mathbb{R}^+$ are positive scaling constants, $\mathbf{p}$ denotes coordinates of current configuration, $\mathbf{Q}$ is the nearest point on the desired path $\mathcal{C}$ to $\mathbf{p}$, hence $\mathbf{Q}-\mathbf{p}$ signifies the distance vector, $\hat{\mathbf{n}} = (\mathbf{Q}-\mathbf{p})/\|\mathbf{Q}-\mathbf{p}\|$ is the normalised tangent vector, which can be determined by subtracting $\mathbf{Q}$ from the next point in the desired path and then normalising. To achieve AAN behaviour, controllers must be forgiving for small deviations from the desired velocity vector and this can be achieved in a number of ways. A common approach is to assign a permissible velocity range (velocity freedom zone), within which no corrective action is taken.

In [163] a velocity field control scheme was implemented in a fully backdrivable lower-limb Indego exoskeleton, intended to provide guidance and assistance to poorly ambulatory individuals during walking without unduly interfering with their ability to maintain balance. A viscous flow field acting on the lower limb joints is emulated. The controller includes only a proportional velocity feedback term and it is claimed to be sufficiently forgiving of movement interference. Results showed that the *flow* controller provides a combination of lower guidance error and lower disturbance to the user, relative to a force field-based controller.

In [165], the authors studied the performance of a precise velocity field controller on a poorly modelled lower-limb exoskeleton, utilising neural networks to compensate for modelling uncertainties. The authors adapted the approach in [166], combining velocity and force field control to develop a path tracking control scheme with minimised robotic intervention to gain active participation from impaired subjects while avoiding large position errors. A velocity field is constructed encoding the desired path, and a force field is considered around the path. A normal force term keeps the robot position arbitrarily close to the path. The combination of these ideas gains the AAN property with adequate freedom of timing: a key factor in reducing the robot intervention.

3.2.3 Offline Adaptive

There is a broad class of control algorithms that, instead of being manually parameterised for each subject and re-parameterised in the case of patient progress, adapt trial-by-trial to the subject's evolving needs. This is usually done using measures of performance and progress during trial execution, on which the parameter adjustment is based. The aim is for these algorithms to adapt autonomously to the changing subject needs, with respect to human-robot interaction, assistance level and joint compliance. Performance indexes vary but they are commonly based on a form of tracking error. A common method for implementing trial-by-trial adaptation is using a feedforward assistance term. This term can be iteratively adapted in order to gradually learn the specific requirements of the subject and improve tracking.

In [134], a PD impedance controller was used, also with a feedforward assistive term, which was adapted during motion based on the dynamics of the patient's extremity, neurological ability, and the patient's effort, modelled through a Gaussian RBF-based neural network. Unknown dynamics are assumed initially then AAN is achieved by adding a force decay term or *forgetting factor* to the adaptive control term which estimates the dynamics. This reduces output from the robot when errors in task execution are small.

This controller was tested with 11 chronic stroke patients. Results displayed the *slacking* behaviour of human motor control, as the central nervous system tried to minimise its efforts. As soon as it could take advantage of the controller's willingness to contribute effort to the task, it would let the controller take over most of the physical work. By adding the force decay term, the controller was able to consistently encourage patient effort.

In [44], the authors implemented iterative learning control: a feedforward assistive term was learnt using trial-by-trial adaptation, based on the error of previous trials and on a fuzzy logic-based nonlinear function of tracking error statistics. Its usability was demonstrated pre-clinic with six impaired subjects as shown in [167].

An AAN controller with adjustable assistance level was proposed in [147]. The authors provided a freedom zone for the user and included an adaptive neural network term to compensate for the uncertainties of the dynamic model used in the controller, in the interest of precise tracking. The term has a forgetting factor to reduce its contribution to the control output over time. In [160], the authors conducted a comparison of learned force field control and adaptive AAN control. The adaptive method estimates the coupled finger-exoskeleton system torque requirement of a subject using RBF networks, adapting RBF magnitudes in real-time to provide a feedforward assistance, improving trajectory tracking. The adaptive controller, unlike the pre-learnt force field controller, was able to adapt to the changing needs of the coupled finger-exoskeleton system. It was better at helping perform tasks with a consistent speed, however, was capable of applying excessive torque in the case of restricted motion, potentially making it less safe.

The authors in [168] developed adaptive inverse optimal hybrid controller, combining inverse optimal control with a neural network-based actor-critic feedforward signal, responsible for learning the uncertain nonlinear dynamics. A simulation and experiment with a rehabilitation robot demonstrated the effectiveness of the scheme.

A different approach for providing subject-specific adaptation of assistance is to modify the closed loop control parameters that dictate system feedback. This can be achieved by altering the feedback gains, which are representative of the stiffness and damping impedance control parameters.

In [169], a joint level adaptive PD controller regulated the stiffness of a robotic exoskeleton. Trial-by-trial adaptation of the feedback gains was driven by the tracking error of the reference trajectory. In [170], a sensorless force estimator, based on a Kalman filter, was used to determine the exoskeleton wearer's ability to perform the desired task, in order to provide the wearer with minimal assistance within user-specified allowable position error bounds. The controllers in both of these examples were tested on healthy subjects, demonstrating the applicability of both approaches. The researchers in [171], also proposed a novel subject-adaptive AAN control based on feedback gain modification for arm rehabilitation. The controller is capable of changing the amount of error allowed during movement execution, while estimating forces provided by the participant that contribute to movement execution. The closed-loop system was shown to be uniformly ultimately bounded which, the authors claim, leads to AAN performance. The controller was validated with a wrist exoskeleton in an experimental study involving five healthy subjects. The algorithm was shown to decrease its feedback control action when subjects shifted interaction behaviour from passive to active engagement.

In [172], an AAN control algorithm tried to anticipate patient deviations from desired reference trajectories, by using an adaptive model of the patient's impairment. The dysfunction profile of the patient can be provided by a therapist, depending on the particular exercise task being targeted and the recovery progress. The predictive quality of the controller was demonstrated in simulation.

The offline adaptive control algorithms considered here are susceptible to lag, and provide a delayed response which is at best appropriate for the previous trial. Offline adaptive approaches do not account for changing individual needs intra-trial. They do not consider that assistance/resistance requirements

can vary significantly for different points in an exercise trajectory and in different areas of the task space. Although trial-by-trial adaptation may be adequate for exercises with physical requirements which change on a relatively long time scale, the assumption of quasi-static assistance requirements is flawed for many cases of dynamic exercise.

3.2.4 Control Approach Discussion

A number of research groups have made progress towards solving AAN control for robotic assistance during active-assist exercise. However, measurement of human participation while performing robotic assistance or therapy is still an area which lacks an optimal solution. Purely model-based methods, such as nonlinear disturbance observers, are unable to distinguish external disturbance from human effort in many situations. AAN control approaches such as adaptive impedance control and velocity/force field control do not adapt assistance for any stage of recovery – they are designed for active-assist exercise only.

Impedance control is a common solution to adjust the level of assistance. Aside from compensation factors, the robot does not assist unless the patient deviates from the planned trajectory, prompting active patient participation. Researchers have developed performance-based progressive assistance approaches using impedance control, in which the patient can express unobstructed movement in an attempt to reach a target position [140]. The controllers modify the time to complete the movement and the amount of corrective assistance, challenging the subject to attempt to move faster and with less error.

Using force field control, no active adaptation takes place, and there is little control of the speed of execution of the task [160]. It is potentially safer, as it allows a subject to progress at their own speed without applying excessive force to correct positional error.

Adaptive control approaches are able to adjust in some ways to the changing needs of a human-robot system during robotic rehabilitation. Adaptation, in some cases, occurs trial-by-trial, and in others happens closer to real-time, as a high level algorithm is applied to adjust the desired impedance parameters based on rules using metrics of task performance or human effort.

AAN control approaches usually change the nature of robotic assistance based on either observed motion tracking error or human effort estimated using model-based methods, such as nonlinear disturbance observation. This approach estimates joint dynamics based on deflection without directly sensing forces at the joint. Both approaches have fundamental problems. Error-based adaptation suffers from latency as the controller can only adapt its policy after the fact, which in the case of dynamic exercise can be too slow to improve the nature of human-robot interaction on-the-fly. Model-based human effort estimation is inherently unable to distinguish human effort from external disturbance, making it suitable for a limited range of predictable environments and tasks.

The AAN control approaches presented in this chapter are not explicitly constrained. This means the safety and perceived comfort characteristics are indirectly dictated by the tuning of a limited set of parameters, such as stiffness and damping, and cannot be explicitly dictated or limited. The choice to explicitly limit system states, such as joint positions and velocities, could be very useful for therapists who want to tailor the control policies of robotic therapy devices for subjects with very heterogeneous health conditions and physical limitations.

Common problems identified in existing AAN control approaches are summarised in Table 3.1. The novel control framework proposed and evaluated in the remainder of this thesis seeks to address these issues and limitations.

Table 3.1: AAN Control Problems. Summary of commonly found and important problems identified within existing AAN control approaches.

A	**Adaptation Latency**	Existing controllers for robotic rehabilitation suffer adaptation latency because they react to past or present patient behaviour, not responding quickly to changes of the wearer's motion intention and rehabilitation requirements.
B	**Singular Modality**	Controllers are not designed to deliver assistance and adapt to requirements for all stages of recovery - they are designed solely for passive, active-assist, or active-resist training modalities.
C	**Unconstrained Behaviour**	It is not usually possible to place constraints on system states, or some linear or nonlinear function of states – useful for introducing personalised physical limitations during therapy, enforcing coordinated movement between joints, and protecting hardware.
D	**Disturbance Identification**	Sensorless model-based human torque estimation is sensitive to external disturbance and cannot distinguish it from human torque.
E	**Human Torque Estimation Latency**	Sensorless model-based human torque estimation introduces latency as it measures the effect of human torque after generation.
F	**Flawed Human Torque Estimation Assumptions**	Existing approaches for modelling human joint torque tend to be highly dependent on specific established biomechanical models and therefore assume healthy and conventional muscle behaviour.

Chapter Summary

In this chapter, examples of existing approaches for robotic control during rehabilitation and assistance were provided. The different human-robot exercise modes of passive, active-assist, and active-resist were defined as control approaches in this application area usually target one of these training modalities. The most prominent control approaches for active-assist exercise, which is the most interesting and challenging training modality from a control perspective, were then explored in detail with recent examples from literature. This included impedance control, velocity/force-field control and offline adaptive control. Then followed a discussion of the state of the art of AAN control approaches, concluding in a summary of the six most significant problems identified within the field.

Chapter 4

Model Predictive Control for Intelligent Robotic Assistance

The previous chapter provided a state of the art assessment for Assist-as-Needed (AAN) control approaches. Training modalities used for robotic rehabilitation were outlined, existing control approaches were summarised, and major problems with existing approaches were identified.

In this chapter, Model Predictive Control (MPC) is outlined as part of a novel AAN control strategy that harnesses the decision making power of fuzzy logic based on subject involvement information, gathered using a human torque estimation model. The chapter consists of four sections detailing the stages involved in development of the AAN control framework and simulation environment. The first section outlines the proposed AAN control framework, and establishes the aims of the novel control approach. The second section focuses on the human-exoskeleton system, and describes seated and standing exoskeleton configurations, and the models used for their application to exercise tasks. This section also describes the process of model parametric identification. In the third section, the choice of MPC tool and controller design is motivated, and details of the problem formulation and optimization algorithm are provided. The

controller setup and parameters are presented with an example problem. In the fourth section, the controller simulation environment is introduced, and a controller comparison study is carried out, which compares the proposed AAN approach to benchmark control methods with respect to important aspects of performance such as constrained assistance, step response, human torque response, trajectory tracking, modelling error, and advance adaptation.

4.1 Control Framework

This thesis presents a novel AAN control framework that addresses the key problems identified with existing AAN control approaches: lack of adaptation to changing physical requirements of the subject during rehabilitation exercise; latency in human effort estimation; adaptation latency, due to delay between perceived performance changes and controller parameter update; and unsafe controller behaviour caused by unconstrained joint velocity and range of motion during exercise. The problems outlined here, summarised in Table 3.1, are all considered in the design of the novel control framework, shown in Figure 4.1, which consists of three main components: 1) a model predictive controller, for model-based optimal control of highly nonlinear human-exoskeleton systems; 2) a Fuzzy Logic Algorithm (FLA), for determining the appropriate mode of assistance based on human behaviour and system state, and 3) a Human Torque Estimation (HTE) model, for predicting human torque from measured muscle activity signals and system states.

The control framework is designed for three modes of human-robot interaction, based on stages of the neurological rehabilitation process – passive mode, active-assist mode, and safety mode. The nature of human-robot interaction is preemptively adapted based on the subject's muscle activity, which is observed prior the corresponding joint motion. MPC provides optimal assistance, evaluated over a predic-

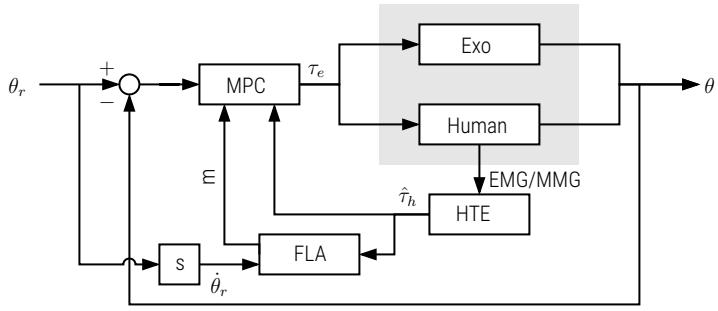


Figure 4.1: Proposed AAN-MPC Control Framework. Schematic showing control framework with Model Predictive Control (MPC), plant (Exo and Human), model for Human Torque Estimation (HTE) from muscle activity signals, and Fuzzy Logic Algorithm (FLA). θ and θ_r are the knee joint angle and reference joint angle, $\dot{\theta}_r$ is the reference joint velocity, τ_e and $\hat{\tau}_h$ are exoskeleton torque and estimated human torque, and m is the assistance mode variable.

tion horizon, while taking into account the intention predicted from muscle activity. A FLA determines the human-robot interaction mode according to the reference angular velocity and the predicted human joint torque from the HTE model. The likelihoods of the three human interaction modes are integrated with MPC by tuning the weighting of costs in the MPC objective function. MPC enables the controller to perform constrained optimisation, which can enforce subject-specific state limitations for safety and to protect hardware.

The aims of the proposed control framework, which specifically address the problems identified with existing AAN control approaches (Table 3.1), are summarised in Table 4.1. The remainder of this chapter explores elements of the control framework in detail and human-exoskeleton system models are developed for example exercise tasks.

4.2 Human-Exoskeleton System

The exercise tasks used for controller testing include seated and standing positions. Because of different physical requirements, in terms of how the device can be fixed to the environment and secured to the

Table 4.1: Control Framework Aims. Summary of the six main AAN-MPC control framework aims, which directly address six common problems identified within existing AAN control approaches from Table 3.1.

A	**Advance Adaptation**	Controller will adjust human-robot interaction based on future patient behaviour using predicted human torque.
B	**Assistance Modes**	Controller will detect the most suitable mode of assistance on-the-fly using a fuzzy logic-based algorithm and augment the nature of assistance by altering the MPC objective function.
C	**Constrained MPC**	Controller will perform constrained optimisation to find optimal assistance within state and input constraints.
D	**Human Torque in MPC**	Directly sensing human torque will distinguish it from external disturbance, meaning the two are not conflated.
E	**Predictive Human Torque Estimation**	Human torque will be predicted by sensing muscle activity directly - removing latency.
F	**Agnostic Human Torque Estimation Approach**	Novel approach to modelling human torque will be biomechanics and muscle condition agnostic, using machine learning to train a predictive model.

subject, specific exoskeleton test configurations were designed for each, as illustrated in Figure 4.2.

4.2.1 Actuator Module

To streamline development and testing, the same actuator was used for each exoskeleton configuration, which could be swapped in and out in a modular fashion. The exoskeleton designs focus on securely providing assistance to the knee joint during lower limb exercise.

Actuation requirements were determined based on previous studies of knee joint dynamics during rehabilitation exercise [173]–[175]. The selected actuator was compact and lightweight with suitable torque-speed characteristics. It could be easily secured to and removed from exoskeleton configurations, to make switching between exercises easier.

The actuator chosen to fit the modular exoskeleton configuration design specification was the HEBI X-Series Actuator, shown in Figure 4.3. The HEBI X-Series is a compact and powerful Series Elastic Actuator

(a) Seated exoskeleton. **(b)** Standing exoskeleton.

Figure 4.2: Exoskeleton Configurations. Schematic of the modular exoskeleton configurations showing (i) thigh module with rigid surface fixing, (ii) thigh module with braces, (iii) shank module with braces, and (iv) common actuator module.

(SEA) with built-in motor control and sensing capabilities. It is also easy and quick to secure and remove using socket screws, so can be used interchangeably in different configurations.

The actuator can simultaneously control and/or measure joint position, velocity, and torque. It has integrated three axis inertial measurement, enabling the reading of orientation. It can be controlled using Application Programming Interface (API) in MATLAB, Python, and C/C++. Low level language integration enables system states such as position, velocity, and torque to be read at joint level in real-time. Feedback can be requested from the actuator at rates of up to 1 kHz.

The device integrates a brushless DC motor, geartrain, springs, encoders, and control electronics in a compact package which communicates using standard 10/100 Mbps Ethernet. The device connects to a control PC using a standard network router. It can be powered using any voltage from 24 V - 48 V, however for the experiments in this thesis a constant 30 V is used (the maximum voltage of the power supply). The actuator has a peak torque of $\pm$20 Nm, maximum speed of 30 rpm, and total weight of 475 g. Detailed technical specifications for the actuator can be found in Appendix A.

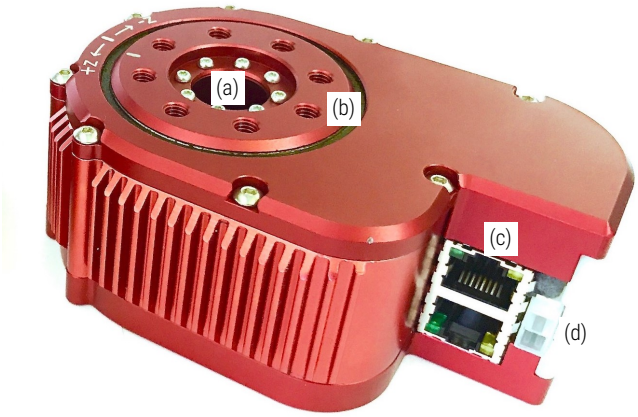


Figure 4.3: HEBI X-Series Actuator. Photograph of HEBI X-Series actuator showing (a) thru bore, (b) motor output, (c) dual Ethernet ports, and (d) Molex power connector.

4.2.2 Seated Configuration

The seated exoskeleton configuration, shown in Figure 4.4, is used for the dynamic extension and flexion exercise. The configuration has a surface-thigh module, a shank module, and the actuator module. Hex socket screws attach the body segment modules either side of the actuator module to make the full exoskeleton. The surface-thigh module is a rectangular structure built using rigid aluminium tubing and support struts. It is securely fixed to the surface using a widely adjustable F-clamp. The module attaches to the flat base of the actuator via an adjustable mounting plate (using the fittings shown in Appendix A). The shank module has a 3D printed shaft which attaches to the motor output using hex socket screws. Two 3D printed and padded braces on the shank segment secure the subject to the exoskeleton using comfortable fabric straps, which are easy to secure and release.

The configuration is designed to ensure subject comfort, safety, and ease of use. For seated knee joint exercise it is necessary for the exoskeleton to attach securely to the shank of the subject. As the thigh segment is naturally stationary during sitting, it is not necessary for the exoskeleton to be secured to the thigh. The exoskeleton is instead fixed to the sitting surface. Any surface can be used, providing

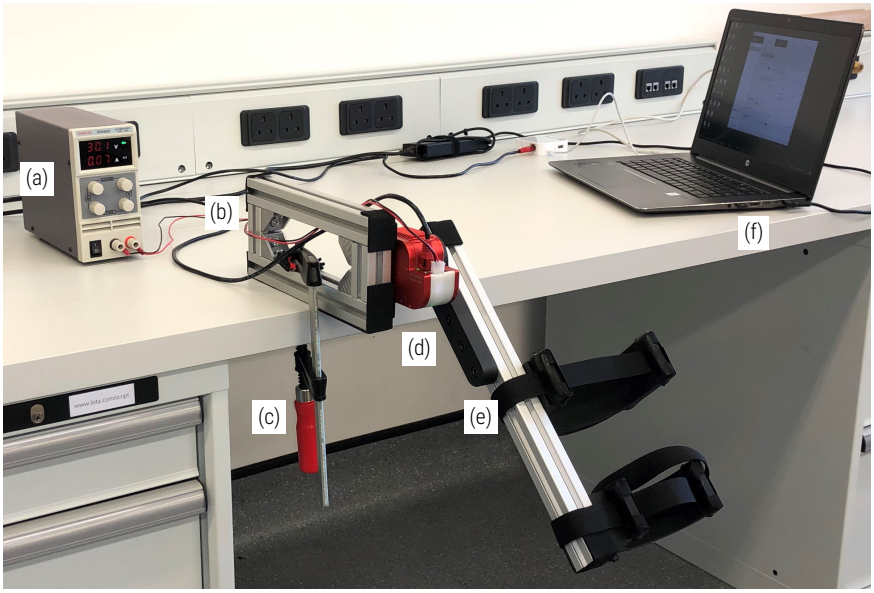


Figure 4.4: Seated Exoskeleton Configuration. Seated exoskeleton configuration with (a) power supply, (b) surface-thigh module, (c) F-clamp, (d) actuator module, (e) shank module, and (f) control computer.

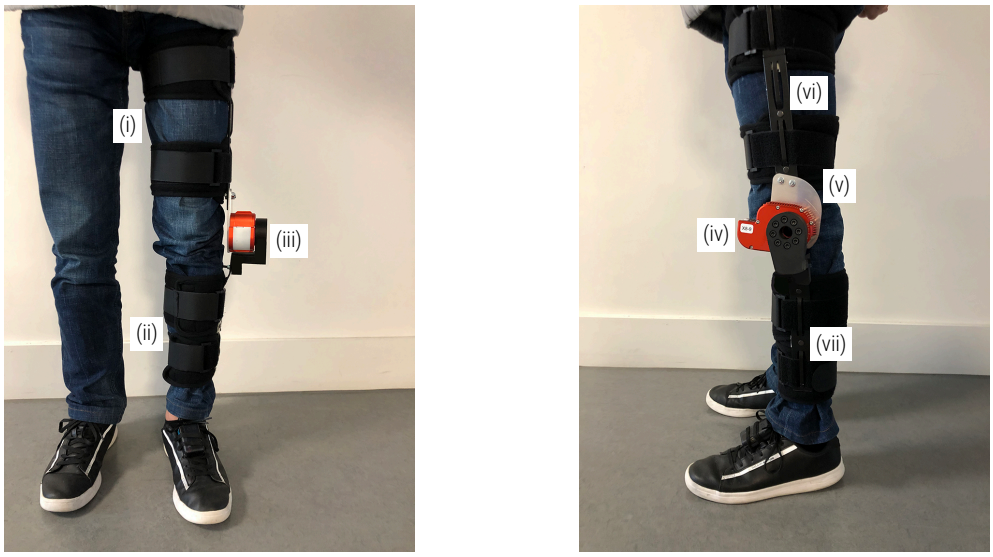
the subject can comfortably mount and dismount, and can extend and flex their leg without obstruction. This configuration makes the structure more secure and unlikely to sustain any accidental damage from falling or over-extension; it is easier to put on and take off and more comfortable for the subject to use, with fewer tight braces on the leg; and it frees up space around the subject's thigh area for placement of muscle activity sensors. Foam padding is applied on both of the shank segment braces and all over the exposed surfaces of the exoskeleton, making the device safe and comfortable. The configuration is very portable and can be attached to almost any rigid over-hanging surface, such as a table or desk. The design is very flexible and it is possible to independently adjust: shank brace height, depth, and rotation; shank segment length; and horizontal and vertical knee joint position. A technical drawing of the seated exoskeleton configuration can be found in Appendix A.

4.2.3 Standing Configuration

The standing exoskeleton configuration, shown in Figure 4.5, is designed for assisting subjects with exercise that involves standing and requires support of both the shank and thigh segments, such as sit-to-stand or overground walking. The configuration consists of a thigh module, a shank module, and the actuator module. As with the seated configuration, the segment modules attach to the motor module by hex socket screws to make the full exoskeleton assembly. This configuration does not attach to a rigid surface, but secures to the subject's thigh and shank segments using padded fabric braces. The fabric braces are fixed to aluminium thigh and shank supports by threading through raised slots. The position of the braces can easily be adjusted along a hook and loop strip on the inner side of the supports for subject comfort and a secure fit. The thigh support is joined by hex socket screws to an aluminium fixed bracket which acts as a mounting plate, securing the flat base of the actuator module. The actuator output is joined to a 3D printed rotating bracket which is joined to the shank support.

The configuration is lightweight and comfortable, feeling relatively transparent to the user. Lightweight materials were selected to reduce overall weight and a lean design with support along the outer leg was chosen. The device could be more secure and rigid with inner and outer supports and solid material brackets, however the lightweight option with fabric braces gives enhanced subject comfort and device transparency.

Muscle activity sensors can be connected while using the standing exoskeleton configuration, however care must be taken over sensor placement in relation to the thigh and shank segment braces. Brace positions can be adjusted to accommodate muscle activity sensors, so the intended sensor placement should be considered before adjusting the brace positions. If the brace positions interferes with the muscle activity sensors and cause physical disturbance during exercise, this can add noise to signals and reduce quality so should be avoided. Both modalities of muscle activity sensor in this thesis (Electromyography



(a) Front view. **(b)** Side view.

Figure 4.5: Standing Exoskeleton Configuration. Subject wearing (disconnected) standing exoskeleton configuration with (i) thigh braces, (ii) shank braces, (iii) rotating bracket, (iv) motor module, (v) fixed bracket, (vi) outer thigh support, and (vii) outer shank support. A short video demonstration of the standing exoskeleton configuration can be found here.

(EMG) and Mechanomyography (MMG)), are physically connected to their amplification units, limiting configuration mobility. However, it is possible to move within a radius of 3-4 meters while using the sensors in a wired configuration, and the TMSi EMG amplifier can use Bluetooth communication, which extends the range of mobility.

The configuration must be connected to the network router and powered at all times, presenting mobility limitations while using a conventional power supply and wired router for communication. A portable battery pack and wireless router can be used to achieve independent mobility for testing. In this case mobility is limited by battery capacity and range of network connection between the control computer and wireless router.

4.2.4 Human-Exoskeleton System Models

At the core of the model-based predictive control approach is the human-exoskeleton system model. It is from the system model that the dynamics are derived and integrated to give the predicted future system states to optimise. The system model is also used to simulate controller performance, with the introduction of some modelling errors, before testing with a real exoskeleton and subject. It is important to have a model that is representative of the real world system, with reasonable assumptions made in the process of simplification and reduction of model complexity.

The modelling approach begins with a general model for a planar lower limb human-exoskeleton system with multiple DOFs. Then the model is simplified based on the physical constraints of a particular exercise task. According to Lagrange formulation, a generic dynamic model of the human lower limb coupled to a knee joint exoskeleton during daily living activities can be written as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{\tau}_f = \boldsymbol{\tau} + \boldsymbol{J}^T\boldsymbol{F} \tag{4.1}$$

where $\boldsymbol{q}$ represents the state vector of joint angles, $\boldsymbol{M}(\boldsymbol{q})$ is the inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ represents the Coriolis and centrifugal forces, and $\boldsymbol{G}(\boldsymbol{q})$ is the gravitational torque. $\boldsymbol{\tau}_f$ represents the friction torque, and $\boldsymbol{F}$ denotes the external force acting on the system, e.g. ground reaction. $\boldsymbol{J}$ is the Jacobian matrix. The input vector $\boldsymbol{\tau}$ is made up of the human joint torque $\boldsymbol{\tau}_h$, and the generated exoskeleton torque $\boldsymbol{\tau}_e$ such that $\boldsymbol{\tau} = \boldsymbol{\tau}_h + \boldsymbol{\tau}_e$.

For the experiments in this work, knee joint exoskeletons are used to intelligently assist extension and flexion movements during two exercises: seated leg extension and flexion, and walking. Robotic assistance of the knee joint for these exercises can be described by simplifying the formulation in (4.1) to a double

pendulum with knee joint dynamics. In this case it can be rewritten as follows

$$J\ddot{\theta} + B(\dot{\theta} - \dot{\theta}_t) + A\text{sgn}(\dot{\theta} - \dot{\theta}_t) + \tau_g \sin\theta + \tau_l = \tau_e + \tau_h \tag{4.2}$$

where θ and θ_t are the knee joint angle and inclination angle of the reference limb (thigh) in the sagittal plane respectively, $J = J_h + J_e$ is the inertia of the human limb and exoskeleton. A is the solid friction parameter of the exoskeleton, $B = B_h + B_e$ is the viscous friction of the human knee joint and exoskeleton, $\tau_g = \tau_{g,h} + \tau_{g,e}$ is the gravitational torque of the human limb and exoskeleton, τ_e and τ_h are the torques generated by the exoskeleton and human muscles at the knee joint, respectively. $\text{sgn}(\cdot)$ is the sign function, and τ_l is the lumped torque caused by unaccounted for body segments and external forces.

Note that the sign function can be approximated using a steep $\arctan(\cdot)$ function, making it continuous and differentiable for calculation of gradients for simulation and control. Also note that movement of the exoskeleton and wearer's thigh and shank segments are considered to be synchronous and simultaneous (i.e. the human and exoskeleton are rigidly coupled).

The model described in (4.2), although relatively simple, can be applied to knee joint assistance during the exercises of seated leg extension and flexion, and walking. The assumptions and simplifications used to apply the general model to each of the exercises are described in Sections 4.2.4.1 and 4.2.4.2.

4.2.4.1 Seated Model: Leg Extension and Flexion

For robotic assistance during the seated leg extension and flexion exercise, it is assumed that the thigh segment is rigidly fixed to the sitting surface and that body segment forces are fully supported. Therefore

$\tau_l = 0$ and $\dot{\theta}_t = 0$. So the model in (4.2) can be rewritten as

$$J\ddot{\theta} + B\dot{\theta} + A\text{sgn}\dot{\theta} + \tau_g \sin\theta = \tau_e + \tau_h. \tag{4.3}$$

The convention used for joint angle origin and direction is illustrated in Fig. 4.6, which also shows the gravitational torque vector and exoskeleton joint torque.

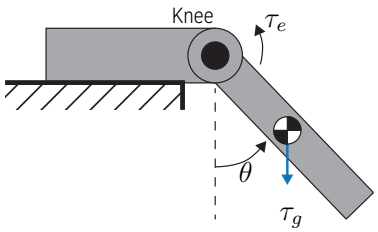


Figure 4.6: Seated Exoskeleton Schematic. Diagram showing the exoskeleton system in the sitting position with θ, joint angle, τ_e exoskeleton torque, and τ_g, gravitational torque of the shank/foot segment.

4.2.4.2 Standing Model: Walking

Using the model (4.2) for robotic assistance of the knee joint during walking, the human limb model varies with respect to the swing phase and stance phase of gait motion. In this work, assistance is only provided during the swing phase so lumped torque, τ_l, is estimated for swing phase only. The model can also be used to derive optimal assistance during the stance phase, but that case is not within the scope of this work. The model for evaluating knee joint dynamics during the swing phase of gait is illustrated in Figure 4.7.

During the swing phase, movement at the hip joint causes the thigh segment inclination, θ_t, to vary. This inclination is calculated in the sagittal plane using a measured orientation quaternion from the HEBI X-Series actuator module. The module is rigidly fixed to the thigh segment using braces, as shown in Figure 4.5b. Dynamics of the knee joint are affected by the movement of the hip joint, causing lumped body

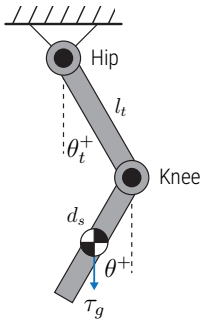


Figure 4.7: Standing Exoskeleton Schematic for Swing Phase. Schematic showing standing exoskeleton configuration in the walking position during the swing phase of gait with joint angle, θ, reference limb (thigh) angle, θ_t, thigh segment length, l_t, and gravitational torque of the shank/foot segment, τ_g.

segment torque which can be expressed as

$$\tau_l = m_s d_s l_t(\ddot{\theta}_t \cos(\theta_t - \theta) - \dot{\theta}_t^2 \sin(\theta_t - \theta)) \tag{4.4}$$

where m_s is the lumped mass of the human shank and the corresponding exoskeleton segment. l_t is the length of the human thigh, and d_s is the length of the human shank part from the knee joint centre to the centre of mass of the shank and foot. θ_t is the hip joint angle [176], as illustrated in Figure 4.7.

The lumped torque, τ_l, can be fully compensated by the exoskeleton according to the optimal control problem formulation. Apart from the model shown in 4.2, the human body model can change with respect to different ground surfaces and contact conditions. For the testing in this thesis, only the condition of walking over level ground is considered.

4.2.4.3 Parametric Identification

Parameters for the human-exoskeleton system models (4.2, 4.3 and 4.4) require identification. The parameters can be identified in three steps.

Table 4.2: Seated Exercise Model Parameters. Seated human subject and exoskeleton system parameters identified using the method outlined in Section 4.2.4.3.

System	J $(Kg \cdot m^2)$	B $(Nm \cdot s \cdot rad^{-1})$	A (Nm)	τ_g (Nm)
Exo	0.0377	0.0207	0.0000	1.7536
S_1	0.4315	0.1676	0.0000	14.256
S_2	0.1927	0.1534	0.0000	7.5008
S_3	0.3060	0.1575	0.0000	10.595

Exoskeleton Parameters For this step, the exoskeleton is driven in the seated configuration, without a human coupled to the system. Sinusoidal reference trajectories, θ_r, with five different frequencies (0.1, 0.2, 0.25, 0.5 and 1.0 Hz) were chosen. These values were chosen to cover the range of frequencies present in walking gait, with greater concentration around more commonly occurring lower frequencies [177]. The exoskeleton is controlled using a basic proportional controller, i.e. $\tau_e = K(\theta_r - \theta)$, while the knee joint angle θ is recorded. The joint angular velocity, $\dot{\theta}$, and joint acceleration, $\ddot{\theta}$, are calculated by first and second derivatives of the joint angle low-pass filtered with a cut-off frequency of 5 Hz. From the system model (4.3), it is clear that θ, $\dot{\theta}$, $\ddot{\theta}$, and τ_e are known, while the parameters J_e, B_e, A_e, and $\tau_{g,e}$ need to be identified. Given there are the same number of unknown variables as known variables, a linear least-squares optimisation algorithm is used to identify the unknown model parameters [178]. Estimated parameter values for the seated exoskeleton configuration are shown in Table 4.2.

Human Limb Parameters The human limb parameters, J_h, B_h, A_h, and $\tau_{g,h}$, can be inferred by superposition once human-exoskeleton parameters and exoskeleton parameters are determined experimentally. Human-exoskeleton system parameters are identified using a similar approach to the exoskeleton parameters, but with human coupled the the exoskeleton. The human subject is instructed to remain relaxed, so human torque can be neglected ($\tau_h = 0$). EMG signals are monitored here to ensure the subject is not exerting any voluntary effort. As with the exoskeleton parameter identification, the system is proportionally controlled with the same reference trajectories and the unknown human-exoskeleton parameters are identified using linear least-squares. A sample of cross-validation results of human-exoskeleton iden-

tified parameters for one seated exoskeleton trajectory frequency can be seen in Figure 4.8. Satisfactory agreement between input torque, τ_e, and estimated torque, $\hat{\tau}_e$, generated from identified parameters is observed (average Root Mean Square Error (RMSE) = 1.47 Nm across all subjects). The estimated human limb parameter values for all subjects are shown in Table 4.2.

Measured Parameters The other parameters of the human-exoskeleton system models can be determined by measuring physical properties of the subject's body segments. For example, mass, centre of mass, and length of the body segments are determined by physically measuring the properties. Although it is difficult to precisely measure the mass and centre of mass of an individual body segment, they can be estimated from a proportion of a human's overall weight and height, using guidelines in the literature [179]. The following parameters were calculated for Subject 1, a male of height 1.88 m and weight 85 Kg: m_s = 3.723 Kg, d_s = 225.25 mm, and l_t = 438.22 mm.

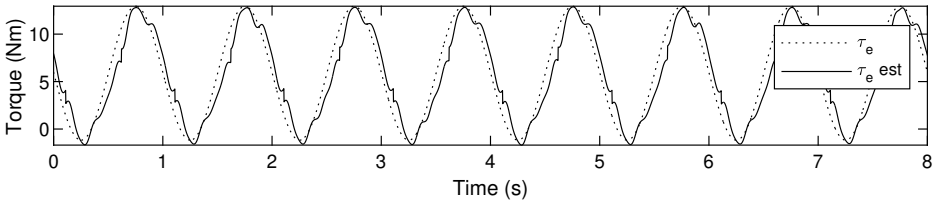


Figure 4.8: Cross-Validation Human-Exoskeleton Torque. Cross-validation results of the human-exoskeleton identified parameters at 1.0 Hz showing high agreement between input torque and estimated torque from the identified seated human-exoskeleton model parameters.

4.3 Model Predictive Controller

Central to the AAN-MPC control approach proposed in this work is the model predictive controller. It is an important component for effectively addressing the latency problem present in existing adaptive AAN control approaches. In this section, background around MPC for robotic assistance is presented, development of the controller is explored, and a simple application example is provided.

4.3.1 Background

MPC has been investigated for AAN control for robotic rehabilitation previously. Ozen et al. applied nonlinear MPC to an upper body rehabilitation virtual task with a delta robot, optimising for training task completion, motor learning, and trainee skill level simultaneously [180]. This approach is promising for neurorehabilitation, however it does not consider human joint behaviour directly. Raza et al. explored using MPC to control a simulated upper limb rehabilitation robot under disturbance conditions. Task precision improved however the approach was unable to distinguish external disturbance from human effort [181]. Teramae et al. applied MPC to a 1-DOF upper limb rehabilitation task by considering an upper-limb exoskeleton model and using muscle activity for human joint torque estimation [182]. The method was able to derive optimal robot torque for completing the task, but did not adapt the objective function for the different assistance modes required during rehabilitation training, particularly safety mode in the case of human resistance to the robot.

4.3.2 Controller Specification

A well-known limitation of MPC is the numerical effort required to solve the underlying optimal control problem. Recently, however, the development of MPC solvers for linear and nonlinear systems has matured to the point where MPC can often be applied in real-time. Having chosen MPC for the AAN control approach in this thesis, a suitable controller development workflow and software package was required. To meet the technical specifications of the problem, it needed to:

- Solve for nonlinear dynamical systems
- Handle state and input constraints

- Be capable of real-time solution, with sampling time in the millisecond range
- Be portable for efficient implementation on hardware, ideally written in C or C++ for integration with existing equipment.

Based on these requirements, a suitable controller development tool was found. The open-source GRAMPC software package was chosen. It specialises in nonlinear continuous-time systems subject to nonlinear state and control constraints. It uses an underlying augmented Lagrangian formulation with a tailored gradient method for the inner minimization problem. This tool, implemented in C code, enables real-time solution of nonlinear highly dynamical systems with sampling times in the (sub)millisecond range [183].

4.3.3 Problem Formulation

The GRAMPC tool can solve nonlinear constrained model predictive control problems with fixed end time. The method uses iterative solution of an Optimal Control Problem (OCP) of the form

$$\min_{\boldsymbol{u}} \quad J(\boldsymbol{u};\boldsymbol{x}_k) = V(\boldsymbol{x}(T)) + \int_0^T l(\boldsymbol{x}(s),\boldsymbol{u}(s),s)\mathrm{d}s \tag{4.5a}$$

$$s.t. \quad \boldsymbol{M}\dot{\boldsymbol{x}}(s) = \boldsymbol{f}(\boldsymbol{x}(s),\boldsymbol{u}(s),t_k+s), \quad \boldsymbol{x}(0) = \boldsymbol{x}_k \tag{4.5b}$$

$$\boldsymbol{h}(\boldsymbol{x}(s),\boldsymbol{u}(s),s) \leq \boldsymbol{0}, \quad \boldsymbol{h}_T(\boldsymbol{x}(T),T) \leq \boldsymbol{0} \tag{4.5c}$$

$$\boldsymbol{x}(s) \in [\boldsymbol{x}_{\min},\boldsymbol{x}_{\max}], \quad \boldsymbol{x}(T) \in \Omega_\beta \tag{4.5d}$$

$$\boldsymbol{u}(s) \in [\boldsymbol{u}_{\min},\boldsymbol{u}_{\max}] \tag{4.5e}$$

with state $\boldsymbol{x} \in \mathbb{R}^{N_x}$, control $\boldsymbol{u} \in \mathbb{R}^{N_u}$, and end time $T \in \mathbb{R}$. The cost to be minimized (4.5a) consists of terminal and integral cost functions $V : \mathbb{R}^{N_x} \times \mathbb{R} \to \mathbb{R}$ and $l : \mathbb{R}^{N_x} \times \mathbb{R}^{N_u} \times \mathbb{R} \to \mathbb{R}$, respectively. N_x and N_u are number of system states and control inputs respectively. The dynamics (4.5b) are shown in

semi-implicit form with the mass matrix $\boldsymbol{M}$, the nonlinear system function $f : \mathbb{R}^{N_x} \times \mathbb{R}^{N_u} \times \mathbb{R} \to \mathbb{R}^{N_x}$, and the initial state $\boldsymbol{x}_0$.

Equations (4.5c) account for inequality and corresponding terminal constraints $\boldsymbol{h} : \mathbb{R}^{N_x} \times \mathbb{R}^{N_u} \times \mathbb{R} \to \mathbb{R}^{\mathbb{N}_h}$, and $\boldsymbol{h}_T : \mathbb{R}^{N_x} \times \mathbb{R} \to \mathbb{R}^{N_{\boldsymbol{h}_T}}$. Finally, (4.5d) and (4.5e) represent box constraints for the system state and optimisation variables.

The MPC-internal time coordinate $s \in [0, T]$ is distributed over the prediction horizon T. The initial state $\boldsymbol{x}_k$ is the measured or estimated system state at current sample time $t_k = t_0 + k\Delta t$, $k \in \mathbb{N}$ with $0 < \Delta t \leq T$ [183]. The first part of the computed control trajectory $\boldsymbol{u}(s), s \in [0, \Delta t)$ is used as control input for the actual plant over time interval $t \in [t_k, t_{k+1})$, before OCP (4.5) is solved again with new initial state $\boldsymbol{x}_{k+1}$ [183]. The working principle of discrete MPC is illustrated in Figure 4.9.

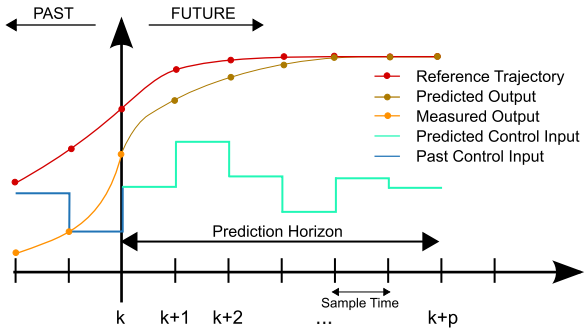


Figure 4.9: Discrete MPC Scheme. Illustration of the basic working principle for a discrete MPC scheme showing reference trajectory, predicted output (states), measured output (states), predicted control input, and past control input. Original drawing by Martin Behrendt, CC BY-SA 3.0.

A popular choice of the cost function (4.5a) is the quadratic form

$$V(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{x}_{\text{des}}\|^2_{\boldsymbol{P}}, \quad l(\boldsymbol{x}, \boldsymbol{u}) = \|\boldsymbol{x} - \boldsymbol{x}_{\text{des}}\|^2_{\boldsymbol{Q}} + \|\boldsymbol{u} - \boldsymbol{u}_{\text{des}}\|^2_{\boldsymbol{R}} \tag{4.6}$$

with the desired setpoint ($\boldsymbol{x}_{\text{des}}$, $\boldsymbol{u}_{\text{des}}$) and the positive (semi-)definite matrices $\boldsymbol{P}$, $\boldsymbol{Q}$, and $\boldsymbol{R}$ representing the weights. The vector-valued L^2-norm is defined by $\|\boldsymbol{x}\|_{L_2} = (\sum_{i=1}^{n} \|x_i\|_{L_2})^{\frac{1}{2}}$ with $\|x_i\|_{L_2} = (\int_0^T x_i^2(t)\mathrm{d}t)^{\frac{1}{2}}$.

Cost function formulation is problem-specific and an example for a 1-DOF lower limb exoskeleton is provided in Section 4.3.6. Stability can be ensured in the MPC by imposing a terminal constraint on $\boldsymbol{x}(T)$ as in (4.5d), where the set $\Omega_\beta = \{\boldsymbol{x} \in \mathbb{R}^{N_x} | V(\boldsymbol{x}) \leq \beta\}$ for some $\beta > 0$ is defined in terms of the terminal cost $V(\boldsymbol{x})$ that can be computed from solving a Lyapunov equation that renders the set Ω_β invariant under a local feedback law [184].

The terminal region and constraints on the state and control input as given in (4.5c) can be expressed as

$$\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} \boldsymbol{x} - \boldsymbol{x}_{\text{max}} \\ \boldsymbol{x}_{\text{min}} - \boldsymbol{x} \\ \boldsymbol{u} - \boldsymbol{u}_{\text{max}} \\ \boldsymbol{u}_{\text{min}} - \boldsymbol{u} \end{bmatrix} \leq \mathbf{0}, \quad \boldsymbol{h}_T(\boldsymbol{x}) = V(\boldsymbol{x}) - \beta \leq \mathbf{0}. \tag{4.7}$$

Note that terminal constraints can be omitted in real-time MPC, in order to reduce computational burden [185]. Real-time feasibility can also be achieved by limiting the number of iterations of the optimisation algorithm per sampling step and using the current solution for warm starting the next MPC step, reducing the sub-optimality over the runtime of the controller [186]. The optimisation algorithm used for solving the OCP is described in Section 4.3.4.

4.3.4 Optimisation Algorithm

The optimisation algorithm in the GRAMPC tool uses an augmented Lagrangian formulation in combination with a real-time projected gradient method. The framework rapidly provides a (sub)optimal solution at low computational cost, which is important for feasibility with real-time applications. The augmented Lagrangian formulation and corresponding optimisation algorithm are summarised here for solving the OCP, with more information available from Englert et al. [183].

The augmented Lagrangian formulation adjoins the constraints (4.5c) to the cost function (4.5a) by means

of multipliers $\bar{\boldsymbol{\mu}} = (\boldsymbol{\mu}_{\boldsymbol{h}}, \boldsymbol{\mu}_{\boldsymbol{h}_T})$ and additional quadratic penalty terms with penalty parameters $\bar{\boldsymbol{c}} = (\boldsymbol{c}_{\boldsymbol{h}}, \boldsymbol{c}_{\boldsymbol{h}_T})$. The inequalities are transformed into equality constraints by means of slack variables, which can be analytically solved for [187]

$$\bar{\boldsymbol{J}}(\boldsymbol{u}, \bar{\boldsymbol{\mu}}, \bar{\boldsymbol{c}}; \boldsymbol{x}_0) = \bar{V}(\boldsymbol{x}, \boldsymbol{\mu}_T, \boldsymbol{c}_T) + \int_0^T \bar{l}(\boldsymbol{x}, \boldsymbol{u}, t, \boldsymbol{\mu}, \boldsymbol{c})\mathrm{d}t. \tag{4.8}$$

The augmented cost function (4.9a) allows one to formulate the max-min-problem

$$\max_{\bar{\boldsymbol{\mu}}} \min_{\boldsymbol{u}} \quad \bar{\boldsymbol{J}}(\boldsymbol{u}, \bar{\boldsymbol{\mu}}, \bar{\boldsymbol{c}}; \boldsymbol{x}_0) \tag{4.9a}$$

$$s.t. \quad \boldsymbol{M}\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{4.9b}$$

$$\boldsymbol{u}(t) \in [\boldsymbol{u}_{\min}, \boldsymbol{u}_{\max}], \quad t \in [0, T]. \tag{4.9c}$$

The iteration structure of the augmented Lagrangian algorithm is summarised in Algorithm 1. In the current augmented Lagrangian iteration, i, the inner minimization is carried out by solving the OCP in (4.9) for the current set of multipliers $\bar{\boldsymbol{\mu}}^i$ and penalties $\bar{\boldsymbol{c}}^i$. Since the only remaining contraints within (4.9) are box constraints on the optimisation variable (4.9c), the problem can be solved using a projected gradient method. The solution of the inner minimization step consists of the control vector $\boldsymbol{u}^i$. The subsequent convergence check assesses the constraint violation as well as convergence behaviour of the previous minimization step. If convergence has not been reached, the multipliers and penalties are updated for the next iteration of the algorithm. Further details of each algorithm step are provided by Englert et al. [183].

Algorithm 1: Augmented Lagrangian algorithm

Initialisation
- Initialise multipliers $\bar{\boldsymbol{\mu}}^0$ and penalties $\bar{\boldsymbol{c}}^0$
- Set tolerances for convergence criterion

---

for $i = 1$ **to** i_{max} **do**
- Compute input $\boldsymbol{u}^i = \boldsymbol{u}^i(t)$ and state $\boldsymbol{x}^i = \boldsymbol{x}^i(t)$ by solving the optimal control problem in (4.9) using a projected gradient method
- If convergence criterion is reached, **break**
- Update multipliers $\bar{\boldsymbol{\mu}}^i = (\boldsymbol{\mu}_{\boldsymbol{h}}^i, \boldsymbol{\mu}_{\boldsymbol{h}_T}^i)$
- Update penalties $\bar{\boldsymbol{c}}^i = (\boldsymbol{c}_{\boldsymbol{h}}^i, \boldsymbol{c}_{\boldsymbol{h}_T}^i)$

end

4.3.5 GRAMPC Structure

The GRAMPC tool enables convenient prototyping, simulation, and real-time implementation of the controller. It is designed to be portable and executable on different operating systems and hardware without using external libraries. The core of the code is implemented in plain C with interfaces to C++ and MATLAB/Simulink.

Figure 4.10 shows the structure of GRAMPC and indicates the steps involved in compiling an executable project. Firstly, the problem must be defined, using C function templates. A problem formulation example is detailed in Section 4.3.6. Problem-specific parameters and algorithmic options are set regarding the MPC objective function, numerical integration in the gradient algorithm, line search strategy, and other features.

A specific problem can be parameterised and solved with C/C++ or Matlab/Simulink. This enables convenient controller design using the Matlab environment, simulation and testing of controller performance using Simulink, and real-time deployment with robotic equipment in C/C++. The real-time workspace of the project, algorithmic options and problem-specific parameters are stored by the structure variable `grampc`, which is easy to access for the problem at hand in all environments [183].

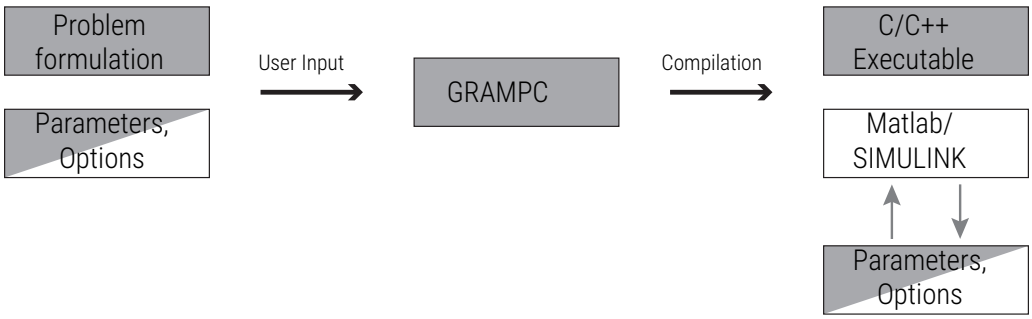


Figure 4.10: GRAMPC Structure. Structure of the GRAMPC toolkit, from user input through to compilation for real-time controller implementation (grey: C/C++ code, white: Matlab environment).

4.3.6 Application Example

The problem formulation steps for the GRAMPC tool are illustrated in the following example for a seated 1-DOF lower limb exoskeleton without human. The dynamics are derived from a nonlinear Lagrange formulation (4.1) and simplified, ignoring human factors and any lumped torque caused by body or external forces. The model can be written as

$$J\ddot{\theta} + B\dot{\theta} + A\text{sgn}\dot{\theta} + \tau_g \sin\theta = \tau_e \tag{4.10}$$

where θ is the knee joint angle, J is the inertia of the exoskeleton, B is the viscous friction parameter, A is the solid friction parameter, τ_g is the gravitational torque, and τ_e is the exoskeleton torque. The knee joint angle origin and direction, gravitational torque vector, and exoskeleton joint torque are illustrated in Figure 4.6. The problem formulation can be written as

$$\min_{u(\cdot)} \quad J(\boldsymbol{u};\boldsymbol{x}_k) = \frac{1}{2}\boldsymbol{\Delta x}^\top(T)\boldsymbol{P}\boldsymbol{\Delta x}(T) + \frac{1}{2}\int_0^T \boldsymbol{\Delta x}^\top(s)\boldsymbol{Q}\boldsymbol{\Delta x}(s) + R\Delta u^2 \mathrm{d}s \tag{4.11a}$$

$$s.t. \quad \begin{bmatrix}\dot{x_1}\\ \dot{x_2}\end{bmatrix} = \begin{bmatrix}0 & 1\\ \frac{-\tau_g \sin x_1}{Jx_1} & \frac{-A\text{sgn}(x_2)}{Jx_2}\frac{-B}{J}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{J}\end{bmatrix}u, \quad \begin{bmatrix}x_1(0)\\ x_2(0)\end{bmatrix} = \begin{bmatrix}x_{k,1}\\ x_{k,2}\end{bmatrix} \tag{4.11b}$$

$$\begin{bmatrix}0\\ -2\end{bmatrix} \leq \begin{bmatrix}x_1\\ x_2\end{bmatrix} \leq \begin{bmatrix}1.4\\ 2\end{bmatrix}, \quad |u| \leq 20 \tag{4.11c}$$

with $x_1 = \theta$, $x_2 = \dot{\theta}$, $\boldsymbol{\Delta x} = \boldsymbol{x} - \boldsymbol{x}_{\text{des}}$, $\Delta u = u - u_{\text{des}}$, $\boldsymbol{P} = \boldsymbol{Q} = \begin{bmatrix} w_{x_1} & 0 \\ 0 & w_{x_2} \end{bmatrix}$, and $R = w_u$. The prediction horizon and sampling time are set to $T = 0.2$ s and $\Delta t = 2$ ms, respectively.

The problem formulation is set in the function template code `probfct.c`. The following listing shows the function structure and problem implementation.

```
/* Problem dimensions */
void ocp_dim(int *Nx, int *Nu, int *Nh, int *NhT)
{
        *Nx = 2;
        *Nu = 1;
        *Nh = 4;
        *NhT = 0;
}

/* Right-hand side of dynamics M dx/dt = f(t,x,u,userparam) */
void ffct(double *out, double t, double *x, double *u, typeUSERPARAM *userparam)
{
        double* pSys = (double*)userparam;
        double A = pSys[0];
        double B = pSys[1];
        double J = pSys[2];
        double tau_g = pSys[3];
        out[0] = x[1];
        out[1] = (u[0] + x[2] - A * tanh(10 * x[1]) -  B * x[1] - tau_g * sin(x
            [0])) / J;
}

/* Integral cost function int l(t,x(t),u(t),xdes,udes,userparam) */
void lfct(double *out, double t, double *x, double *u, double *xdes, double *
    udes, typeUSERPARAM *userparam)
{
        double *pCost = (double*)userparam;
        double w_theta = pCost[4];
        double w_tau = pCost[5];
        out[0] = w_theta * (x[0] - xdes[0]) * (x[0] - xdes[0]) + w_tau * (u[0] -
             udes[0]) * (u[0] - udes[0]);
}

/* Terminal cost function V(T,x(T),xdes,userparam) */
void Vfct(double *out, double T, double *x, double *xdes, typeUSERPARAM *
    userparam)
{
        double *pCost = (double*)userparam;
        double w_theta = pCost[4];
        out[0] = w_theta * (x[0] - xdes[0]) * (x[0] - xdes[0]);
}

/* Inequality constraints h(t,x(t),u(t),userparam) <= 0 */
void hfct(double *out, double t, double *x, double *u, typeUSERPARAM *userparam)
{
        double *pCons = (double*)userparam;
        out[0] =  pCons[6] - x[0];
        out[1] = -pCons[7] + x[0];
        out[2] =  pCons[8] - x[1];
        out[3] = -pCons[9] + x[1];
}
```

Problem-specific parameters can be accessed inside the functions via the pointer `userparam`. The variable type `typeUSERPARAM` passes the model parameters, cost function weights, and constraint

values to the function templates via the `void`-pointer. The function names follow the nomenclature of the general OCP formulation in (4.5) apart from `ocp_dim` which defines the dimensions of the MPC problem. The GRAMPC tool also requires explicit definition of the function derivatives w.r.t. state $\boldsymbol{x}$ and control input $\boldsymbol{u}$ in order to evaluate the optimality conditions. Jacobians have to be represented for use in multiplied form, i.e. $(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}})^\top \boldsymbol{\lambda}$. This helps to avoid zero multiplications in case of sparse Jacobians. The following listing shows a truncated selection of the corresponding gradient functions for state derivatives.

```
/* Jacobian df/dx multiplied by vector vec, i.e. (df/dx)^T*vec */
void dfdx_vec(double *out, double t, double *x, double *vec, typeUSERPARAM *
    userparam)
{
        double* pSys = (double*)userparam;
        double A = pSys[0];
        double B = pSys[1];
        double J = pSys[2];
        double tau_g = pSys[3];
        out[0] = vec[1] * (-tau_g * cos(x[0]) / J);
        out[1] = vec[0] + vec[1] * ((A * (10 - 10 * tanh(10 * x[1]) * tanh(10 *
            x[1])) - B) / J);
}
...
/* Gradient dl/dx */
void dldx(double *out, double t, double *x, double *xdes, typeUSERPARAM *
    userparam)
{
    double *pCost = (double*)userparam;
        double w_theta = pCost[4];
        out[0] = 2 * w_theta * (x[0] - xdes[0]);
}
...
/* Gradient dV/dx */
void dVdx(double *out, double T, double *x, double *xdes, typeUSERPARAM *
    userparam)
{
        double *pCost = (double*)userparam;
        double w_theta = pCost[4];
        out[0] = 2 * w_theta * (x[0] - xdes[0]);
}
...
/* Jacobian dh/dx multiplied by vector vec, i.e. (dh/dx)^T*vec */
void dhdx_vec(double *out, double t, double *x, double *vec, typeUSERPARAM *
    userparam)
{
        out[0] = -vec[0] + vec[1];
        out[1] = -vec[2] + vec[3];
        out[2] = 0;
        out[3] = 0;
}
...
```

GRAMPC provides a number of functions for initialising the MPC solver. The following listing shows part of how to initialise the application example MPC problem using the exoskeleton model parameters from the literature [176].

```
#define A       0.187f
#define B       0.408f
#define J       0.080f
```

```
#define tau_g   0.415f
#define w_theta 50
#define w_tau   1
...
/* User parameters */
double pSys[9] = { A , B , J , tau_g , w_theta, w_tau, 0.0 , 1.4 , -2 , 2 };
typeUSERPARAM *userparam = pSys;

/* GRAMPC initialisation */
grampc_init(&grampc, userparam);

/* Set parameters */
grampc_setparam_real_vector(grampc, "x0", [ 0.2, 0.0 ]);
grampc_setparam_real_vector(grampc, "xdes", [ 1.2, 0.0]);
grampc_setparam_real_vector(grampc, "u0", [ 0.0 ]);
grampc_setparam_real_vector(grampc, "udes", [ 0.0 ]);
grampc_setparam_real_vector(grampc, "umax", [ 20 ]);
grampc_setparam_real_vector(grampc, "umin", [ -20 ]);
grampc_setparam_real(grampc, "Thor", 0.2);
grampc_setparam_real(grampc, "dt", 0.002);

/* Set options */
grampc_setopt_int(grampc, "Nhor", 20);
grampc_setopt_int(grampc, "MaxGradIter", 4);
grampc_setopt_int(grampc, "MaxMultiIter", 6);
grampc_setopt_string(grampc, "InequalityConstraints", "on");
grampc_setopt_string(grampc, "Integrator", "heun");
...
```

The listing shows the initial and desired system states and control input as well as the rigidly enforced box constraints on the input ($u_{\max}$,$u_{\min}$). It also shows some of the algorithmic settings, e.g. the number of discretisation points `Nhor` for the prediction horizon $[0, T]$, the maximum number of iterations ($i_{\max}$,$j_{\max}$) for the augmented Lagrangian multiplier update and projected gradient optimisation algorithms, and choice of integration scheme for solving the dynamical system equations. Options for the integration scheme are (modified) Euler, Heun, 4th order Runge-Kutta as well as the solver RODAS [188]. The Euler and Heun methods are a fixed step size, determined by the number of discretisation points (`Nhor`), whereas RODAS and Runge-Kutta use adaptive step size control. Choice of integrator has a significant impact on the computation time and allows a trade-off between accuracy and computational efficiency. Other options not shown in the listing include the settings for scaling and offsetting the input and state variables of the optimisation problem (`xScale`,`xOffset`,`uScale`,`uOffset`).

The example constrained nonlinear MPC problem (4.11) was formulated using the GRAMPC tool and simulated in the MATLAB/Simulink environment. Human involvement, external disturbance, and modelling error are not introduced in this example, so the plant model and internal MPC model have identical prop-

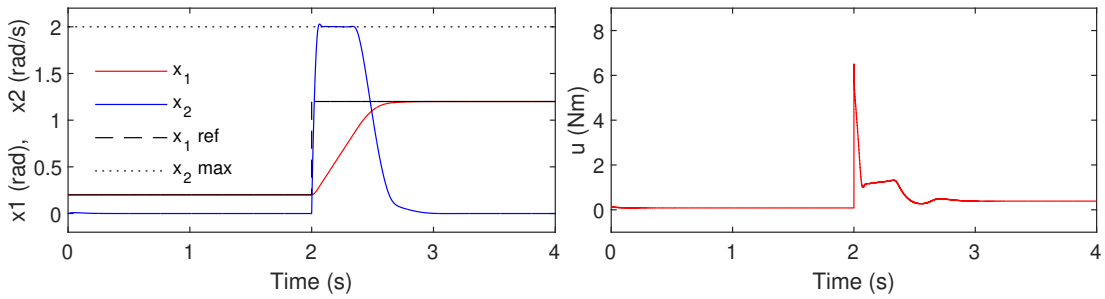


Figure 4.11: MPC Application Example. Simulated MPC trajectories of state, $\boldsymbol{x}$, and control input, u, for a 1-DOF exoskeleton example problem for a step response.

erties. System state and control input responses are shown for a unit step response with knee joint angle reference in Figure 4.11.

At 2 s in the simulation, the reference knee joint angle x_1 jumps from 0.2 rad to 1.2 rad. The MPC responds by rapidly increasing control input torque u. Shortly after this increase, the nonlinear constraint on knee joint velocity of $x_2 \leq 2$ from (4.11c) is activated and the joint velocity is limited to a maximum of 2 rad/s by reducing u after marginally violating the constraint. The behaviour regarding violation of constraints can be made more compliant by increasing the maximum number of iterations of the augmented Lagrangian and gradient algorithms (i_{max} and j_{max}), in particular the former, i. For the remaining simulation time, the reference angle is tracked closely and the MPC provides the (sub)optimal control input without violating constraints on system states, $\boldsymbol{x}$, or control input, u.

4.3.7 Cost Function

The proposed AAN-MPC control framework uses the MPC tool specified in Section 4.3.2 applied to the human-exoskeleton systems modelled in Sections 4.2.4.1 and 4.2.4.2. Instead of using a conventional cost function formulation for trajectory following, such as (4.11), the AAN-MPC controller uses a novel cost function which enables assistance to be provided in a number of modalities: passive mode, active-assist mode, and safety mode. The controller provides intelligent constrained nonlinear optimal control for

the human-exoskeleton system according to information about human effort, tracking error, and suitable assistance mode. The optimisation problem of the AAN-MPC for the seated configuration can be written as

$$\min_{u} \quad J = \sum_{k=t}^{t+T-1} (m_k w_\theta (\theta_{r,k} - \theta_k)^2 + w_\tau u_k^2) \tag{4.12a}$$

$$s.t. \quad \ddot{\theta}_{k+1} = \frac{1}{J}(B\dot{\theta}_k + \tau_g \sin\theta_k - \hat{\tau}_{h,k} - \tau_{e,k}) \tag{4.12b}$$

$$\theta \in [\theta_{\text{min}} \ \theta_{\text{max}}], \quad \dot{\theta} \in [\dot{\theta}_{\text{min}} \ \dot{\theta}_{\text{max}}] \tag{4.12c}$$

$$\tau_e \in [\tau_{e,\text{min}} \ \tau_{e,\text{max}}] \tag{4.12d}$$

where T shows the prediction horizon, and k denotes the kth sample. The prediction horizon can be chosen to reflect the length of time interval from EMG signal activation to joint torque output [189]. w_θ is the tracking weight, affecting tracking error cost, and w_τ is the torque weight, affecting the robot torque cost. m is the mode function which changes robot behaviour based on the preferred mode of assistance by scaling w_θ. The value of $m \in [0 \quad 1]$ is determined by a fuzzy logic-based algorithm, (described in section 5.2) which uses inputs of the reference angular velocity, $\dot{\theta}_r$, and the estimated human torque, $\hat{\tau}_h$, evaluated using a human torque estimation method, described in section 6.2.

A benefit of the AAN-MPC approach over existing AAN controllers is the ability to find optimal solutions while applying constraints to the state variables and control input by adjoining the constraint equations to the objective function. Constrained optimisation inherently limits the range and velocity of joint motion to prevent subject discomfort and protect hardware. Constraints of minimum and maximum knee joint angle, $\theta \in [0 \quad 1.4]$, angular velocity, $\dot{\theta} \in [-2 \quad 2]$, and control input torque, $\tau_e \in [-20 \quad 20]$, are applied. The constraints on joint states can be used by clinical practitioners to personalise the control framework for the patient based on recovery progress and physical limitations.

4.4 Controller Simulation

The proposed AAN-MPC controller is tested using a simulation environment in Simulink with the system system model from (4.2.4.1) for the seated leg extension and flexion exercise integrated over time using a variable step automatic solver. The layout of the simulation environment is shown in Figure 4.12. It includes blocks for the proposed controller and the benchmark PID and Impedance controllers. The environment has functionality for enabling/disabling control framework features of simulated human torque, τ_h, human torque estimation (HTE), and fuzzy logic-based algorithm (FLA) for assistance mode (see Section 5.2). There is also functionality for introducing modelling error (between MPC model and plant model) and configuring process (plant input) and measurement (plant output) noise.

4.4.1 Constrained Assistance

A significant benefit of the AAN-MPC approach proposed in this thesis over traditional approaches such as PID control, impedance control, and velocity/force field control, is the ability of MPC to include constraints on system states or control inputs in the overall optimisation problem. Inequality constraints can be defined when designing the controller in the problem formulation and the constraint minimum and maximum parameter values can be updated easily. Constraints can be enforced on any system state, or any linear/nonlinear combination of system states. In the context of robotic rehabilitation and assistance, this makes it possible to design a controller with safety considerations at the centre of the control policy. Coordinated movement between joints and/or limbs can be enforced in systems with multiple degrees of freedom. This is shown to be beneficial in [111] and demonstrated in [190]. A simple example of a coordinated movement goal in a multi-DOF system would be for the hip joint angle to be within a certain range of the knee joint angle during a sit-to-stand movement. Another application of constrained optimisation is

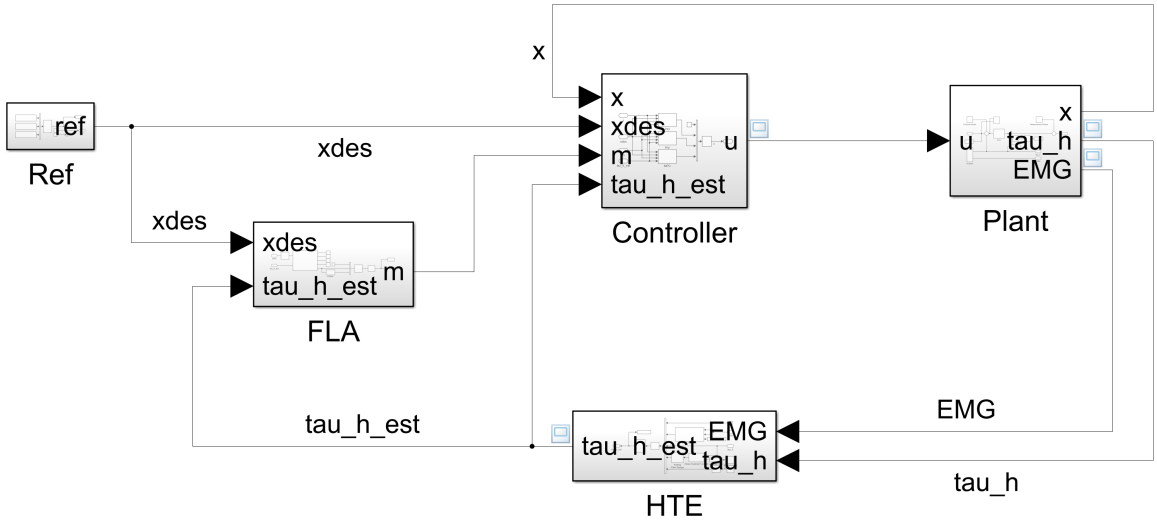


Figure 4.12: Controller Simulation Environment. Schematic showing the layout of the Simulink controller simulation environment, including: the state references block (Ref), PID/Impedance/MPC controller configuration block (Controller), human-exoskeleton system plant block (Plant), Human Torque Estimation (HTE) block, and the Fuzzy Logic Algorithm (FLA) block.

enforcing a velocity limit on a system state. These limits can be variable, functions of time or position, and need not be be the same in each direction of motion. Physical limitations can be implemented explicitly using MPC and could be useful for clinical rehabilitation practitioners seeking to modify robotic assistance for a specific patient, given their ability and requirements. Knee joint position and velocity constraints for the 1-DOF knee joint exoskeleton are demonstrated in simulation.

4.4.1.1 Position Constraints

Figure 4.13 shows a simulation of the proposed AAN-MPC controller during a seated leg extension end flexion tracking task, with maximum position constraints varying from 0.4 rad to 1.0 rad in increments of 0.2 rad. For all of the simulations, the reference joint angle follows an identical path. As the position constraints are adjoined to the cost function, activation of the maximum position constraints causes an increase in the cost value, shown in Figure 4.13 c), with the more strictly limiting constraints causing a greater increase in cost.

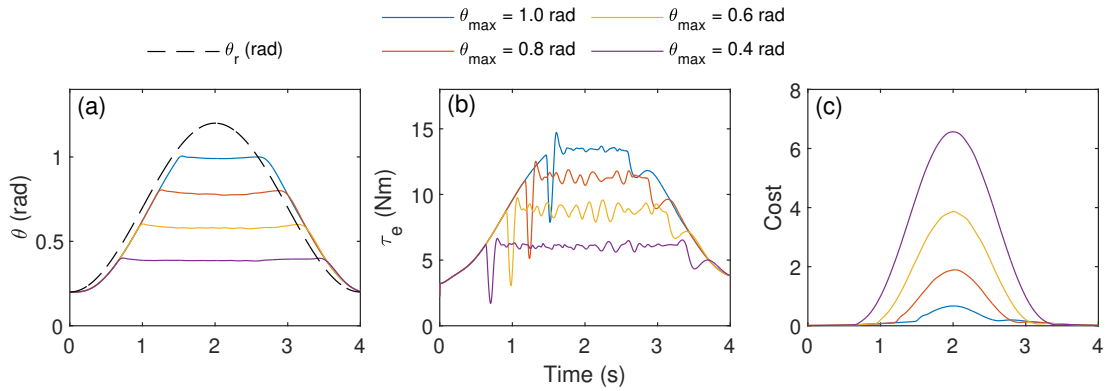


Figure 4.13: Constrained Position MPC Simulation. Simulation demonstrating MPC using maximum position constraints adjoined to the objective function during seated leg extension and flexion exercise. The maximum position constraint, $\theta_{\max}$, was varied from 0.4 rad to 1.0 rad in 0.2 rad increments.

4.4.1.2 Velocity Constraints

Figure 4.14 shows a simulation of the proposed AAN-MPC controller for a knee joint reference angle step response, beginning at 0.2 rad with a step of 0.5 rad. For all of the simulated step responses, the reference values are identical, however the maximum velocity constraints are of different magnitude, ranging from 0.2 rad/s to 2.0 rad/s. For each of the velocity constraint cases, following the joint angle step change, occurring at 0 s, the joint velocity rapidly rises to the level of the constraint, using the maximum available joint torque of 20 Nm, and remains at the maximum constrained velocity level until the reference joint angle is met. The explicit constraint of joint velocity is a powerful feature of the AAN-MPC framework. It can be used to design controllers for robotic assistance which implicitly consider the safety of the subject and comfortable nature of motion for each joint. The maximum and minimum velocity parameters can be set, depending of the mobility and rehabilitation progress of a subject's joints.

4.4.2 Comparison Study

The controller simulation environment, shown in Figure 4.12, was used to conduct a controller comparison, in which the proposed AAN-MPC controller is tested and compared to benchmark controllers (PID and

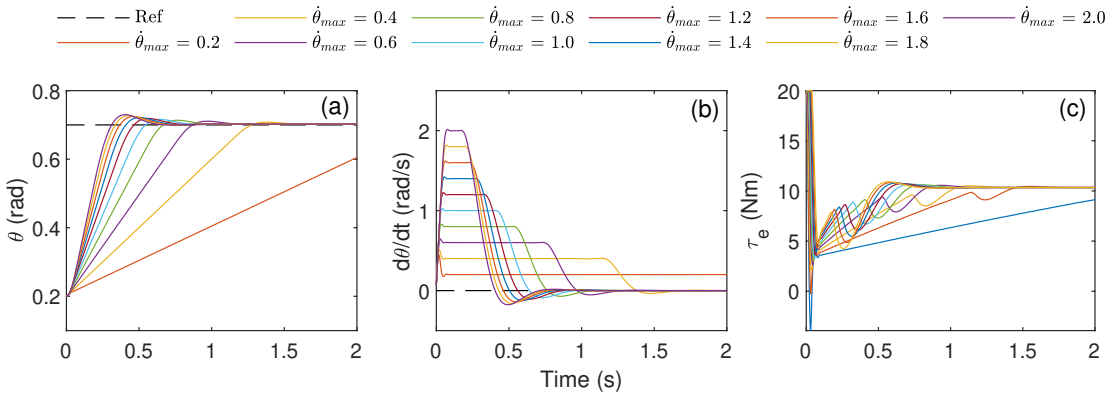


Figure 4.14: Constrained Velocity MPC Simulation. Simulation demonstrating MPC using maximum velocity constraints adjoined to the objective function during seated leg extension step response exercises. The maximum velocity constraint, $\dot{\theta}_{\max}$, varies from 0.2 rad/s to 2.0 rad/s in 0.2 rad/s increments.

Impedance) in terms of performance across a number of important scenarios: response to a reference step change; response to sudden human torque; reference tracking; presence of controller modelling error; and advance adaptation.

4.4.2.1 Step Response

The first test of the simulated controller comparison study demonstrates the responses of the following controllers to a reference joint angle step change from 0.2 rad to 1.0 rad occurring at 0.5 s: a finely tuned PID controller (using transfer function-based automated tuning in the PID Tuner App of MATLAB R2020a); a model-based impedance controller, shown in Figure 3.2, with viscous friction and gravity compensation (using identical model parameters to the MPC) and comparable stiffness and damping parameters ($K_p = 100$ and $K_d = 10$) to [137]; and the proposed MPC controller, with constrained joint position and velocity, omitting human torque estimation and the fuzzy logic algorithm. The responses of the controllers are shown in Figure 4.15.

The PID controller displays significant overshoot following the joint angle reference step change, with joint angle extending past the reference point of 1.0 rad to almost 1.2 rad, before eventually returning to a steady

state close to the reference value around a time of 2.0 s. This increased settling time is undesirable, and suggests that PID control is unsuitable for controlling this highly nonlinear system. The PID controller also displays initial position error, caused by uncompensated gravitational acceleration, which is largely corrected by the time the step change occurs at 0.5 s. Because both the MPC and impedance controllers include gravity compensation, this initial position error is not observed for them. The impedance controller has a similar rise time to the PID controller and much less overshoot. The settling time is much smaller than with PID control, suggesting that model-based impedance control benefits significantly from the viscous friction and gravity compensation features when applied to this highly nonlinear problem. The MPC controller demonstrated a greater rise time than the PID and impedance control approaches. This is not a limitation of the MPC, but a design decision, as the maximum joint velocity was constrained to 2 rad/s, as indicated in Figure 4.15. The joint velocity constraint can protect the subject from discomfort and injury in the case of a sudden change in reference state, and can be set to an appropriate value based on the condition and rehabilitation progress of the subject's joint. Activation of this constraint occurs between 0.6 s and 0.9 s in the middle plot of Figure 4.15. The MPC controller had the least overshoot, and preemptively reduces output torque after 0.8 s to prepare for approaching the reference position, demonstrating the predictive advantage of MPC. The controller torque outputs were limited to ± 20 Nm, as shown in the right plot of Figure 4.15, to represent the limitations of the actuator specified in Section 4.2.1.

4.4.2.2 Human Torque Response

The next part of the simulated controller comparison study tested the response to a sudden 10 Nm human torque step for a constant reference position with the PID, impedance, and MPC controllers, as well as the MPC controller with ideal human torque estimation ($\tau_h = \hat{\tau}_h$). The responses of the four controllers is shown in Figure 4.16.

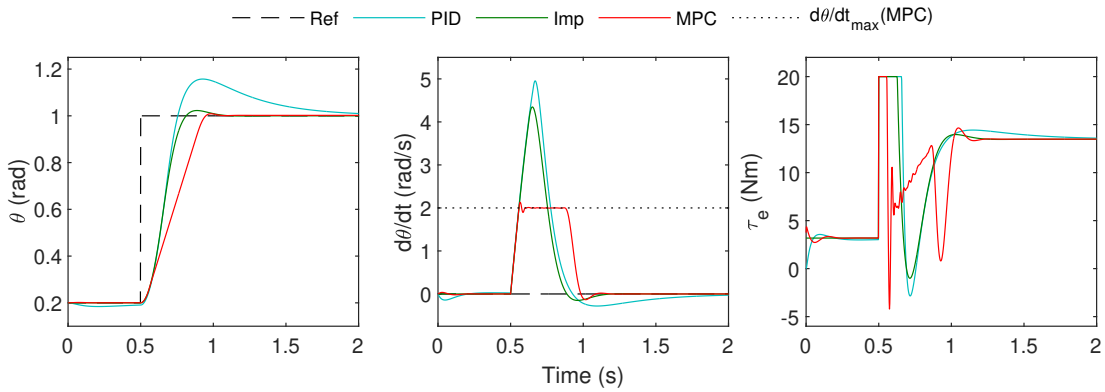


Figure 4.15: Simulated Controller Step Response Comparison. The simulated responses of the PID, Impedance, and MPC (without FLA and HTE) controllers are shown for a position reference step change from 0.2 rad to 1.0 rad.

The impedance controller demonstrated the greatest compliance, extending from the constant reference position of 0.2 rad to a steady state position around 0.3 rad. The constant positional error demonstrates the flexible, spring-like properties associated with impedance control. The PID controller showed little compliance, deflecting to a position around 0.24 rad initially, with the position error reducing over time. The reduction in residual error over time under constant disturbance is associated with the integral term of the PID which adds a control effect proportional to the accumulative value of the error. The MPC controller deflects to a steady state position around 0.23 rad, with an error magnitude dependent on the relative weighting of tracking and robot torque in the MPC cost function. When the MPC controller is combined with ideal human torque estimation (MPC w/ HTE in Figure 4.16), the controller can reject the disturbance of sudden human torque, minimising position error. In practice, ideal human torque estimation is not possible, however this simulation demonstrated the value of including estimation in the control framework. In Chapter 6, human torque estimation is investigated and a method for human torque prediction using a combination of muscle activity signals and system states is developed.

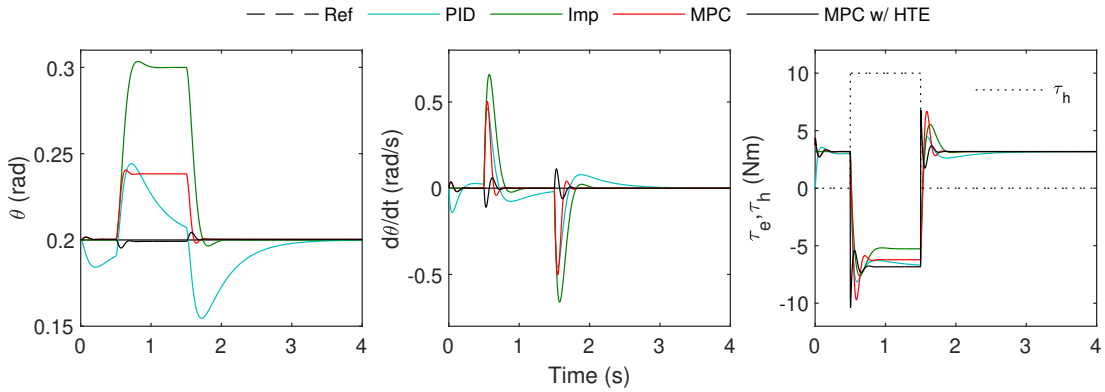


Figure 4.16: Simulated Controller Human Torque Response Comparison. The simulated responses of the PID, Impedance, MPC (without FLA and HTE), and MPC (with HTE) controllers are shown for a constant position reference and sudden 10 Nm human torque step for a duration of 1.0 s.

4.4.2.3 Seated Tracking

Controller tracking for seated leg extension and flexion is demonstrated in simulation for the PID, impedance, MPC, and MPC w/ HTE controllers. Three simulation conditions are tested for the seated tracking exercise: default (high gain) controller parameter values with no human torque ($\tau_h = 0$); default controller gains with a sudden human torque step; and significantly reduced controller gain parameters (by an order of magnitude) with the same human torque step. Simulating the controller responses with reduced gains emphasises the disparity in controller behaviour to demonstrate the relative merits and drawbacks of each approach. The responses of the four controllers in each of the three conditions is shown in Figure 4.17.

For the condition of default gains with no human torque, all controllers perform well and closely track the 0.25 Hz sinusoidal reference joint angle trajectory. For the two conditions of default and reduced gains with a 10 Nm sudden human torque step, introduced at 0.5 s for a duration of 1.0 s, the impedance controller demonstrates the greatest compliance, with noticeable steady state position error during the disturbance. The PID controller initially allows some position error before the integral term acts to reduce the positional error effect. The MPC without human torque estimation, as in Section 4.4.2.2, shows steady state positional error proportional to the ratio of tracking and robot torque weights in the MPC cost func-

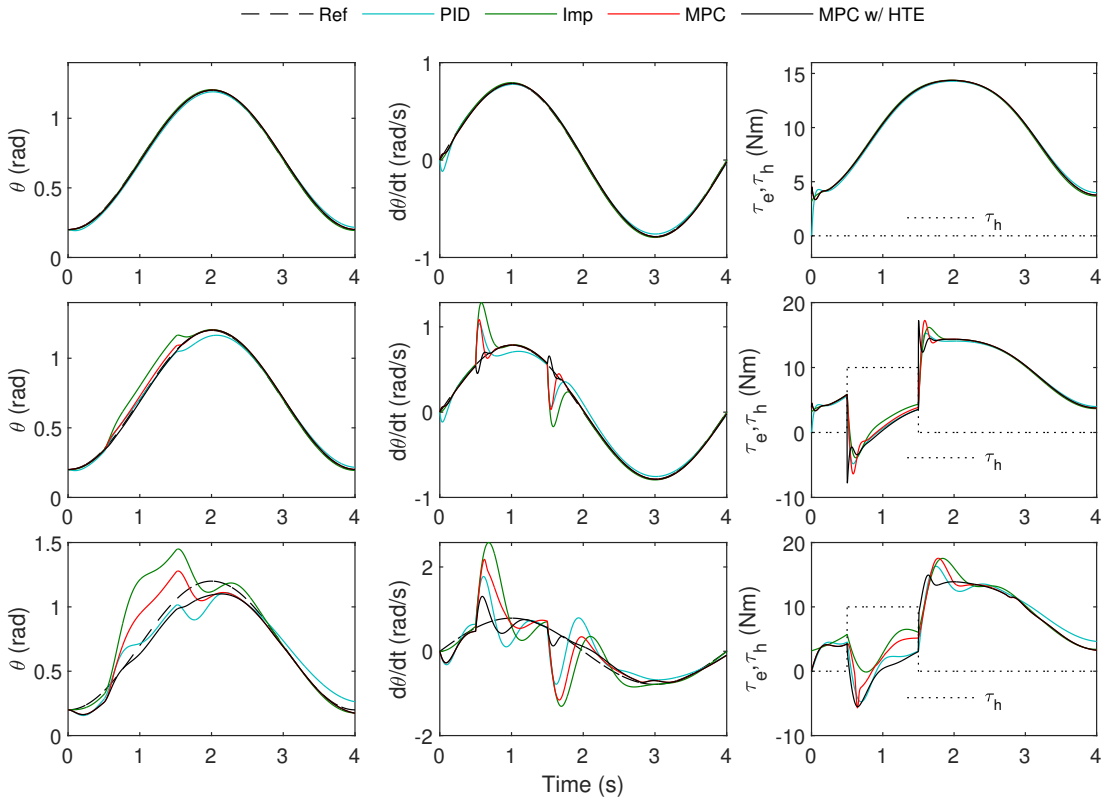


Figure 4.17: Simulated Controller Seated Tracking Comparison. The simulated tracking performance of the PID, Impedance, MPC (without FLA and HTE), and MPC (with HTE) controllers are shown for seated leg extension and flexion of the human-exoskeleton system in three cases: on the top row, high gains with no human torque; on the middle row, high gains with human involvement; and on the bottom row, low gains with human involvement.

tion. The MPC w/ HTE demonstrated the least positional error by far. This further emphasises the benefit of including human torque estimation in the control framework for exercise tasks where close tracking is important, even in the case of sudden human effort.

4.4.2.4 Model Error

This controller comparison test investigated the effect of introducing model error in the form of discrepancy between the controller model parameters (for model-based impedance and MPC control) and simulation plant model parameters. This was tested for seated leg extension and flexion exercise using the PID, impedance, and MPC (without HTE) controllers with no human torque ($\tau_h = 0$). Three different conditions

were simulated: no model error, $+50\%$ plant model error, and -50% plant model error. The errors were applied to the static friction, A, viscous friction, B, inertia, J, and gravitational torque, τ_g, parameters of the plant. The increasing (+ve) plant model errors caused the controllers to underestimate the plant parameters, whereas the decreasing (-ve) plant model errors caused the controllers to overestimate the plant model parameters. Results of the simulations are shown in Figure 4.18.

The joint angle and tracking error plots on the left of Figure 4.18 simulate seated tracking with no modelling errors. The controllers track the reference position closely with minimal tracking error. The middle plots show simulated tracking with $+50\%$ plant model error, and the right plots show tracking with -50% plant model error. The PID controller is least affected by the model error. As this control approach is not model-based, tracking performance is not dependent on model parameters being representative of the plant dynamics. The tracking performance of the MPC controller, and to a greater extent the impedance controller, are significantly impacted by the introduction of model error. The MPC controller is less affected because high tracking weight in the cost function has a corrective effect on the output torque. The impedance controller is more affected because the stiffness and damping parameters are set to provide compliant, human-like properties at the robot joint, and so have less corrective effect. This may be favourable in situations where compliance is of greater importance than tracking, however there is a significant trade-off between them. The impedance and MPC controllers underestimating plant model parameters causes undershoot from the reference trajectory, and overestimating the parameters causes overshoot. The mean simulated tracking errors for all three plant error conditions are summarised in Table 4.3.

Table 4.3: Mean Tracking Errors. Mean tracking error values (rad) simulated for seated leg extension and flexion of the human-exoskeleton system with the PID, impedance, and MPC (without FLA and HTE) controllers with various plant model error, e, values.

	e = 0	e = + 0.5	e = - 0.5
PID	-0.0019	-0.0029	-0.0009
Imp	0.0001	-0.0498	0.0557
MPC	0.0056	-0.0216	0.0345

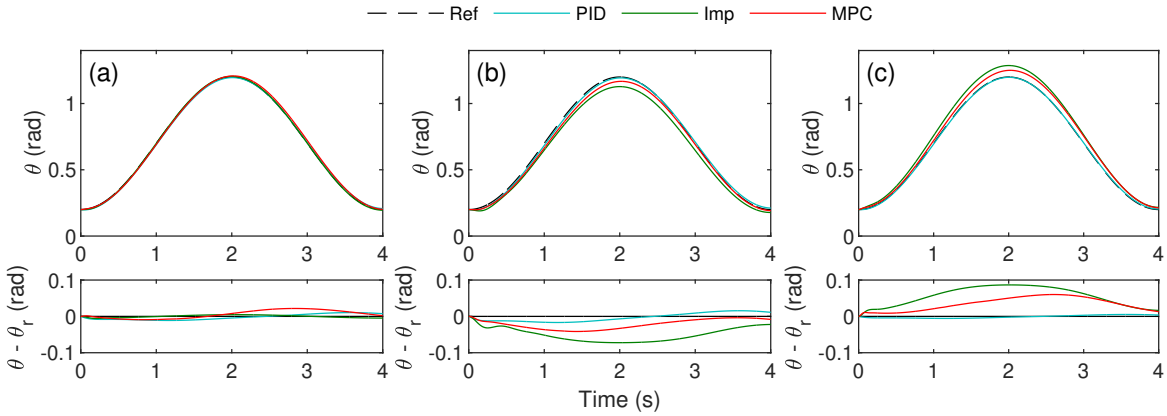


Figure 4.18: Simulated Controller Tracking Comparison with Modelling Errors. Simulated tracking performance for seated leg extension and flexion of the human-exoskeleton system with the PID, impedance, and MPC (without FLA and HTE) controllers with various plant model error, e, values. The upper row shows reference, θ_r, and simulated, θ, angular positions, and the lower row shows the position errors, $\theta - \theta_r$.

4.4.2.5 Advance Adaptation

An advantage of the control approach proposed in this thesis over existing AAN controllers is the ability to predict human torque from muscle activity signals in advance of joint torque output and optimise control input accordingly. To show the theoretical benefit of this, a simulation of the response to sudden human joint torque is simulated for the MPC controller and PID controller. The delay between EMG signals firing in the muscle and force being produced in the muscle is significant [189]. Considering latency introduced by the torque estimation method, a human torque prediction 0.1 s in advance is simulated for the MPC. The output of the FLA, described in Section 5.2, is included so the assistance mode changes from passive to active-assist. This mode demonstrates more compliance than passive mode, permitting greater tracking error.

The simulation is shown in Fig. 4.19. It demonstrates the MPC controller reacting in advance to the human torque, adjusting τ_e at 0.4 s to accommodate human-robot interaction. The PID controller with no human torque prediction does not adjust τ_e until after the human torque is produced at 0.5 s.

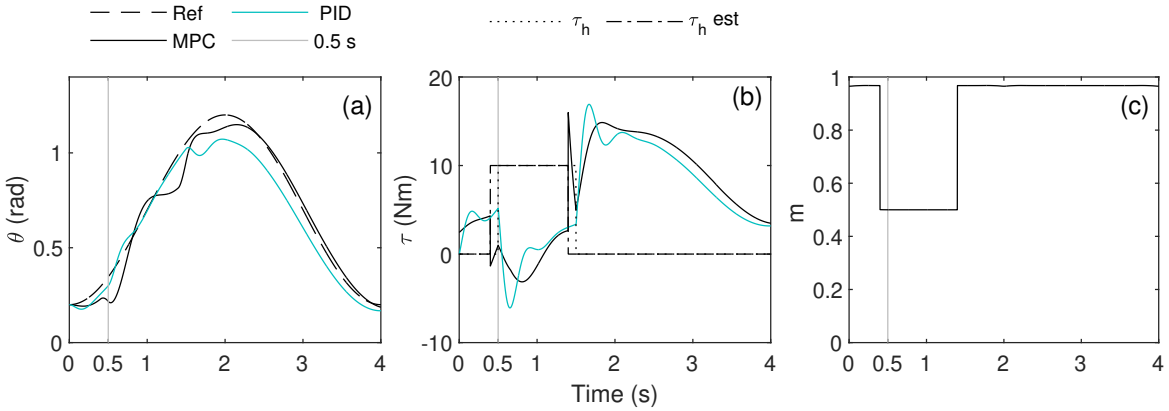


Figure 4.19: Simulated Controller Advance Adaptation. Simulation of the MPC and PID controller responses to a human torque step of 10 Nm from 0.5 s to 1.5 s. Human torque estimated using muscle activity signals, $\hat{\tau}_h$, is observed 0.1 s in advance of the human joint torque output, τ_h. The MPC changes assistance mode from passive to active-assist at 0.4 s, indicated by a change in the assistance mode variable, m. This increases compliance to accommodate human-robot interaction.

Chapter Summary

In this chapter, MPC was established as the central component of the AAN-MPC control framework, which uses the decision-making power of fuzzy logic to estimate the most suitable mode of assistance by evaluating human-robot interaction condition in real-time. The framework uses a human torque estimation function to predict human joint torque from muscle activity information and system states. The human-exoskeleton system design configurations and models were presented for seated and standing exercises, targeted at robotic rehabilitation and mobility assistance. Using the system models, the control framework can provide informed optimal assistance, specific to the exercise and conditions in question. The controller development workflow, design, and implementation were presented with an example application. A controller simulation environment was presented and, finally, the results of a simulated controller comparison study were presented and discussed.

Chapter 5

Fuzzy Logic Algorithm for Assistance Mode

The previous chapter described the proposed AAN-MPC control framework. Human-exoskeleton system were presented, including an outline of the exoskeleton configurations, actuator module, system models, and parametric identification methods. The MPC design and implementation was detailed, and an application example was provided. The chapter concluded with presentation of a controller simulation environment and a comparison study involving various benchmark controllers.

The focus of this chapter is the component of the control framework that evaluates the assistance mode. A fuzzy logic-based algorithm is developed for evaluating the assistance mode variable m, which is used in the MPC cost function to adapt the nature of human-robot interaction. The algorithm designed in this chapter uses inputs of estimated human torque, $\hat{\tau}_h$, and knee joint reference velocity, $\dot{\theta}_r$. For the control framework evaluation in Chapter 7, human torque is estimated using the linear model (n=2) described in Chapter 6.

In this chapter the background section provides context on fuzzy logic and a brief review of how it has been used previously in AAN control. The materials and methods section describes development of the

algorithm, the role of the p_A parameter, and a trajectory adaptation method. The chapter concludes with a results and discussion section in which the algorithm performance is tested for a number of human involvement conditions, the effect of the p_A parameter value is explored, the trajectory adaptation method is demonstrated, and application of the assistance mode function to various benchmark controllers is compared in simulation.

5.1 Background

Fuzzy logic is a form of many-valued logic in which the truth (or likelihood) values of variables may be any real number between 0 and 1, both inclusive. It is used to handle the concept of partial truth, where the true value of some state or property may range between completely true and completely false. It is in contrast to Boolean logic, where truth values of variables may only be the integer values 0 or 1 [191]. Fuzzy logic is based on the observation that people make decisions based on imprecise and non-numerical information. It is a way of representing vagueness and imprecise information in a way that is rigorous and can be computed repeatedly.

Fuzzy logic has been applied to many fields of research, from control theory to artificial intelligence, and used in the design of control systems for rehabilitation robots. It was incorporated in an implementation of force-position hybrid control [192], where it realised a set of distinct position and force control rules to constrain movement to the desired direction and maintain a constant force along the moving direction. The authors in [193] and [194] used fuzzy control with FES for the subject joint angle, regulating the muscle stimulation pulse-width required to drive FES-assisted walking gait. The authors in [195] used fuzzy approximation to realise adaptive backstepping control of an exoskeleton for upper limbs. The approximators estimated the dynamical uncertainties of the human-robot system, and an iterative learning scheme com-

pensated for time-varying periodic disturbances. Fuzzy neural network approaches have been developed for robotic control, such as in [196], where a fuzzy neural network was used for nonlinear dynamics estimation. Fuzzy logic was used for a gait monitoring system in [197], using ground contact forces as inputs. In [198] and [199], fuzzy inference systems were used to regulate impedance controller parameters for upper limb robots.

5.2 Materials and Methods

This section describes the development of a novel fuzzy logic algorithm for determining assistance mode using fusion of data from exoskeleton and human states.

5.2.1 Fuzzy Logic-based Algorithm

An important element of the AAN-MPC control framework is the FLA for determining assistance mode. The role of this algorithm is to make an evaluation of the human-robot interaction mode based on the predicted human effort contribution and desired system state. This estimation is the likelihood of the subject either being passive, actively participating in the exercise task, or resisting the motion of the exercise. The resulting controller behaviour varies greatly depending on which of these human-robot interactions is shown by the subject. An accurate evaluation of interaction behaviour would make the algorithm very powerful. Potentially enabling the control framework to adjust on-the-fly, achieving subject-specific human-robot interaction for various exercise modes. This continuous decision making can enable the controller to behave intelligently and sensitively, akin to an experienced physical therapist.

To simplify the complex nature of human-robot interaction, distinct and representative human-robot in-

teraction modes are defined which can be evaluated using a small number of measurable inputs from human muscle activity and the human-exoskeleton system. The inputs that determine interaction mode at the joint are reference angular velocity, $\dot{\theta}_r$, and predicted human joint torque, $\hat{\tau}_h$, estimated using a HTE model (see Chapter 6). For most exercise tasks, including those in this study, the human-robot interaction condition can be evaluated at the (knee) joint based on the polarity of these two inputs.[1]

With the human behaving passively during exercise tasks, the model-based controller is designed to fully compensate for gravitational, inertial and frictional effects of the human limb and exoskeleton. Therefore, voluntary human effort is not required to support the device and limb. It can be inferred that estimated human joint torque is either contributing to the motion of the device, in order to complete the instructed rehabilitation task, or obstructing the motion of the device, due to resistance from the subject.

In the passive interaction condition, human torque is close to zero, regardless of the reference joint velocity polarity during the exercise task. In the condition of human assisting with the exercise task, the human joint torque polarity acts in the direction of reference joint velocity. Likewise, in the condition of human resisting an exercise task, the direction of human joint torque tends to oppose reference joint velocity polarity. The phenomenon of human joint torque and reference angular velocity polarity indicating human-robot interaction condition is utilised by the FLA to inform the controller. This information is included in the MPC cost function to adapt assistance during exercise to achieve optimal human-robot interaction on-the-fly for the conditions defined in the FLA.

The output of the FLA is *assistance mode*, represented by a single variable, m, as shown in Figure 5.1. This variable continuously changes controller behaviour, adapting the nature of human-robot interaction by acting on the MPC cost function (see Section 4.3.7). It scales the tracking weight (cost), w_θ, between 0 and 1 to change the robot's effective compliance, permitting a suitable ratio of human torque to robot

[1] In this context, polarity refers to whether the input value can be classified as close to zero, acting in the positive direction, or acting in the negative direction.

Figure 5.1: Fuzzy Logic-based Algorithm Inputs and Output. Schematic of the FLA function with inputs of reference angular velocity, $\dot{\theta}_r$, and estimated human torque, $\hat{\tau}_h$, at the joint, and output of the assistance mode variable, m.

torque during the exercise task. When the FLA output $m = 0$, tracking weight is eliminated from the MPC cost function and the system operates with the human dominant. It allows significant tracking error, showing maximum compliance. When the FLA output $m = 1$, tracking weight is the maximum default gain value and the system operates with the robot dominant, showing minimal compliance and allowing little tracking error. The value of m, and the nature of assistance, on the spectrum of human/robot dominance, is determined using fuzzy logic.

5.2.1.1 Assistance Modes

When reference angular velocity, $\dot{\theta}_r$, and estimated human joint torque, $\hat{\tau}_h$, act in the same direction, a cooperative human-robot interaction condition is identified and the FLA determines that the robot should operate in "Active-assist" mode – A. When $\hat{\tau}_h$ is small, the human-robot interaction condition is identified as passive and the FLA determines that the robot should operate in "Passive" mode – P. With $\hat{\tau}_h$ acting against the direction of $\dot{\theta}_r$, the FLA identifies a resistive human-robot interaction condition, and determines that the robot should operate in "Safety" mode – S. The three human-robot interaction conditions defined here are inclusive of all possible combinations of the inputs, and give rise to three corresponding "Modes" of robot operation: Active-assist (A), Passive (P), and Safety (S). The mapping of these modes on a Cartesian plane, with reference angular velocity, $\dot{\theta}_r$ on the x-axis, and estimated human joint torque, $\hat{\tau}_h$, on the y-axis is illustrated in Figure 5.2. Safety mode is inclusive of the region about zero reference angular velocity to

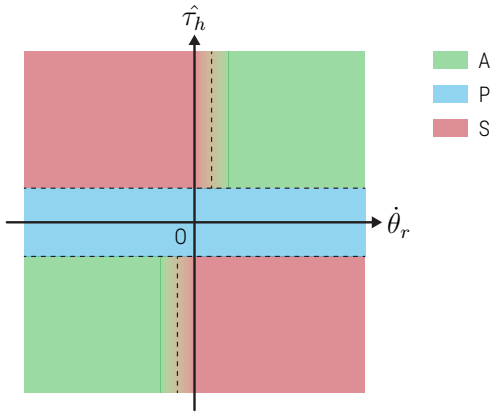


Figure 5.2: Assistance Mode Mapping. Assistance mode mapped against reference angular velocity, $\dot{\theta}_r$, on the x-axis and estimated human torque, $\hat{\tau}_h$, on the y-axis. The modes are Active-assist (A) – Green, Passive (P) – Blue, and Safety (S) – Red.

ensure that safety mode is not compromised, and robot dominance enforced, upon the reference position pausing in the same position or changing direction.

5.2.1.2 Applying Likelihood Values

During robot operation the assistance modes are assigned a membership value which represents the degree of truth. This can be interpreted as the likelihood that the current human-robot interaction condition belongs to each of the categories. The likelihood of mode A and S being suitable for the current interaction condition are defined as, $\mu_A \in [0 \quad 1]$ and $\mu_S \in [0 \quad 1]$. Using the aforementioned reasoning, a set of logical rules are designed, as shown in Table 5.1.

Table 5.1: Fuzzy Rule Bases. The fuzzy rules are outlined for the inputs of estimated human joint torque, $\hat{\tau}_h$, and reference joint velocity, $\dot{\theta}_r$, with input state combinations leading to active-assist, μ_A, or safety, μ_S, Fuzzy Membership Value (FMV)s.

Rule	$\hat{\tau}_h$	$\dot{\theta}_r$	FMV
1	Negative	Negative	$\mu_A \rightarrow 1$
2	Positive	Positive	$\mu_A \rightarrow 1$
3	Negative	Non Negative	$\mu_S \rightarrow 1$
4	Positive	Non Positive	$\mu_S \rightarrow 1$

The estimated human joint torque, $\hat{\tau}_h$, is classified into two levels, "Negative" and "Positive", while the reference angular velocity, $\dot{\theta}_r$, is classified into four levels, "Negative", "Positive", " Non Positive", and "Non Negative". For evaluating the "Negative" and "Positive" values for $\hat{\tau}_h$ and $\dot{\theta}_r$, a sigmoidal membership function is used. The sigmoidal function can be written as

$$f(x) = \frac{1}{1 + e^{-a(x-c)}} \tag{5.1}$$

where x is the input, and function parameters a and c affect steepness and centre of the distribution respectively. To find the membership values of "Non Positive" and "Non Negative" for $\dot{\theta}_r$, the fuzzy logic $\mathrm{NOT}(x)$ operator is used, given by

$$\mathrm{NOT}(x) = 1 - x \tag{5.2}$$

where x is the input. The memberships values can therefore be expressed as

$$f^{NP}(\dot{\theta}_r) = 1 - f^{P}(\dot{\theta}_r) \tag{5.3}$$

$$f^{NN}(\dot{\theta}_r) = 1 - f^{N}(\dot{\theta}_r) \tag{5.4}$$

The full set of membership functions are shown in Figure 5.3. All of the functions are continuous and smooth over the entire range of possible input values. This means controller behaviour does not change suddenly and jumping between discrete assistance modes is avoided [197]. This makes the control framework more stable and predictable, reduces the occurrence of unwanted vibrations, and leads to greater comfort for the subject. The function parameters are chosen such that membership values cannot sum to greater than one. Membership functions for all inputs are summarised in Table 5.2.

Figure 5.4 shows the complete scheme of the proposed fuzzy logic algorithm. The Larsen product impli-

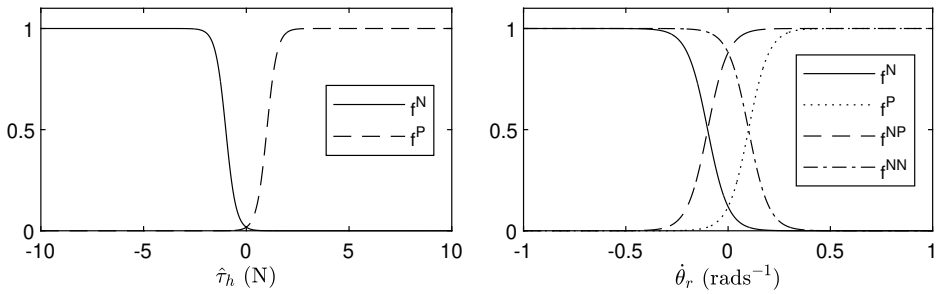


Figure 5.3: Membership Functions. From left to right, membership functions of estimated human torque for negative, $f^N(\hat{\tau}_h)$, and positive, $f^P(\hat{\tau}_h)$, and of the reference angular velocity for negative, $f^N(\dot{\theta}_r)$, positive, $f^P(\dot{\theta}_r)$, non positive, $f^{NP}(\dot{\theta}_r)$, and non negative, $f^{NN}(\dot{\theta}_r)$.

cation method is used as the inference operator to calculate the likelihoods μ_A and μ_S as follows [200]:

$$\mu_A = f^N(\hat{\tau}_h) \times f^N(\dot{\theta}_r) + f^P(\hat{\tau}_h) \times f^P(\dot{\theta}_r) \tag{5.5}$$

$$\mu_S = f^N(\hat{\tau}_h) \times f^{NN}(\dot{\theta}_r) + f^P(\hat{\tau}_h) \times f^{NP}(\dot{\theta}_r) \tag{5.6}$$

where f shows the membership function. The superscripts N, P, NP and NN denote the "Negative", "Positive", "Non Positive", and "Non Negative" respectively. For example, for the 3rd rule in Table 5.1, if $\hat{\tau}_h$ is negative and $\dot{\theta}_r$ is non negative (either zero or positive) (i.e. $f^N(\hat{\tau}_h) \to 1$ and $f^{NN}(\dot{\theta}_r) \to 1$) then safety mode likelihood tends to one, i.e., $\mu_S \to 1$. It should be noted that when the human is relaxed ($\hat{\tau}_h = 0$), both μ_A and μ_S are set to zero according to the designed membership functions (see Figure 5.3) and rules (see Table 5.1).

Table 5.2: Membership Function Details. Details and parameters of membership functions used in the fuzzy logic-based algorithm.

Membership Function	Type	σ	a	c
$f^{Negative}(\hat{\tau}_h)$	Sigmoidal	N/A	-4	-1
$f^{Positive}(\hat{\tau}_h)$	Sigmoidal	N/A	4	1
$f^{Negative}(\dot{\theta}_r)$	Sigmoidal	N/A	-20	-0.1
$f^{Positive}(\dot{\theta}_r)$	Sigmoidal	N/A	20	0.1

5.2.1.3 Assistance Mode Function

Based on the estimated likelihoods μ_A and μ_S, the assistance mode function, m is designed as follows:

$$m = 1 - (p_A \mu_A + p_S \mu_S) \tag{5.7}$$

Where p_A and p_S are the active-assist penalty and safety penalty. The p_A and p_S parameters are constants, with $p_A, p_S \in [0 \quad 1]$.

Using the assistance mode function, m, the three assistive modes, Passive (P), Active-Assist (A) and Safety (S), can be transferred between smoothly, according to the likelihoods μ_A and μ_S. For this study the parameters are set as $p_A = 0.5$ and $p_S = 1.0$, to reflect the desired level of robot compliance in each operation mode. In the passive mode, with $\mu_A \to 0$ and $\mu_S \to 0$, the assistance mode variable m is set to 1, irrespective of the p_A and p_S values, and human-robot interaction is exoskeleton dominant with minimal compliance and tracking error. In active-assist mode, as $\mu_A \to 1$, $m \to 0.5$ causing the tracking error cost in the MPC cost function (w_θ) to halve, making the robot compliant and enabling suitable human-robot cooperation during the tracking exercise. In safety mode, with $\mu_S \to 1, m \to 0$, eliminating tracking error cost and causing the robot to show high compliance.

5.2.2 p_A Parameter

In clinical practice, the value of the active-assist penalty parameter, p_A, can be tuned by a therapist for the needs of a patient to vary the ratio of human-to-robot effort exerted during a cooperative rehabilitation task. Alternatively, p_A could be automatically adjusted within the controller by an algorithm, based on a performance-based metric of subject ability. The p_A parameter could also be used to compensate for

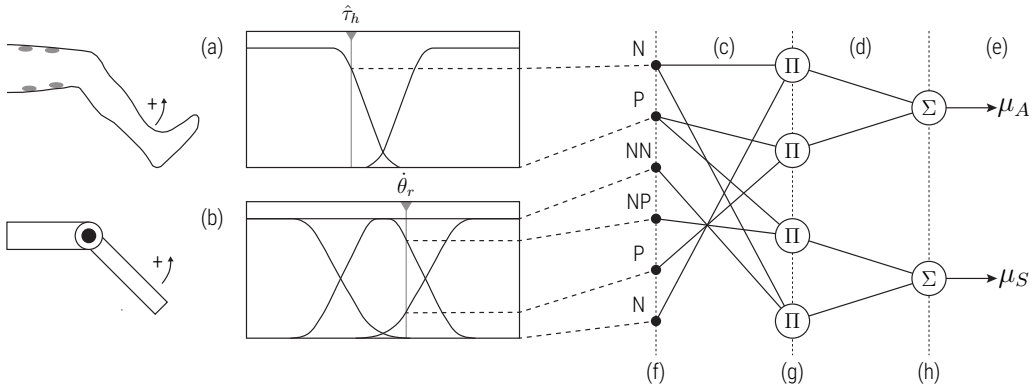


Figure 5.4: Fuzzy Logic-based Algorithm Schematic. Fuzzy logic algorithm for inferring the suitable assistance mode for the current human involvement condition. (a) Membership functions for estimated human joint torque. (b) Membership functions for reference angular velocity. (c) Fuzzy rule bases (Table 5.1). (d) Rule outputs before summation. (e) Outputs of fuzzy logic. (f) Fuzzy membership values. (g) AND operators – fuzzy rule bases. (h) OR operators – summation for assistance modes.

subject fatigue. This can be done by scaling the p_A parameter by a fatigue factor which is a function of time elapsed since the beginning of an exercise session. The ratio of robot effort during the task could be gradually increased over time, maintaining subject participation.

5.2.3 Trajectory Adaptation

Performance of the proposed AAN-MPC approach with FLA for assistance mode detection was evaluated for knee joint flexion and extension movements in a seated position. Knee joint movements of the swinging leg during a gait cycle have a pendulum oscillatory flexion and extension behaviour that is close to the natural walking frequency [176]. This makes such exercise suitable as a proxy for early-stage gait rehabilitation in stroke patients.

For the reference movement profiles, a sinusoidal knee joint angle trajectory with an amplitude of 1 rad and frequency of 0.25 Hz is considered. This can be written as

$$\theta_r(t) = 0.5\cos(2\pi f t - \pi) + 0.7 \tag{5.8}$$

where θ_r is the reference joint angle in radians, f is the repetition frequency in Hz, and t is the time in seconds.

Trajectory adaptation, triggered by excess human resistance, is investigated as a utilisation of high-level human-robot interaction information inferred from the FLA. The proposed implementation of the feature can be written as

$$\theta_r(k+1) = \begin{cases} 0.5\cos(2\pi f t_{k+1} - \pi) + 0.7 & \text{if } \mu_S \leq R_{th} \\ \theta_r(k) & \text{if } \mu_S > R_{th} \end{cases} \tag{5.9}$$

where k denotes the kth sample. R_{th} is a human resistance threshold. The time-dependent trajectory (5.8) is halted when the likelihood of the safety mode is higher than the resistance threshold, $\mu_S > R_{th}$. The reference joint angle remains constant until the estimated likelihood of the safety mode, μ_S, falls back below the threshold, at which point the reference trajectory continues. Trajectory adaptation can halt the reference task in real-time, making the exercise session more manageable for the subject. The threshold value R_{th} can be changed according to the subject's range of movement, fatigue level, or other clinical factor. The value could also be updated by an online algorithm.

5.3 Results and Discussion

5.3.1 FLA Performance

To demonstrate the behaviour of the FLA, five implementation test cases are highlighted. The cases are representative of possible human-robot interaction conditions for a 1-DOF robotic knee joint during extension and flexion exercise. The test cases demonstrate the FLA identifying the interaction condition and returning the corresponding assistance mode. The conditions are as follows:

a) **Neutral (N)**: Subject relaxes during the exercise, contributing minimal torque and allowing the robot to dominate effort contribution for extension and flexion movements – in this condition, the robot should operate in Passive mode (P).

b) **Extension Assist (EA)**: Subject contributes useful assistive torque during the extension phase of exercise, but relaxes during flexion – in this condition, the robot should operate in Active-assist mode (A) during extension.

c) **Extension Resist (ER)**: Subject contributes obstructive resistive torque during the extension phase of exercise, but relaxes during flexion – in this condition, the robot should operate in Safety mode (S) during extension.

d) **Flexion Assist (FA)**: Subject relaxes during extension, but contributes useful assistive torque during the flexion phase of exercise – in this condition, the robot should operate in Active-assist mode (A) during flexion.

e) **Flexion Resist (FR)**: Subject relaxes during extension, but contributes obstructive resistive torque during the flexion phase of exercise – in this condition, the robot should operate in Safety mode (S) during flexion.

The performance of the FLA, for test cases involving each of the human-robot interaction conditions is shown in Figure 5.5.

For each of the five test cases, the FLA performs well, consistently identifying the appropriate assistance mode for the extension and flexion phases of the movement. Importantly, with the subject in the Neutral condition, (N), both μ_A and μ_S remain low, demonstrating that the FLA can consistently recognise human relaxation and operate in Passive mode (P). The assistance mode variable, m, behaves as expected, falling close to zero for Safety mode (S), given $p_S = 1.0$, and close to 0.5 for Active-assist mode, given $p_A = 0.5$.

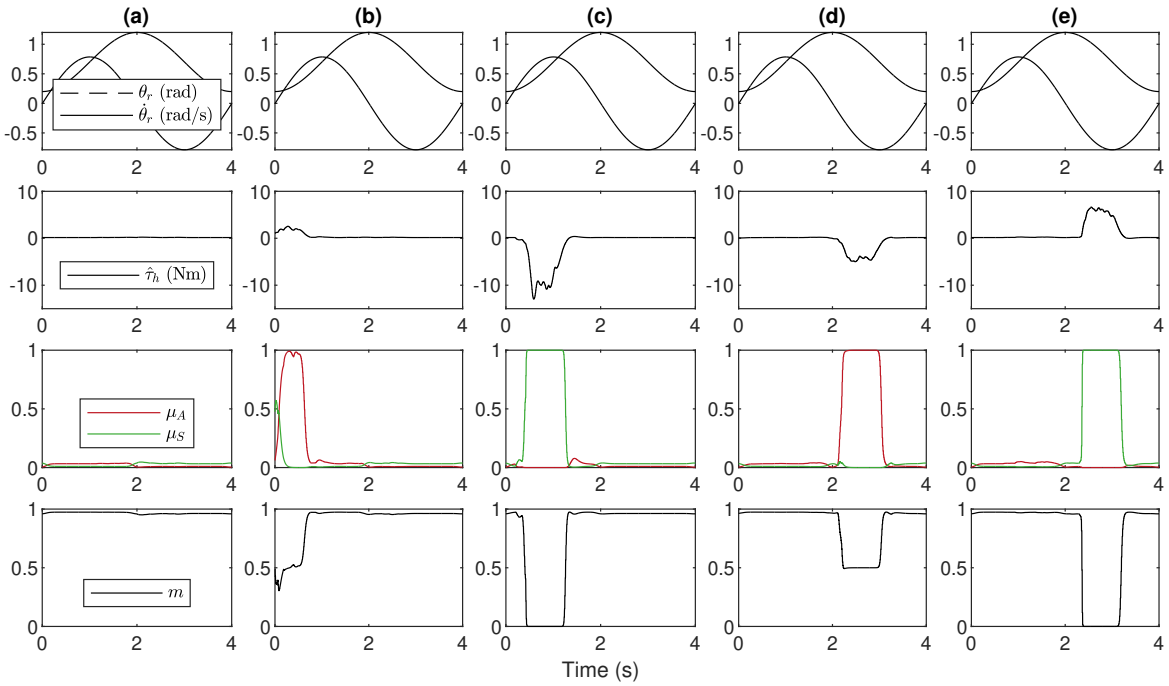


Figure 5.5: Fuzzy Logic-based Algorithm Performance. Fuzzy logic-based algorithm performance for knee joint trajectory tracking at 0.25 Hz in five human-robot interaction conditions. From left to right: a) Neutral (N), b) Extension Assist (EA), c) Extension Resist (ER), d) Flexion Assist (FA), and e) Flexion Resist (FR). From top to bottom, plots are of reference joint angle and angular velocity (θ_r and $\dot{\theta}_r$), estimated human joint torque ($\hat{\tau}_h$), active-assist and safety mode likelihoods (μ_A and μ_S), and assistance mode variable (m).

The observed assistance mode likelihood values show that μ_A and μ_S are activated on a mutually exclusive basis.

Although the FLA is proposed and demonstrated for a 1-DOF robotic knee joint, the approach is generalisable to any multi-DOF system, given that a good human joint torque estimate can be found for all DOFs. The FLA methodology can also be applied with respect to the reference velocity of an end-effector in polar or Cartesian coordinates, given that a transformation is applied for mapping the forward dynamics.

5.3.2 Effect of p_A

The effect of varying the p_A parameter on the FLA output, m, and controller performance was investigated. Controller behaviour was simulated for an assistive human-robot interaction condition during the extension phase of a seated leg extension and flexion exercise. The simulations were run for identical conditions over a period of 4 s, using offline data recorded during the human joint torque estimation data collection, described in Chapter 6. Simulations were conducted for values of the p_A parameter ranging from 0 to 1 inclusively at intervals of 0.1. The results of these simulations are shown in Figure 5.6.

As the p_A parameter is increased from 0 to 1, the assistance mode output variable m varies proportionally during assistive human-robot interaction conditions, as shown in Figure 5.6c. This reduces the MPC cost function tracking weight linearly as it is scaled by the factor of m. The effect on robot torque is a reduction in the magnitude of corrective action, which is perceived as an increase in robot compliance. The compliance is shown in Figure 5.6a as a significant increase in tracking error. It can be seen in Figure 5.6b that increasing p_A causes a reduction in robot torque, due to the increased relative weight of robot torque in the MPC cost function. During the simulations, which used offline data from previous experiments, the estimated human torque value briefly falls to a low value around 0.5 s. The controller reacts to this with an increase in the value of the assistance mode variable and a transition of behaviour from cooperative to dominant and back again.

5.3.3 Trajectory Adaptation

An example of the trajectory adaptation method for extension resistance can be seen in Fig. 5.7. When resistive human torque is estimated, the controller identifies the human-resistive involvement condition and an increase in safety mode likelihood μ_S is observed. The human resistance threshold, $R_{th} = 0.5$, is

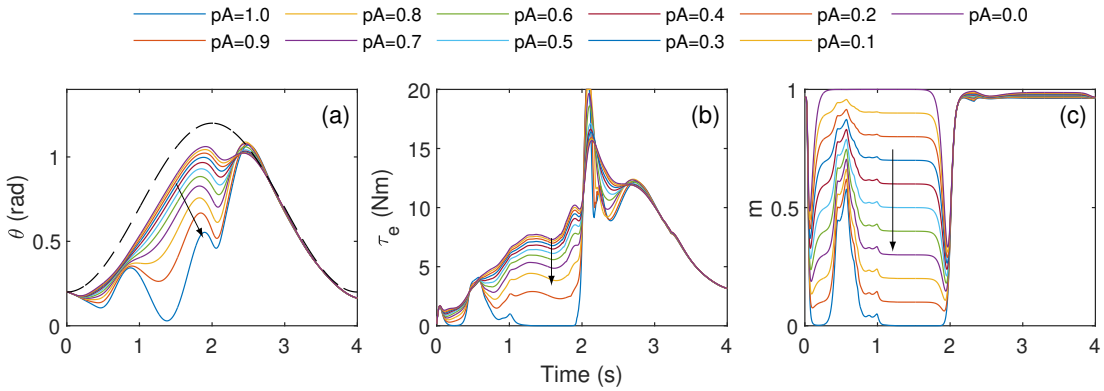


Figure 5.6: Effect of the p_A Parameter. Simulation showing the effect of the p_A parameter on controller behaviour during an assistive human-robot interaction condition. Black arrows indicate the direction of increasing p_A. The dashed line indicates the reference joint position, θ_r.

chosen to represent a suitable maximum level of subject discomfort. Once human resistance reaches the threshold ($\mu_S > R_{th}$), the controller recognises this and halts trajectory progression, giving the subject the opportunity to relax and catch up with the reference trajectory. There are regions where μ_S is above the resistance limit in Figure 5.7c that trigger corresponding regions of trajectory pausing ($\dot{\theta}_r = 0$) in Figure 5.7a.

Wearer safety can be enhanced by using the trajectory adaption method. It can prevent the subject experiencing uncomfortable levels of resistance to the rehabilitation task. It can also prevent the subject falling behind with the task, enabling them to pick up at the reference point where they were last comfortable and benefit more from the session.

5.3.4 Assistance Mode Comparison Study

The effect of scaling benchmark controller gains by the assistance mode output variable, m, was simulated (using the environment described in Section 4.4) for three human interaction conditions: human neutral, human assisting, and human resisting. The output of the FLA was simulated and m used to scale: PID controller gains of proportional, K_p, integral, K_i, and derivative, K_d; impedance controller parameters of

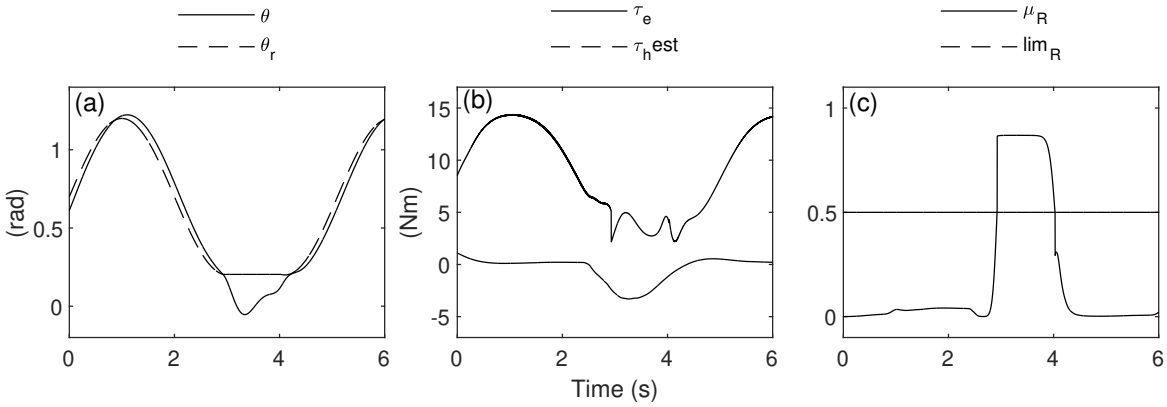


Figure 5.7: Trajectory Adaptation Demonstration. Demonstration of the trajectory adaptation method for the transition phase of a trajectory tracking task. Trajectory adaptation is observed over the region where $\dot{\theta}_r = 0$ between approximately $t = 3s$ and $t = 4s$.

stiffness, K_p, and damping, K_d; and for the AAN-MPC controller, the tracking weight, w_θ, as described in Section 4.3.7. The simulations were configured with ideal human torque estimation. The tracking results of the assistance mode comparison study are shown in Figure 5.8.

For the human neutral condition with $\tau_h = 0$ (first column in Figure 5.8) all of the simulated controllers closely tracked the reference joint position. For the assistive human torque condition (middle column) the assistance mode function increases joint compliance for all controllers. The impedance controller has the greatest compliance, displaying significant tracking error. For the resistive human torque condition (right column) the PID controller shows compliance from 0.5 s, at the onset of human torque, with tracking error as high as 1.0 rad, before rebounding with velocity of up to 8 rad/s at 1.0 s when the assistance mode changes from resistive to passive. The PID controller also demonstrated significant overshoot, after returning to the reference position, to a joint angle of 1.6 rad, potentially causing harmful over-extension of the knee joint. This very sudden and aggressive behaviour is unfavourable and could cause discomfort for the subject. The impedance controller, acted on by the assistance mode function, behaves in a similar way, suddenly snapping at high velocity in response to the transition from resistive to passive human condition. However, the impedance controller does not demonstrate overshoot of the knee joint angle after rebounding. The MPC controller with assistance mode function demonstrates favourable compliance when the

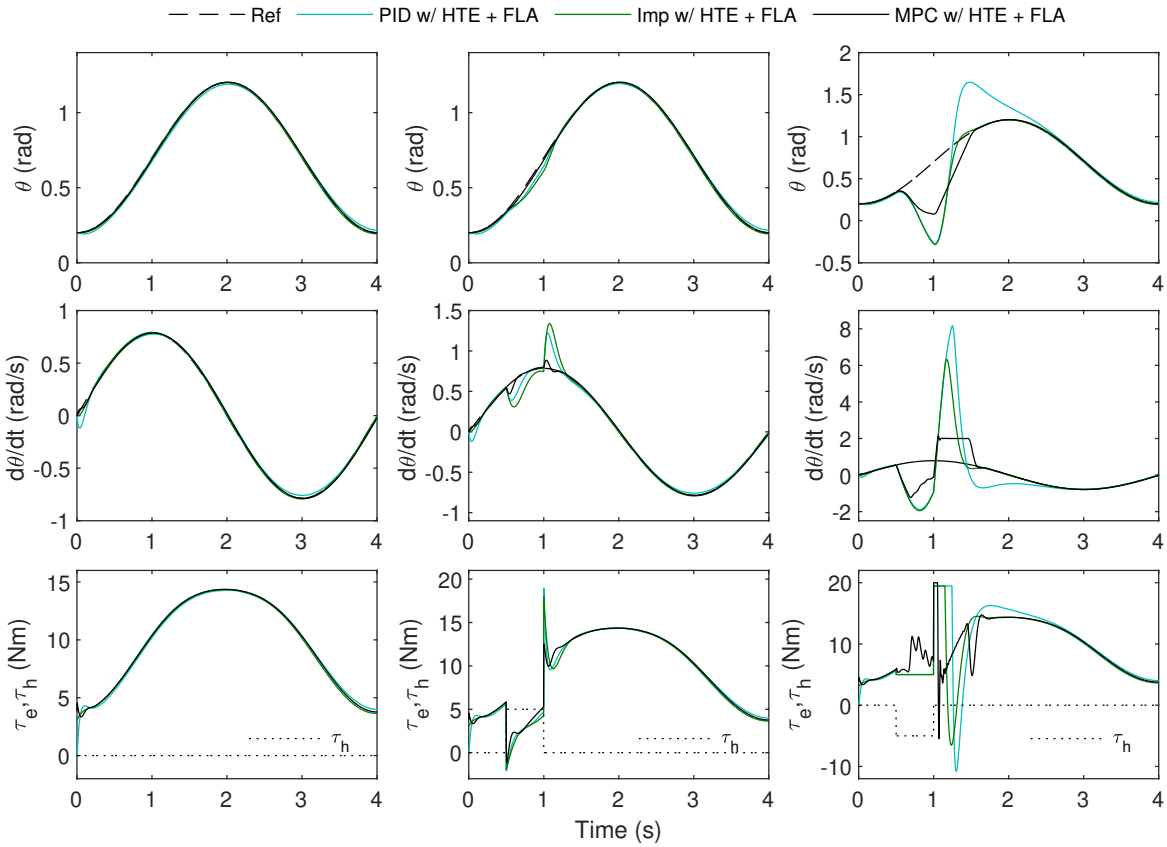


Figure 5.8: Assistance Mode Controller Comparison. Simulation comparing the response during seated tracking in neutral, assisting, and resisting human interaction conditions of the PID, Impedance, and MPC controllers, all with the HTE and FLA elements of the control framework included.

human interaction condition transitions from neutral to resistive at 0.5 s. The MPC contributes corrective torque between 0.5 s and 1.0 s in response to the minimum position constraint of 0 rad being activated. As the human interaction condition transitions from resistive to neutral at 1.0 s, the MPC controller corrects the joint angle error safely in a controlled fashion, as the configured maximum velocity constraint of 2 rad/s is activated and limits the rebound response. Tracking error returns steadily to a low value at around 1.5 s. The favourable behaviour of the MPC controller combined with the FLA output demonstrates the benefit of MPC over PID or impedance control within the proposed control framework.

Chapter Summary

In this chapter, a fuzzy logic-based algorithm was developed for evaluating the assistance mode. This is used within the AAN-MPC control framework to adapt the nature of human-robot interaction. The method uses inputs of estimated human joint torque, $\hat{\tau}_h$ (method in Chapter 6), and reference angular velocity, $\dot{\theta}_r$, to evaluate human-robot interaction on-the fly. The beginning of this chapter provided background information about previous usage of fuzzy logic within AAN control. The materials and methods section described development of the fuzzy logic-based algorithm, the functionality of the p_A parameter, and a trajectory adaptation method, which uses high-level human-robot interaction information from the algorithm to adapt the reference trajectory. In the results and discussion section, algorithm performance for a number of test cases was evaluated, the effect of the p_A parameter value was explored, the trajectory adaptation method was demonstrated, and a comparison study was simulated to compare using the assistance mode function output, m, for PID, impedance, and MPC controllers.

Chapter 6

Estimating Human Joint Torque

In the previous chapter the Fuzzy Logic Algorithm (FLA) component of the proposed AAN-MPC control framework was developed and demonstrated for a representative set of human interaction conditions. It uses two inputs to evaluate the current mode of human-robot interaction: reference angular velocity, and estimated human joint torque. The latter is also used in the AAN-MPC model and cost function to optimise the control input by considering human torque contribution. In this chapter, the method for estimating (and theoretically predicting) human joint torque is developed and evaluated for number of different input sets and modelling techniques.

This chapter begins with a background section, providing information about previous usage of muscle activity signals for estimating human muscle force and joint torque. It also considers existing assumptions in the models used to relate electrical muscle activity and force output. In the following section, the methods and materials involved in recording Mechanomyography (MMG) and Electromyography (EMG) muscle activity signals and the exoskeleton system states is provided. The experimental procedure and exercises used for recording human joint torque estimation data are presented, and modelling approaches are explored. Finally, the results of estimating the human joint torque, for a variety of input datasets and

modelling approaches, are presented and discussed.

6.1 Background

Muscle forces cannot be calculated directly and can only be measured precisely with invasive procedures. Therefore, muscle force is often estimated based on surface EMG [175] and MMG [201] measurements. There are a number of problems associated with using surface myography for muscle force estimation. The main issues relate to precision [202] and representativeness [203]. Lack of precision arises directly from the stochastic nature of EMG and MMG signals. They are the summation of a series of randomly occurring polyphasic motor unit potentials and motor unit contractions respectively [204]. This results in random constructive and destructive superimpositions. Representativeness is an issue because of the structural and functional heterogeneity of muscles across the human population [205]. Despite the issues surrounding surface myography, it is possible, with appropriate recording, signal processing and modelling techniques, to achieve reasonably accurate and representative estimates of isometric and isotonic (dynamic) muscle force and joint output torque using muscle activity signals.

The force output of a single motor unit is regulated by its firing rate. The motor unit produces a contraction, or twitch, after a delay caused by the process of excitation-contraction coupling. The twitch has a duration much longer than the duration of the motor unit potential, and due to this duration, twitches will overlap and sum at the lowest motor unit firing rates, and this causes the functionally relevant increase in force with observed motor unit potential. The correlation between output force and motor unit potential saturates at around 30-40 pulses per second [206]. It is thought that firing rates above this frequency are used to control rate of force, rather than magnitude of force. As all increases in firing rates are reflected in the measured EMG signal, force estimation accuracy in compromised as most models assume a direct (albeit

time delayed) relationship between EMG and muscle force. An exception to this is presented in [207] with a forward dynamic model of the wrist taking muscle activation dynamics into account. Overshoot of force prediction from EMG is often observed at the onset of contraction [208]. In the analysis of the relationship between muscle force, human joint torque, and muscle activity signals, often it is not the actual muscle force being observed, but the net output of a series of synergistic antagonistic muscles. Also, effects of gravity and joint stiffness are usually ignored in models despite being substantial [207].

Biomechanical human muscle force models have been proposed and implemented with some success, such as those in [175] and [208], however the approaches are based on assumptions about conventional biomechanics of the human body that cannot necessary be applied to subjects with limited muscle function or abnormal gait.

6.2 Materials and Methods

An important feature of the AAN-MPC control approach is the utilisation of muscle activity signals and system states for estimation of human knee joint torque. In this section, a number of human torque estimation models are developed which use the inputs of user muscle activity signals and current system states to fit a function to measure human knee joint torque. Data is recorded during representative isometric and dynamic knee joint exercise which captures the full range of motion and frequency of typical knee joint movement during daily activities. The measured human knee joint torque, active thigh segment muscle activity, and recorded system states are used to develop a predictive estimator (or predictor) of human knee joint torque.

6.2.1 Muscle Activity Signals

There are a number of non-invasive ways to measure human muscle activity that are suitable for real-time implementation in robotic control systems. The most common methods, and those investigated in this study, are surface MMG and EMG. Human torque estimation experiments were conducted with both types of signal, and the most effective is used for estimating human joint torque in the AAN-MPC control framework evaluation in Chapter 7.

6.2.1.1 Mechanomyography

MMG is the mechanical signal observable at the surface of a muscle when the muscle is contracted. The onset of muscle contraction causes gross changes in the muscle shape and a large peak in the MMG. Subsequent vibrations are due to oscillations of the muscle fibres at the resonant frequency of the muscle. The MMG signal is low frequency (2-200 Hz) [201] and can be measured using an accelerometer or microphone placed on the skin over the belly of the muscle. When using a microphone it may be termed an acoustic MMG.

6.2.1.2 Electromyography

EMG involves detection of an electromyogram and is an electrodiagnostic medicine technique for evaluating and recording the electrical activity produced by skeletal muscles. EMG signals are recordings of the electrical potential generated by muscle cells when they are electrically or neurologically activated. EMG signals are commonly used in control strategies for prosthetic devices and rehabilitation robots [33], [182], [209]–[211]. There are two kinds of EMG: surface EMG and intramuscular EMG. The latter is performed using recording electrodes, often in the form of a fine needles, which penetrate the surface of the skin

and are therefore not appropriate for a clinical rehabilitation or mobility assistance setting. Surface EMG assesses muscle activity from the skin surface above the muscle. It provides a relatively limited assessment of muscle activation and is usually recorded by a pair of electrodes, but can be used in complex arrays of many electrodes to build a dimensional mapping of muscle activity. More than one electrode is required because recordings display the potential difference (voltage) between electrodes. Surface EMG and acoustic MMG are limited to recording only from superficial muscles close to the skin surface. For lower limb rehabilitation and mobility assistance, the most important muscles are available. Signal quality can be influenced by the depth of subcutaneous tissue at the site of recording, which can be highly variable and depends on the weight of a subject. Surface EMG is the most widely used method of measuring muscle activity [201].

6.2.1.3 Feature Comparison: MMG & EMG

Both MMG and EMG signals can be used to indicate muscle activity. It is necessary to compare and contrast them when considering which is most suitable as a control signal for robotic assistance purposes and both have their relative merits. To summarise, MMG is less expensive, simpler and easier to use as it does not require application of conductive gel and is less sensitive to variations in skin contact and impedance. MMG sensors can be used multiple times without replacement because they do not require adhesive or application of conductive gel. However, MMG suffers from significant cross-talk when used in close array formations and is much more sensitive to movement artefacts and external vibrations. EMG is more costly and takes more time and effort to set up. Dependence on conductive gel or disposable adhesive electrodes causes inconvenience and incurs recurring costs. It also has a low signal-to-noise ratio and the signal requires more filtering stages before is it ready to use. However, if set up properly and good skin contact is achieved, EMG signals are much more reliable and will provide signals that are

more accurate and representative of muscle activity, particularly in the presence of external sources of vibration.

6.2.1.4 Sensor Positioning

Placement of muscle activity sensors is an important consideration when using muscle activity signals for estimating human joint torque. Optimal sensor positioning is dependent on the specific task and some regions of the body may be unavailable due to physical barriers such as braces securing an assistive device to the subject. The experiments and analysis in this thesis focus on actuation and interaction at the knee joint. For human torque estimation at the knee joint, muscle groups were identified in the thigh segment which are responsible for knee joint extension and flexion.

From the quadriceps group of muscles on the front of the thigh, the Vastus Medialis (VM) and Rectus Femoris (RF) muscles are targeted. The VM is an extensor muscle located medially in the thigh. It is involved in knee extension and also contributes to correct tracking of the patella. The RF is situated in the middle of the front of the thigh. Its functions are to flex the thigh and to extend the leg at the knee joint [212].

From the hamstring group of muscles at the back of the thigh, the Biceps Femoris (BF) and Semitendinosus (SE) muscles are targeted. The BF is located to the back of the thigh. As its name suggests, it has two parts, one of which (the long head) forms part of the hamstring. Both heads of the BF perform knee flexion. Since the long head originates in the pelvis it is also involved in hip extension. The long head is a weaker knee flexor when the hip is extended. For this reason, it is important that the subject sits upright during the knee extension and flexion exercise, and utility of this muscle activity signal will likely be reduced for standing exercise. The SE is a long superficial muscle in the back of the thigh. It lies posteromedially[1] in

[1]Posteromedial means located on or near the dorsal (back) midline of the body or a body part.

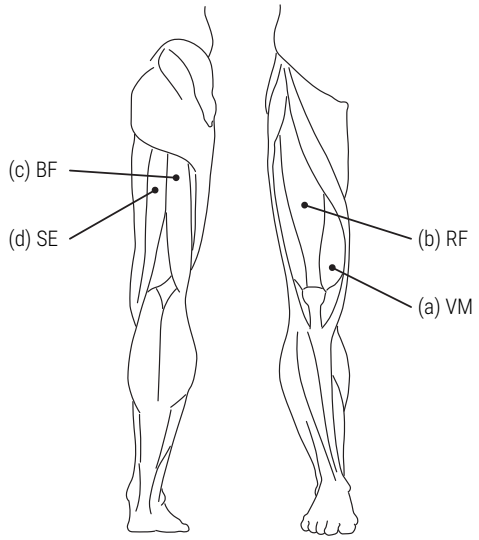


Figure 6.1: Human Thigh Segment Muscles. Illustration showing muscles of the thigh segment of the human leg. Muscles targeted for muscle activity sensing are highlighted: a) Vastus Medialis (VM), b) Rectus Femoris (RF), c) Biceps Femoris (BF), and d) Semitendinosus (SE).

the thigh. The SE is one of three hamstring muscles that work collectively to flex the knee and extend the hip [213].

These four (VM, RF, BF, SE) are the most dominant of the superficial muscles during knee extension and flexion. It is proposed that muscle activity signals from these muscles will be most effective for predicting knee joint torque. All four superficial muscles are close to the skin surface and conveniently positioned for directly sensing muscle activity at the muscle belly. Location of the muscles is shown in Figure 6.1.

6.2.1.5 Signal Measurement and Processing

MMG The MMG sensors used to record muscle activity in this study consist of a diaphragm covering a sealed chamber with dimensions which capture low-frequency vibrations [201]. The device is placed on the skin above a superficial muscle to detect low-frequency vibrations caused by lateral muscle contractions. The nominal frequency is 25 Hz and a large proportion of the signal power exists below 50 Hz. A small microphone sensor detects pressure changes from the base of the chamber which is covered with

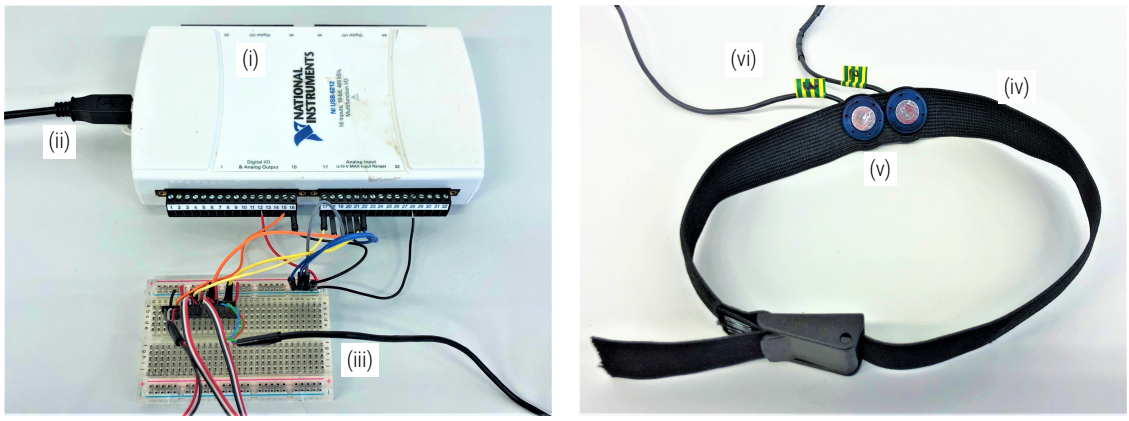
a piece of Mylar. The material is aluminium-coated to avoid allergic risks. The changes in pressure are caused by disturbance of the membrane. The vibrations propagate through the membrane and create a pressure difference in the chamber [214]. The chamber has a height of 5 mm and diameter of 7 mm. This was designed in [201] to produce maximum gain at the resonant frequency and reduce high-frequency distortions. The sensors are attached to a triaxial cable, which carries the ground, power, and signal lines. They are powered by low voltage direct current at 3.3 V. An analog signal varying between 0 and 1.65 V is produced by the microphone and processed by an Analog-to-Digital Converter (ADC).

The unprocessed analog MMG signals are read using a National Instruments DAQ (USB-6212), shown in Figure 6.2, which is a multi-function device capable of simultaneous analog input/output and digital input/output. It has the capacity to read up to 16 16-bit analog signals, with a maximum total sampling rate of 400 kS/s and a maximum update rate of 250 kS/s (i.e. maximum sampling rate of 100 kHz with 4 signals, and maximum rate of 250 kS/s for any single signal). The device has two analog output ports, which can be configured independently to provide constant or varying voltages updated at a maximum rate of 250 kHz. Multiple sensors can be powered using one of the analog output ports set to a constant voltage, so they do not require a separate power supply. The DAQ is connected to a PC by USB and can be controlled using APIs in C/C++ and MATLAB. The analog MMG signals are processed to remove drift and noise. The processing steps are as follows:

1. The signals are high-pass filtered with 10 Hz cutoff frequency to remove drift from the signal and eliminate low frequency noise below the dominant MMG signal bandwidth – HP1.

2. The signals are low-pass filtered with 100 Hz cutoff frequency to isolate the dominant MMG signal bandwidth and remove high frequency noise – LP1.

3. Full-wave rectification provides conditioned signals which are entirely positive, ensuring they do not average to zero and that signal energy is conserved.



(a) National Instruments DAQ Setup. **(b)** MMG Sensor Band for BF and SE Muscles.

Figure 6.2: MMG Sensor Setup. Setup for MMG measurement which shows: i) data acquisition device (NI DAQ), ii) USB power supply and PC interface, iii) sensor interface breadboard, iv) elasticated sensor band (one of three), v) MMG sensor housings, and vi) MMG sensor cables.

4. Finally, a low-pass filter with 2 Hz cutoff frequency is applied to create signal envelopes, yielding measures of muscle activation as positive continuous values – LP2 [215].

EMG The bioelectric activity inside muscles can be detected using surface EMG electrodes. They provide a non-invasive technique for measurement of the muscle activity signal. For the experiments in this study, Covidien pre-gelled disposable EMG electrodes are used (type SG124) that have sticky conductive gel on one side and a pop button on the other side. Silver-silver chloride (Ag/AgCl) composite makes up the metallic part of the electrode. The AgCl layer allows current from the muscle to pass freely across the junction. The gelled electrolytic substance acts as an interface between skin and electrode. Oxidation and reduction reactions take place at the metal electrode junctions, leading to a difference in electrical potential. Skin preparation involves clearing the local area of any dead cells, cleaning with alcohol, and removing hair around the electrode site (if the volume is significant and could affect signal quality). When applying the electrodes, there should be no remaining moisture on the skin.

For each muscle activity measurement, two electrodes measure the signal – one for ground and the other to measure the potential difference. The electrodes are precisely positioned on the skin over the muscle

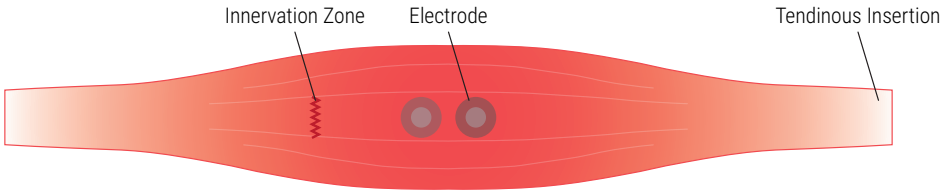


Figure 6.3: EMG Electrode Positioning. Ideal positioning of the two electrode detecting surfaces between the innervation zone (or motor unit) and the tendinous insertion of the muscle.

using a systematic method based on anatomical landmarks outlined in [216]. The detecting surfaces are arranged in a bipolar configuration for the four target muscles in the thigh. For the best possible signal electrodes should be close together (distance of 1 to 2 cm between detecting surface centres) between the motor unit and tendinous insertion of the muscle. The axis of the poles should be parallel to the muscle fibres [217]. The belly of the muscle is a good location as muscle fibre density is the highest. Ideal positioning is illustrated in Figure 6.3. With the electrodes arranged in this way, the detecting surfaces intersect a greater proportion of the same muscle fibres and superimposed activation results in improved signal quality.

In addition to the electrodes used to measure each muscle activity, a subject ground electrode (G) is used as a global reference potential. This is placed on a part of the body close to the EMG measurements that has no bioelectric activity. It can be placed anywhere that is not over a muscle and is on the side of the knee for these experiments. Electrode positioning for the four thigh muscles and ground is shown in Figure 6.4.

The signals are recorded and amplified using a TMSi Porti amplifier, shown in Figure 6.5. The device uses proprietary signal processing technology to provide signals free of mains interference and cable movement artefacts. The device records the minute electrical potential differences generated during muscle activity, amplifies the signals, and converts the values from analog to digital. The “raw” digital signals, sampled at 2048 Hz, are not suitable for human joint torque estimation so are further processed. The processing

(a) Hamstrings. **(b)** Quadriceps.

Figure 6.4: EMG Electrode Placement. Photographs showing the EMG electrode placement for recording muscle activity on the front (quadriceps) and rear (hamstrings) of the thigh segment: i) Vastus Medialis (VM), ii) RRectus Femoris (RF), iii) Biceps Femoris (BF), iv) Semitendinosus (SE), and v) Ground – G.

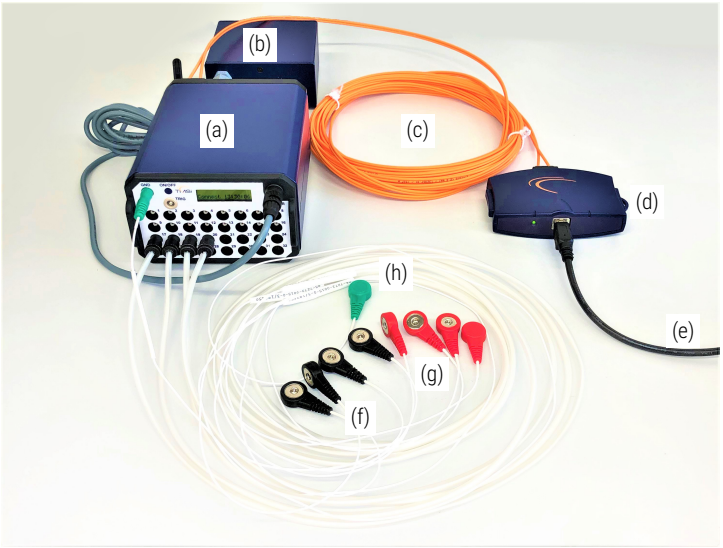


Figure 6.5: EMG Sensor Setup. Setup for EMG measurement which includes: a) data acquisition device (TMSi Porti amplifier), b) mains power supply, c) optical fibre, d) fibre-USB interface module, e) USB cable (to PC), f) reference electrodes (black), g) measurement electrodes (red), h) subject ground electrode (green).

stages from amplified EMG signals to clean measures of muscle activation are as follows:

1. The signals are high-pass filtered with 10 Hz cutoff frequency to remove drift from the signal and eliminate low frequency noise below the dominant EMG signal bandwidth – HP1.
2. The signals are low-pass filtered with 500 Hz cutoff frequency to isolate the dominant EMG signal bandwidth and remove high frequency noise – LP1.
3. A low magnitude deadband is applied to the signals to remove low level stationary noise.
4. The signals are clipped above the largest reasonable magnitude for an EMG signal to negate sizeable motion artefacts.
5. Next, full-wave rectification provides conditioned signals which are entirely positive, ensuring they do not average to zero and that signal energy is conserved.
6. Finally, a low-pass filter with 2 Hz cutoff frequency is applied to create signal envelopes, yielding measures of muscle activation as positive continuous values – LP2 [215].

Filter Design The low-pass and high-pass filters used for processing the MMG and EMG muscle activity signals are designed to provide effective filtering with good noise attenuation and phase characteristics while being efficient for real-time implementation at high sampling rates. Butterworth filters are chosen as they are designed to have a frequency response as flat as possible in the passband. This is important for avoiding ripple and conserving natural behaviour over the full bandwidth of the muscle activity signals. The reported estimates of these bandwidths vary, but there is a general consensus that the majority of muscle activity signal power occurs between 10 Hz and 500 Hz [218] for EMG and between 10 and 50 Hz for MMG.

Because ideal filters cannot be made, compromise is necessary in the approach to finding the desired characteristics. EMG signals are observed between 50 ms and 200 ms prior to muscle force output [189], and MMG signals have been found to precede EMG signals [201]. Therefore, muscle activity signals can theoretically predict human joint torque, so the introduction of phase delay should be avoided as much as possible. For this robotic assistance application, it is important to quantify and reduce signal delays, while strong attenuation of noise is less important [219]. To achieve a more favourable flat phase response, low-order ($n = 2$) filters were used. Low-order Butterworth filters are not computationally expensive and require little cascading so are suitable for real-time implementation. However, they have much less favourable frequency response with decreased rolloff steepness around the cutoff frequency.

The transfer function for a digital second-order ($n = 2$) Butterworth filter in discrete-time representation is expressed in terms of b and a as

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_1 + b_2 z^{-1} + b_3 z^{-2}}{a_1 + a_2 z^{-1} + a_3 z^{-2}}. \tag{6.1}$$

The parameters (Table B.2) and C++ code for the filter stages, for both MMG and EMG, including the transfer function coefficients are provided in Appendix B.5. Processing of MMG and EMG muscle activity signals is displayed over various filtering stages in Figures 6.6 and 6.7 respectively. The effect of processing is illustrated for example MMG and EMG signals measured from the BF and VM muscles of a healthy subject, respectively. The final signal envelopes are positive, smooth, and continuous measures of muscle activity which are suitable inputs for human joint torque estimation models.

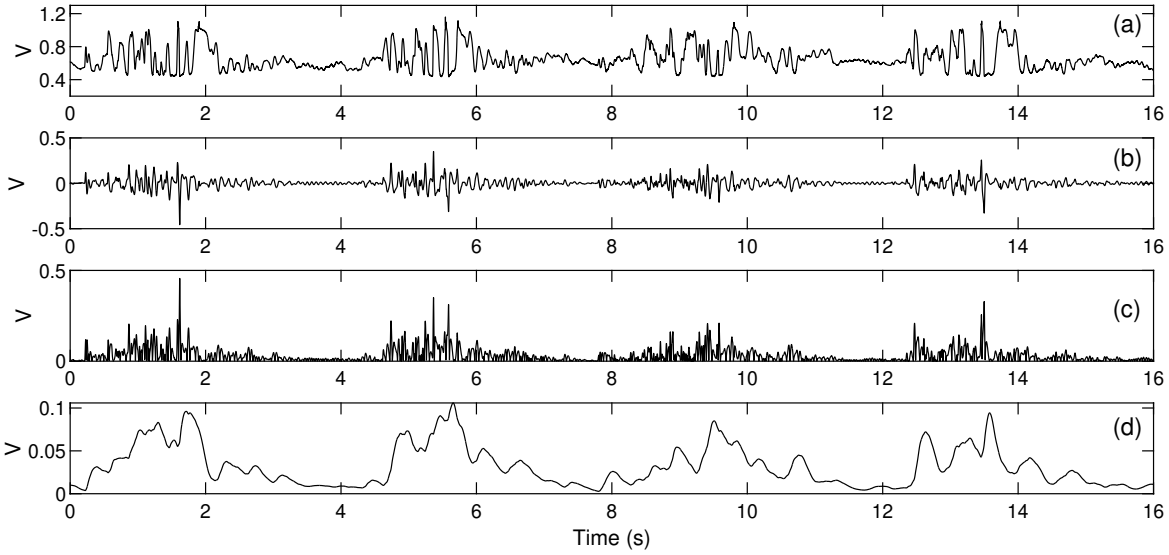


Figure 6.6: MMG Signal Processing Stages. Demonstration of MMG signal processing for contractions of the RF muscle: a) raw microphone signal, b) after steps 1 - 2: HP1 and LP1, c) after full-wave rectification, and d) after signal enveloping using LP2.

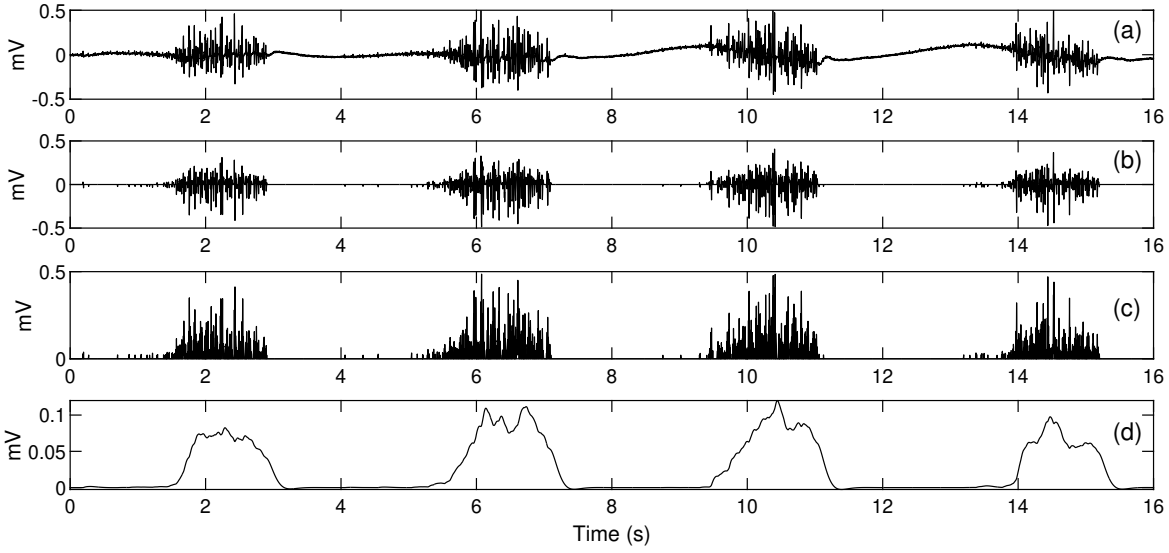


Figure 6.7: EMG Signal Processing Stages. Demonstration of EMG signal processing for contractions of the VM muscle: a) raw amplified signal, b) after stages 1 - 4: HP1, LP1, deadzone filtering, and clipping, c) after full-wave rectification, and d) after signal enveloping using LP2.

6.2.2 Experimental Procedure

For the exercise tasks of upright walking and seated leg extension and flexion, the proposed AAN-MPC control framework requires an effective model for human torque estimation. To build the model, the dependency of human joint torque on muscle activity and system states was determined. Muscles surrounding the knee joint that contribute to extension and flexion were identified and targeted for MMG and EMG sensing. Data capturing information from the knee joint and surrounding area was recorded during experimental procedures that represent the relevant exercise tasks and human involvement conditions involved in lower-limb robotic assistance.

An important consideration for the experimental protocol is human involvement. A model can capture the relationship between muscle activity, system states, and human joint torque for any human involvement condition that data is specifically recorded for. Data should be recorded during all the major types of human-robot interaction to ensure the model can be effective in all conditions. This should include human relaxation, human torque generation with the robot, and human torque generation against the robot. The range of the recorded human joint torque should span the full range of torque that could be observed during robot-assisted exercise. It is also necessary to capture frequency-dependent characteristics by recording data during exercise at various repetition frequencies. The same is true for joint position, so the experimental protocol should include exercises which span the full range of joint motion.

Joint torque is measured at a particular joint to build a model. It is not within the scope of this work to model human joint torque at the ankle or hip, as the exoskeleton configurations described in Section 4.2 do not include the actuation at those joints. A limitation of this method is the consideration of joint torques produced only in the main rotational degree of freedom at a particular joint – flexion and extension of the knee in this case.

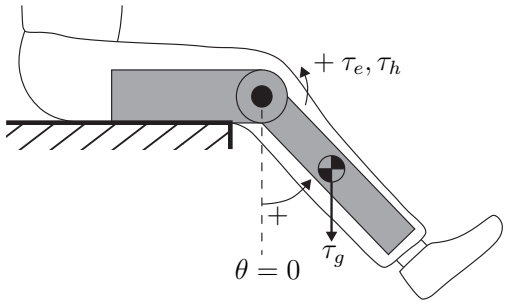


Figure 6.8: Seated Human-Exoskeleton System. Schematic of the human-exoskeleton system in the seated position used for the human joint torque estimation exercises showing joint angle, θ, exoskeleton torque, τ_e, human torque, τ_h, and gravitational torque, τ_g, of the shank/foot segment.

The human knee joint torque estimation data was recorded during isometric and dynamic (isotonic) exercises. The joint conditions were representative of robot-assisted exercise in terms of human involvement, range of motion, and movement frequency. Major superficial muscles active in the thigh during knee extension and flexion were targeted for muscle activity measurement, as described in Section 6.2.1.4. The recorded datasets include measured human joint torque, τ_h, muscle activity, ch_i, and the system states of joint position and velocity, $[\,\theta; \dot{\theta}]$.

6.2.2.1 Knee Joint Torque Estimation Exercises

The knee joint torque estimation data was recorded during seated exercises. The seated test exoskeleton configuration described in Section 4.2.2 was used. The arrangement is illustrated in Figure 6.8, indicating the direction and origin of joint angle, human torque, and exoskeleton torque. Two exercises were carried out for human torque estimation data collection: 1) isometric contraction and 2) dynamic extension and flexion.

Exercise 1: Isometric Contraction This involved the subject contracting their knee joint in both directions (extension and flexion) in different fixed positions spread over a range of motion from 0.2 rad to 1.2 rad. This exercise was recorded in different knee joint positions to fully capture the relationship between

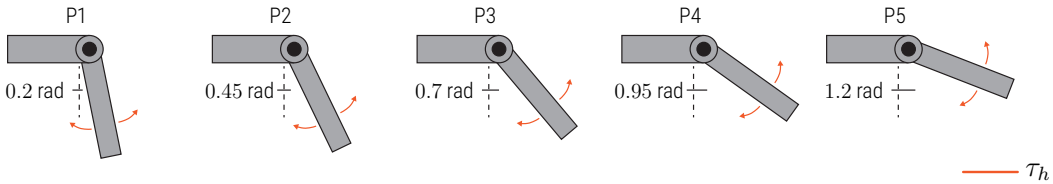


Figure 6.9: Isometric Contraction Positions. Illustration showing the set of positions, P, of constant reference joint angles, θ_r. Human joint torque, τ_h, is recorded in the five positions for human involvement conditions of Isometric Neutral (IN), Isometric Extension (IE), and Isometric Flexion (IF) during exercise 1.

muscle activity, system states (i.e. joint angle), and human joint torque. Isometric contraction exercises are well studied in the literature [201], [212], [220], and are useful for characterising and comparing muscle activity using MMG and EMG signals. The positions, P, of the constant reference joint angles used for collecting data during isometric contraction exercises were equidistant across the range of motion with spacing of 0.25 rad. This gave an array of $\mathrm{P} = \{0.2, 0.45, 0.7, 0.95, 1.2\}$ rad, as shown in Figure 6.9. For each position the knee joint was held in a fixed position by the robot, while data was recorded during three instructed human involvement conditions:

- Isometric Neutral (IN): Subject relaxes, contributing minimal torque,
- Isometric Extension (IE): Subject contributes torque in the direction of leg extension,
- Isometric Flexion (IF): Subject contributes torque in the direction of leg flexion.

For each trial, the robot closely tracked a constant reference position command, P, using position control. High gain settings minimised tracking error, rejecting disturbance to hold the joint in an isometric state regardless of human torque or external factors such as gravitational acceleration. The control law is given in section 6.2.2.2 and shown in Figure 6.13.

The variable of human joint torque cannot be measured directly so was inferred by comparing the IE and IF exoskeleton torque τ_e profiles with those for the IN condition at each position P. The inference method is described for the dynamic extension and flexion exercise in Section 6.2.2.1.

Exercise 2: Dynamic Extension and Flexion This exercise involved movement along smooth sinusoidal joint angle trajectories of constant amplitude and various constant frequencies, comfortably within knee joint range of motion. Knee joint extension-flexion exercises are well established for rehabilitation purposes and have been studied previously [221], [222]. The human-exoskeleton shank segment was position controlled with high gain. As with exercise 1, minimal tracking error was allowed despite the presence of disturbance from human involvement and gravitational acceleration.

The reference joint angle trajectories were time-dependent and sinusoidal with various frequencies, defined by

$$\theta_r(t) = \frac{\cos(2\pi f t - \pi)}{2} + 0.7 \tag{6.2}$$

where θ_r is the reference joint angle in radians, f is the repetition frequency in Hz, and t is the time in seconds. The trajectories represent close to full range of motion, starting at 0.2 rad and stretching to 1.2 rad, measured from the neutral position (shank pointing down for $\theta = 0$). The extension and flexion motions are phases illustrated in Figure 6.10. Repetitions frequencies were low ($f = 0.125$ Hz), medium ($f = 0.25$ Hz) and high ($f = 0.5$ Hz), based on examples of gait analysis and robotic rehabilitation [177], [223]. The reference knee joint angle and velocity trajectories, for a duration of 8 s, are shown in Figure 6.11. The sinusoidal frequencies are constant for each trajectory but the joint angular velocities vary over the ranges shown in Table 6.1. The variation ensures there is a set of diverse and representative conditions in the human joint torque estimation dataset.

Table 6.1: Dynamic Extension and Flexion Trajectory Information. Information describing the low, medium, and high frequency trajectories used for the seated knee joint dynamic extension and flexion exercise.

Trajectory	Freq. (Hz)	Period (s)	Velocity Range (+-rad/s)	Motion Range (rad-rad)
Low Freq.	0.125	8.0	+-0.004	0.2-1.2
Medium Freq.	0.25	4.0	+-0.008	
High Freq.	0.5	2.0	+-0.016	

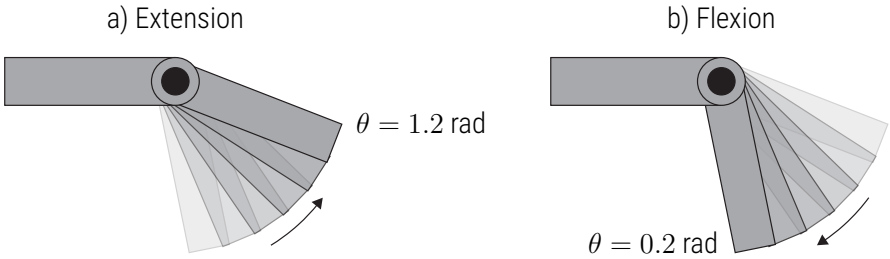


Figure 6.10: Dynamic Knee Joint Exercise Range of Motion. Exercise range of motion illustrated for a) extension and b) flexion phases of exercise 2.

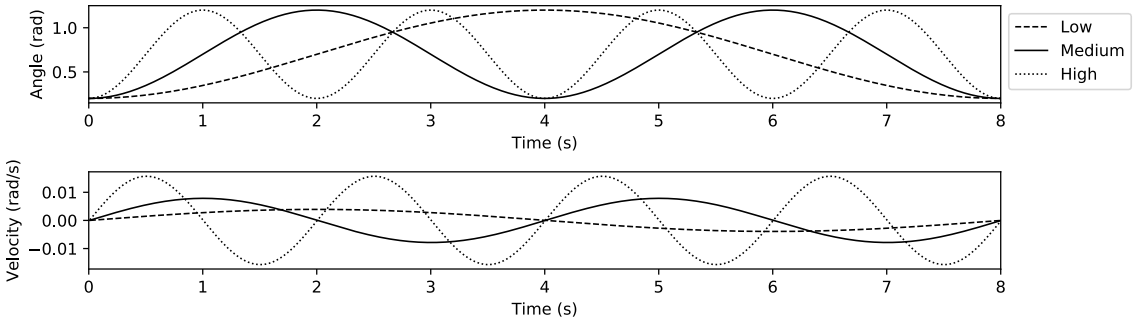


Figure 6.11: Dynamic Extension and Flexion Trajectories. Knee joint angle and velocity reference trajectories for low, medium, and high frequency repetitions of exercise 2: dynamic extension and flexion. During extension the joint angle increases from 0.2 rad to 1.2 rad and during flexion it decreases back to 0.2 rad.

The subject was instructed to contribute torque in the following involvement conditions, as illustrated in Figure 6.12:

- Neutral (N): Subject relaxes during the exercise, contributing minimal torque and allowing the robot to dominate extension and flexion movements,
- Extension Assist (EA): Subject contributes assistive torque during extension and relaxes during flexion,
- Extension Resist (ER): Subject contributes resistive torque during extension and relaxes during flexion,
- Flexion Assist (FA): Subject relaxes during extension and contributes assistive torque during flexion,

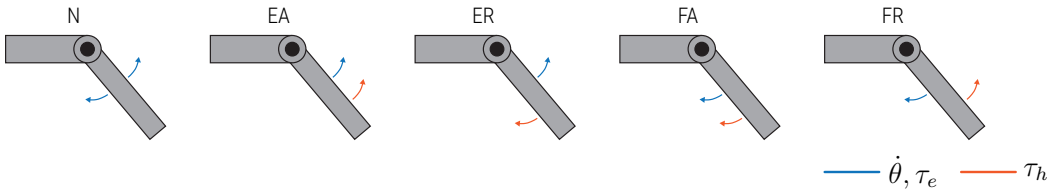


Figure 6.12: Human Involvement Conditions. Illustration showing the directions of joint velocity, $\dot{\theta}$, robot torque, τ_e, and human torque, τ_h, in the five human involvement conditions recorded for human torque estimation exercise 2: dynamic extension and flexion.

- Flexion Resist (FR): Subject relaxes during extension and contributes resistive torque during flexion.

With a variety of involvement conditions, the human joint torque estimation dataset can be representative of all human interaction modes. If data recording was random and improvised without specific instructions, there would be no quantifiable and repeatable way of providing representation of all interaction conditions.

During the trials, only exoskeleton joint torque, τ_e, is measured at the actuator output, so human torque, τ_h, must be inferred. The exoskeleton joint torque for each human involvement condition (C) is compared with the Neutral (N) condition as the reference of zero human torque. Given that total joint torque, τ, for each trial is equal for the same actual trajectory, and $\tau = \tau_e + \tau_h$,

$$\tau_{h,\mathrm{C}} = \tau_{e,\mathrm{N}} - \tau_{e,\mathrm{C}} \tag{6.3}$$

where $\tau_{h,\mathrm{C}}$ is the inferred human joint torque for the involvement condition, $\mathrm{C} = \{\mathrm{EA}, \mathrm{ER}, \mathrm{FA}, \mathrm{FR}\}$. $\tau_{e,\mathrm{N}}$ is the reference exoskeleton torque recorded during the neutral human involvement condition, N (with $\tau_h = 0$), and $\tau_{e,\mathrm{C}}$ is the measured exoskeleton torque during the human involvement condition, C. For each condition, the realised joint angle trajectory is similar with a small margin of error, as shown in Appendix B.1. The control law is provided in Section 6.2.2.2.

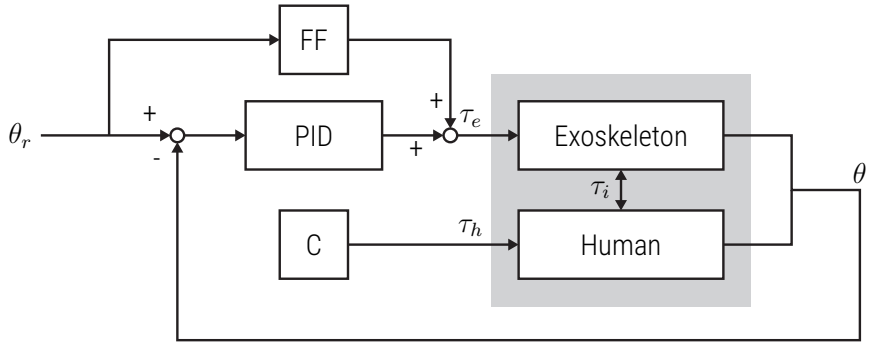


Figure 6.13: Control Loop for Human Torque Estimation Tests. PID control loop with feedforward (FF) for human torque estimation trials. τ_i is the human-exoskeleton interaction torque. C represents the instructed human interaction condition, out of: Neutral (N), Extension Assist (EA), Extension Resist (ER), Flexion Assist (FA), Flexion Resist (FR).

6.2.2.2 Exoskeleton Control Law

PID control with high gain and feedforward ensured that tracking error was minimal (RMSE $<$ 0.1 rad) for both exercises. The controller, illustrated in Figure 6.13, rejected the effects of external disturbance from gravitational acceleration and human joint torque. With close to identical realised trajectories, the torque required to reject the human interaction effects can be inferred. The controller is described with the following control law:

$$\tau_e(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt} + K_{ff}\theta(t) \tag{6.4}$$

where $e(t) = \theta_r(t) - \theta(t)$, $\theta_r(t)$ is the reference joint angle (rad), $\theta(t)$ is the joint angle (rad), $K_p = 150$, $K_i = 200$, $K_d = 4$, and $K_{ff} = \tau_g$ (subject 1 from parametric identification in Section 4.2.4.3). Note that feedforward term $K_{ff}\theta(t) \approx K_{ff}sin(\theta(t))$ for small angles and overcompensates for gravitational torque as $\theta \to \pi/2$. The high gains achieved close tracking for all human involvement conditions, making the total torque, τ, equivalent for all trials.

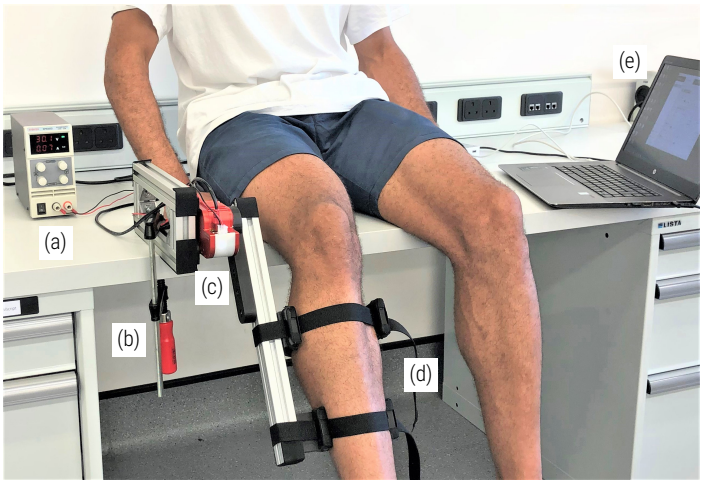


Figure 6.14: Setup for Human Torque Estimation Tests. Experimental setup (not including muscle activity measurement setups) of the single DOF human-exoskeleton system showing (a) power supply, (b) F clamp, (c) HEBI X8-9 actuator, (d) shank segment braces, and (e) computer with GUI.

6.2.2.3 Recording System States and Joint Torque

Joint position, θ, angular velocity, $\dot{\theta}$, and exoskeleton torque, τ, are measured at the knee joint using the HEBI X-Series actuator, which can simultaneously control and/or measure position, velocity, and torque. The actuator is shown in the seated human-exoskeleton configuration in Figure 6.14. It is controlled through an API in C++ that enables real-time system state measurement and joint torque control. Feedback is requested from the actuator and commands sent at a specified rate of 500 Hz.

The subject interacts with the seated exoskeleton configuration via a real-time GUI, which informs the subject of the current exercise, the current position/trajectory and provides instruction of the desired human involvement. The GUI displays a time window of the reference trajectory, giving the subject time to prepare for periods of specified human-robot interaction. The GUI also provides the subject with real-time feedback of signal measurements, as shown in Figure 6.15.

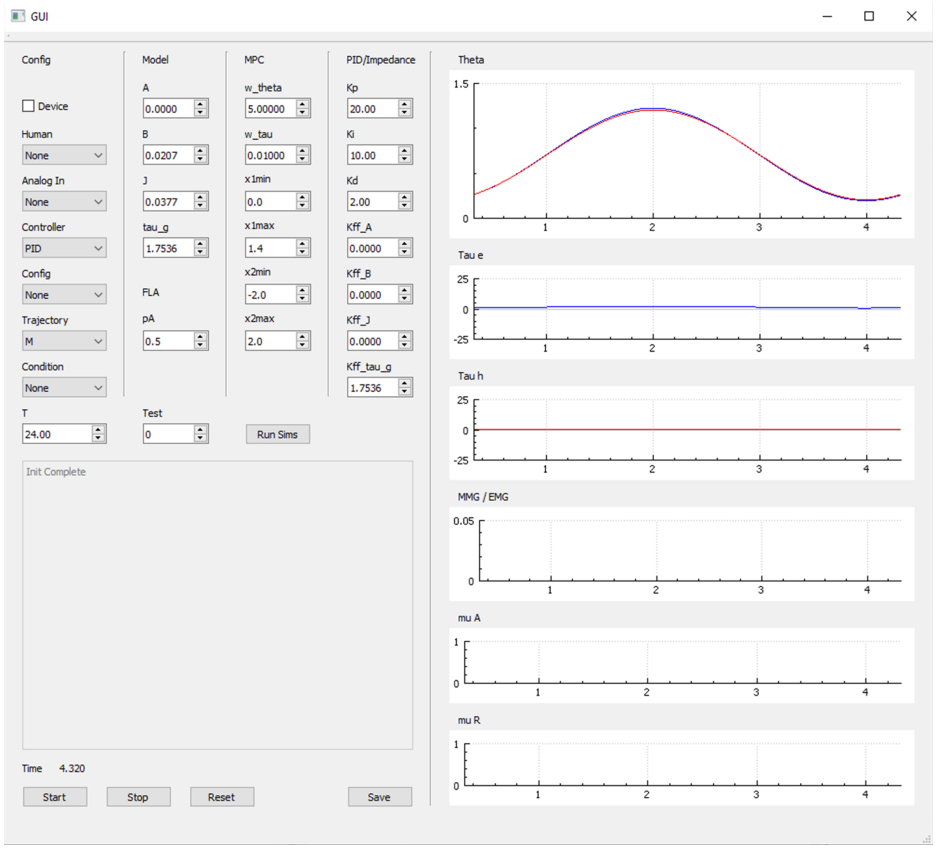


Figure 6.15: Testing Interface. GUI for subject instruction and real-time feedback during exercises.

6.2.2.4 Test Configurations

Data was recorded for each human joint torque estimation exercise in a number of static and dynamic trajectories in each of the instructed human involvement conditions for both MMG and EMG muscle activity signals. For exercise 1 (isometric contraction) five constant position trajectories were used (P1-P5) with three human involvement conditions (Isometric Neutral (IN), Isometric Extension (IE), and Isometric Flexion (IF)) giving 15 configurations. These were recorded for MMG and EMG in tests 1 and 2 respectively, giving a total of 30 configurations for exercise 1. For exercise 2 (dynamic extension and flexion) three trajectories (low, medium, and high) were used with five human involvement conditions (Neutral (N), Extension Assist (EA), Extension Resist (ER), Flexion Assist (FA), and Flexion Resist (FR)) giving 15 configurations. These were recorded for MMG and EMG in tests 3 and 4 respectively, giving a total of 30

configurations exercise 2. A total of 60 configurations were used for recording human torque estimation data over tests 1 to 4. The configurations are summarised in Table 6.2.

6.2.2.5 Data Collection

For the first exercise, data was recorded for a duration of eight seconds in each configuration. This was long enough to provide a meaningful number of samples for each condition, but not long enough for the subject to become fatigued. The eight second recording was repeated three times to increase the number of samples. The repeated trials were not consecutive, instead they were conducted for all of the configurations of that test, then the whole test repeated from the start. This yielded a total duration of 24 s in each configuration. For the second exercise, data was recorded in each configuration over a duration of 24 s in one continuous trial. As the trajectories start and finish in the same position, it was convenient for repetitions to occur consecutively. Repeating the tests made it easier for the subject to become familiar with trajectory timing and better synchronise human involvement with the extension and flexion phases. The datasets were collected over $\mathrm{N} = 60$ periods of 24 s at 500 Hz sampling rate giving 720,000 data points.

The human joint torque estimation data was collected from a single healthy subject: a male of age 27 years, who gave permission for the results to be included in this thesis. The data collection process is repeatable for any healthy subject. For a subject with a neurological disorder, the procedure may need to

Table 6.2: HTE Test Configurations. There are trials for each position/trajectory in each human involvement condition, making a total of 15 configurations for each test.

Test	Exercise	Signal	Positions / Trajectories	Conditions	N	T (s)	Rep's
1	Isometric	MMG	P1, P2, P3, P4, P5	IN, IE, IF	15	8	3
2	Isometric	EMG	P1, P2, P3, P4, P5	IN, IE, IF	15	8	3
3	Dynamic	MMG	Low, Med, High	N, EA, ER, FA, FR	15	24	1
4	Dynamic	EMG	Low, Med, High	N, EA, ER, FA, FR	15	24	1

be adjusted for range of motion and mobility limitations.

6.2.2.6 Input Set Configurations

The human joint torque estimation dataset was split into a number of input sets to evaluate the dependency of model performance on particular input signals. The signals were separated into the following input sets, summarised in Table 6.3: A, containing only the most dominant muscle activity signals from the hamstrings and quadriceps (VM and BF respectively); B, containing VM, BF, and the first joint state (θ); C, containing VM, BF, and both joint states (θ and $\dot{\theta}$); D, containing all muscle activity signals (VM, RF, BF, and SE); E, containing all muscle activity signals and θ; and F, containing all muscle activity signals and both joint states, θ and $\dot{\theta}$. Human joint torque estimation results are evaluated across all of the input sets for both MMG and EMG muscle activities. The entire human joint torque estimation datasets, recorded for tests 1 to 4 are shown in Appendix B.1.

6.2.3 Human Torque Models

Training models for estimating human joint torque in advance involves fitting an estimation function, $h(\cdot)$, which relates the inputs of processed muscle activity signals, $\mathbf{ch}$, and system states, $\boldsymbol{\theta}$, to the time-shifted

Table 6.3: HTE Input Sets. Input sets for human joint torque estimation datasets (n = number of input variables).

	A	B	C	D	E	F
ch_1	✓	✓	✓	✓	✓	✓
ch_2				✓	✓	✓
ch_3	✓	✓	✓	✓	✓	✓
ch_4				✓	✓	✓
θ		✓	✓		✓	✓
$\dot{\theta}$			✓			✓
n	2	3	4	4	5	6

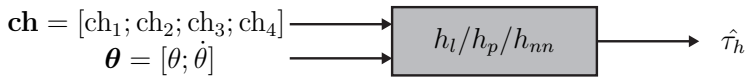


Figure 6.16: Human Joint Torque Estimation Function Block. HTE function example for input set F showing the three estimation models (h_l, h_p, h_{nn}), six function inputs (four processed muscle activity signals, $\mathrm{ch}_1, \mathrm{ch}_2, \mathrm{ch}_3, \mathrm{ch}_4$, and two system states, $\theta, \dot{\theta}$) and output of estimated human joint torque, $\hat{\tau}_h$.

output of human joint torque, τ_h. The estimation function can be written as

$$\hat{\tau}_h(t - t') = h(\mathbf{ch}, \boldsymbol{\theta}, t) \tag{6.5}$$

where t' is a time shift of 100 ms, which approximates the electromechanical delay between observed muscle activity signal and muscle force output [189], [201]. Given the dataset contains input and output variables, building a human joint torque estimator amounts to fitting a function to the data. There are modelling approaches of varying complexity. Three are investigated and compared in this section: linear regression (h_l), polynomial regression (h_p), and a neural network-based method (h_{nn}). The estimation models, function inputs, and function output are illustrated in Figure 6.16.

6.2.3.1 Linear Regression Model

A basic approach to human joint torque estimation is to fit a linear regression model to the data. Linear regression models the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). This was a case of multiple linear regression, as there were a maximum of six explanatory variables: four muscle activity signals, $(\mathrm{ch}_1, \mathrm{ch}_2, \mathrm{ch}_3, \mathrm{ch}_4)$, and two system states, $(\theta, \dot{\theta})$. For this analysis the muscle activity and system state vectors were concatenated to give a single input vector, e.g., for input set F, $\boldsymbol{x} = \{\mathrm{ch}_1, \mathrm{ch}_2, \mathrm{ch}_3, \mathrm{ch}_4, \theta, \dot{\theta}\}$.

Given the dataset for input set F, $\{\tau_{hi}, \mathrm{ch}_{1i}, \mathrm{ch}_{2i}, \mathrm{ch}_{3i}, \mathrm{ch}_{4i}, \theta_i, \dot{\theta}_i\}_{i=1}^{n}$ of n statistical units, a linear regression model assumes that the relationship between the dependent variable τ_h and the vector of regressors X is linear. This is modelled through an error variable ϵ – an observed random variable that adds noise to the linear relationship between the dependent variable and regressors. The model takes the form of

$$\tau_{hi} = \beta_0 + \beta_1 \mathrm{ch}_{1i} + \beta_2 \mathrm{ch}_{2i} + \beta_3 \mathrm{ch}_{3i} + \beta_4 \mathrm{ch}_{4i} + \beta_5 \theta_i + \beta_6 \dot{\theta}_i + \epsilon_i = \boldsymbol{x}_i^T \boldsymbol{\beta} + \boldsymbol{\epsilon}_i, \quad i = 1, ..., n \tag{6.6}$$

where ^{T} denotes the transpose, so that $\boldsymbol{x}_i^T \boldsymbol{\beta}$ is the inner product between vectors $\boldsymbol{x}_i$ and $\boldsymbol{\beta}$. The n equations, representing each sampling point in time, are stacked together and written in matrix notation as

$$\boldsymbol{\tau}_h = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{6.7}$$

where

$$\boldsymbol{\tau}_h = \begin{pmatrix} \tau_{h1} \\ \tau_{h2} \\ \vdots \\ \tau_{hn} \end{pmatrix}, \tag{6.8}$$

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_1^T \\ \boldsymbol{x}_2^T \\ \vdots \\ \boldsymbol{x}_n^T \end{pmatrix} = \begin{pmatrix} 1 & \mathrm{ch}_{11} & \mathrm{ch}_{12} & \mathrm{ch}_{13} & \mathrm{ch}_{14} & \theta_1 & \dot{\theta}_1 \\ 1 & \mathrm{ch}_{21} & \mathrm{ch}_{22} & \mathrm{ch}_{23} & \mathrm{ch}_{24} & \theta_2 & \dot{\theta}_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \mathrm{ch}_{n1} & \mathrm{ch}_{n2} & \mathrm{ch}_{n3} & \mathrm{ch}_{n4} & \theta_n & \dot{\theta}_n \end{pmatrix}, \tag{6.9}$$

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_6 \end{pmatrix}, \quad \boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}. \tag{6.10}$$

A constant, $x_{i0} = 1$, is included as one of the regressors which corresponds to the element β_0 and is called the intercept. The regressor variables can be nonlinear functions of another regressor, such as a

polynomial or trigonometric function. The model remains linear as long as it is linear in the parameter vector $\boldsymbol{\beta}$. The parameter vector includes the intercept value β_0, however, this parameter is optional. $\boldsymbol{\epsilon}$ is a vector of error terms ϵ_i. It captures any other factors that influence the dependent variable τ_h other than the regressors $\boldsymbol{x}$.

Fitting the linear regression model to the human joint torque estimation dataset means estimating the regression coefficients $\boldsymbol{\beta}$ such that the error term $\boldsymbol{\epsilon} = \boldsymbol{\tau}_h - \boldsymbol{X}\boldsymbol{\beta}$ is minimised. It is common to use the sum of squared errors $||\boldsymbol{\epsilon}||$ as the quality of fit. The set of linear equations is solved computationally to find the set of parameters $\boldsymbol{\beta}$ giving the minimum sum of squared error terms using OLS. The resulting estimator h_l, for input set F, can be expressed as

$$\hat{\tau_h} = h_l(\mathbf{ch}, \boldsymbol{\theta}) = \beta_0 + \beta_1 \mathrm{ch}_1 + \beta_2 \mathrm{ch}_2 + \beta_3 \mathrm{ch}_3 + \beta_4 \mathrm{ch}_4 + \beta_5 \theta + \beta_6 \dot{\theta}. \tag{6.11}$$

6.2.3.2 Polynomial Regression Model

The polynomial regression human joint torque model is based on the linear regression model, with additional power-2 terms included in the set of independent variables for each of the original linear variables. This method fits a nonlinear model to the data, however as a statistical estimation problem it is linear, in that the regression function is linear in the unknown parameters that are estimated. The parameters of the quadratic polynomial regression model are also solved using OLS to find the minimum sum of squared errors $||\epsilon||$. The code for fitting the linear and polynomial regression models, used for input sets A to F,

can be found in Appendix B.2. The polynomial estimator h_p, for input set F, can be expressed as:

$$\begin{aligned}\hat{\tau_h} = h_p(\mathbf{ch}, \boldsymbol{\theta}) = \\ \beta_0 + \beta_1 \mathrm{ch}_1 + \beta_2 \mathrm{ch}_2 + \beta_3 \mathrm{ch}_3 + \beta_4 \mathrm{ch}_4 + \beta_5 \theta + \beta_6 \dot{\theta} + \\ \beta_7 \mathrm{ch}_1^2 + \beta_8 \mathrm{ch}_1 \mathrm{ch}_2 + \beta_9 \mathrm{ch}_1 \mathrm{ch}_3 + \ldots + \beta_{12} \mathrm{ch}_1 \dot{\theta} + \\ \beta_{13} \mathrm{ch}_2^2 + \beta_{14} \mathrm{ch}_2 \mathrm{ch}_3 + \ldots + \beta_{17} \mathrm{ch}_2 \dot{\theta} + \\ \beta_{18} \mathrm{ch}_3^2 + \beta_{19} \mathrm{ch}_3 \mathrm{ch}_4 + \ldots + \beta_{21} \mathrm{ch}_3 \dot{\theta} + \\ \beta_{22} \mathrm{ch}_4^2 + \beta_{23} \mathrm{ch}_4 \theta + \beta_{24} \mathrm{ch}_4 \dot{\theta} + \\ \beta_{25} \theta^2 + \beta_{26} \theta \dot{\theta} + \\ \beta_{27} \dot{\theta}^2.\end{aligned} \tag{6.12}$$

6.2.3.3 Neural Network Model

The third human joint torque modelling approach used a basic neural network to fit input set F to the output of time-shifted human joint torque. The neural network was designed with a simple architecture, using relatively few layers and model parameters, for repeatability and convenient real-time implementation at high sampling frequencies. The model architecture is described in Table 6.4 in terms of layer size ($s = 8$) and number of inputs ($n = 6$ for input set F). The sequential model contains two densely connected hidden layers and an output layer that returns a single discrete value. The model uses Rectified Linear Unit (ReLU) activation functions for computational simplicity and linear behaviour so the gradients remain proportional to the node activations [224].

The neural network was developed and trained using the open-source TensorFlow machine learning platform. The dataset was split randomly into a training dataset and test dataset with ratio of 0.8 to 0.2. The test dataset was used to evaluate the model performance. The input data values were normalised prior to

training by subtracting the mean for each variable and dividing by the standard deviation for the variable. The evaluated statistics of mean and standard deviation were saved to be applied for evaluating the test set and later implementing the model in real-time.

The model was built initially with random untrained parameter values. These parameters (137 in total) were trained by gradually optimising the parameters to learn the dataset and make predictions about the unseen data, using the RMSprop optimisation algorithm from the TensorFlow library. The training dataset is further split into training and validation datasets with a ratio of 0.8 to 0.2. Over the course of training, the training and validation accuracy metrics are stored in a *history* object. The model is trained indefinitely until the validation dataset is seen to no longer show improvement in the validation loss metric of MSE. An early-stopping callback is used with a patience value of 10, meaning that if the validation loss does not improve for 10 training epochs, the model training is automatically stopped. The model training required a total of 97 epochs. The model training performance for the metric of MAE is shown in Figure 6.17. The code used for building and training the neural network model can be found in Appendix B.3.

6.3 Results and Discussion

In this section, the results of human joint torque estimation modelling are provided. The relationship between the inputs of processed muscle activity signals and system states, and the output of future human

Table 6.4: Neural Network Model Architecture. Model architecture including: Input Dimensions (ID); Output Dimensions (OD); Activation Function (AF). Dimensions are provided in terms of layer size ($s = 8$) and number of inputs ($n = 6$).

Layer	ID	OD	AF	Num. Params
Input	[1,n]	-	-	-
Dense 1	[1,s]	[1,s]	ReLU	s(n+1)
Dense 2	[1,s]	[1,s]	ReLU	s(s+1)
Output	-	[1,1]	-	s+1
			Total	s(n+s+3)+1

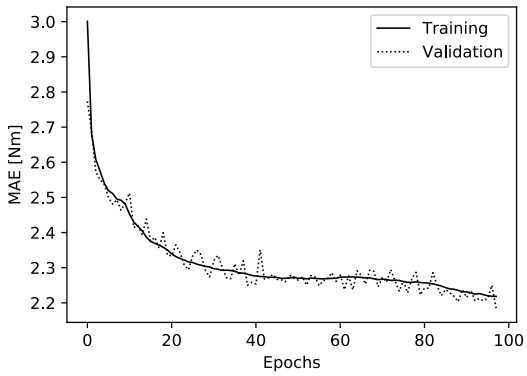


Figure 6.17: Model Learning Curve. Neural network training performance showing the metric of mean absolute error for the training and validation subsets.

joint torque was investigated for different choices of muscle activity signals, input sets, and modelling approaches. The efficacy of various approaches is evaluated and compared.

6.3.1 MMG and EMG Input Comparison

To compare the effectiveness of using MMG and EMG muscle activity signals for predicting human torque, the linear and polynomial regression modelling approaches, described in Section 6.2.3.1 and Section 6.2.3.2, were applied to the MMG and EMG datasets (Appendix B.1) across input sets A to F (Table 6.3). The results are also compared for the different sets of exercises described in Section 6.2.2.

The EMG dataset outperforms the MMG dataset, with significantly lower MAE between reference and estimated human joint torque for the linear and polynomial regression models, as shown in Figure 6.18 and Appendix B.4. The MMG dataset shows much less correlation with human joint torque during isometric contraction (P1 to P5) than during dynamic extension and flexion (M, L, and H). The high modelling error using the MMG dataset increased for isometric contraction from close to 3 Nm to around 7 Nm as the knee joint angle increases from 0.2 rad in P1 to 1.2 rad in P5. This trend is present across all input sets. Increasing error could be due to vibrations associated with the greater robot torque required to maintain the

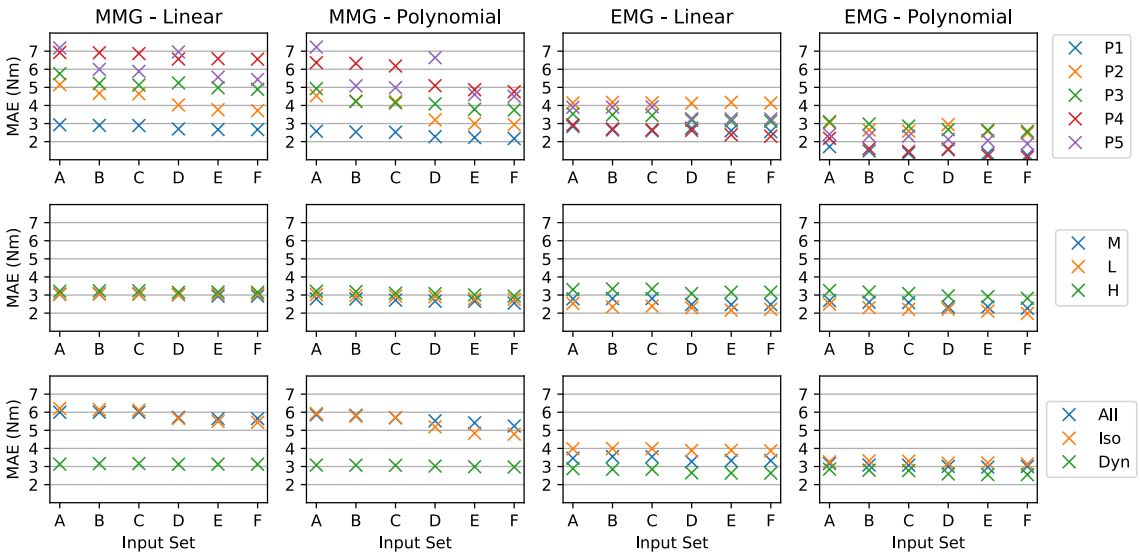


Figure 6.18: Mean Absolute Error Comparison of MMG and EMG. MAE for human joint torque estimation from MMG and EMG muscle activity signals using the linear and polynomial regression models across input sets A to F and various isometric and dynamic exercise sets. The top row shows individual isometric contraction positions, P1 to P5; The middle row shows individual dynamic extension and flexion trajectories, M, L, and H frequency; and the bottom row shows values averaged across all positions and trajectories (All), all isometric positions (Iso), and across all dynamic trajectories (Dyn).

extended human shank position. For dynamic extension and flexion, the MMG human joint torque estimation performance is similar to that of EMG. For the linear regression model, the average MAE in human joint torque estimation is slightly greater for MMG (3.13 Nm) than EMG (2.64 Nm) averaged across dynamic trajectories for the best-performing input set – F. The results show that EMG is much more effective than MMG for predicting human joint torque across isometric and dynamic (isotonic) exercise conditions.

EMG performs better than MMG for the linear regression, polynomial regression, and neural network models, as shown in Fig. 6.19. The MMG dataset predicts human joint torque less effectively during isometric exercise than during dynamic exercise. The linear model prediction error using MMG signals from isometric exercise was around 5.4 Nm, compared to just over 3 Nm using MMG signals from dynamic exercise. This trend is present across all of the models. The results show that EMG predicts human joint torque more effectively than MMG during human-exoskeleton interaction, and that MMG performance is closer to that of EMG during dynamic exercises.

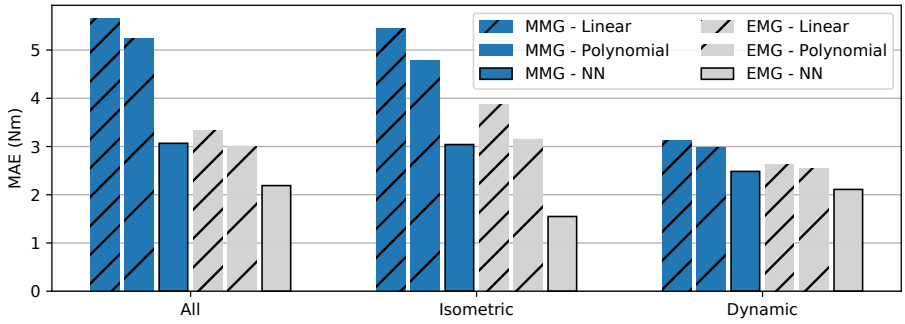


Figure 6.19: Mean Absolute Error Comparison across all Muscle Activities and Models. Mean absolute error comparison for human joint torque prediction using MMG and EMG muscle activity signals with linear regression, polynomial regression, and neural network models trained using the full dataset (All), isometric exercise dataset, and dynamic exercise dataset.

6.3.2 Model Comparison

The polynomial model achieves a better fit than the linear model for regression between input sets A to F and the time-shifted reference human joint torque, as shown in Figure 6.18. Reduced modelling error comes at the expense of more model parameters. For input set F, the polynomial model has 28 parameters and the linear model has 7. The polynomial regression models (6.12) consistently achieve a better fit than the linear regression models (6.6), as shown in Fig. 6.19. Reduced prediction error comes at the expense of more model parameters.

More parameters in the polynomial models allow a better fit to the larger dataset with both exercises (All). The linear models are less effective for the larger dataset, but the difference reduces when the models are trained on the individual types of exercise.

The neural network models performed well when trained with the input dataset including both exercises (All). Prediction error was overall much lower than the linear and polynomial regression models, as shown in Fig. 6.19 and Table 6.5.

The polynomial model is a much better fit when trained on larger datasets, such as for all data (All), all

isometric data (Iso) from exercise 1, and for all dynamic data (Dyn) from exercise 2. The linear model is much less effective for these cases, however difference in estimation error is lower for individual exercise positions and trajectories.

The neural network model was trained for the most general case across all positions and trajectories for both exercises (All) for the full input set F. It performed significantly better than the linear and polynomial regression models, as shown in Table 6.5. It estimated the time-shifted human joint torque with lower MAE and greater Pearson's correlation coefficient. The increased performance comes at the expense of training time and computational power. The trained model also has a much higher number of parameters (137) than the linear (7) and polynomial (28) models. This neural network model is relatively simple with a low number of parameters compared to other models, for example, those used in deep learning. However, it is more complex than the linear and polynomial regression models and so less convenient to deploy in a high frequency real-time control framework.

Fig. 6.20 illustrates the absolute error quartiles for human joint torque prediction using the MMG and EMG signals from both exercises (All). For the regression models, the absolute error values are much lower more closely packed for EMG than for MMG. The median values for the linear and polynomial models are similar for MMG and EMG respectively. The median absolute error values for the neural network models are much lower in both cases. The interquartile range, which provides a measure of variability of the absolute estimation errors, is greatest for the linear models, lower for the polynomial models, and lowest by far for

Table 6.5: Model Summary Statistics. Human joint torque prediction model performance summary statistics using datasets with both exercises (nMAE is normalised MAE).

Signals	MMG (All)			EMG (All)		
Model	Linear	Poly	NN	Linear	Poly	NN
Params	7	28	137	7	28	137
MAE	5.65	5.24	2.77	3.33	3.00	2.19
r	0.61	0.70	0.67	0.78	0.83	0.90
nMAE	0.109	0.101	0.063	0.074	0.066	0.048

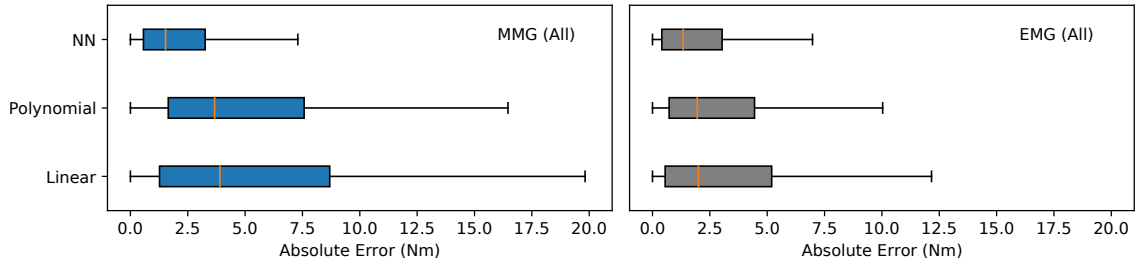


Figure 6.20: Model Error Distributions for MMG and EMG. Distribution of human joint torque prediction absolute errors for the linear, polynomial, and neural network models, estimated using MMG and EMG datasets across both exercises (All). The box plots shows the distribution of quartiles and outlier ranges.

the neural network models.

Figure 6.21 shows the correlation of the estimated $\hat{\tau_h}$ against the time-shifted reference τ_h, trained using input set F with EMG datasets for both exercises. The neural network model has the greatest correlation ($r = 0.90$) with closest grouping around the line of $x = y$. The distributions for the polynomial and linear models appear to have quite similar distributions, however statistical analysis shows that the correlation of the polynomial model ($r = 0.83$) is greater than that of the linear model ($r = 0.73$). Histogram plots of the error distributions show observed errors divided into 50 bins of equal width. The polynomial and linear model errors have similar distributions. The polynomial model errors are slightly more dense around zero (density of around 0.2). The histogram of neural network model error has a much greater density of observations (almost 0.3) around zero, indicating a greater number of highly accurate estimates. It has a narrow and symmetrical distribution of errors, suggesting fewer cases of high error and more consistent performance with respect to over- and under-estimation.

A similar trend is seen in Figure 6.22, which illustrates the absolute error quartiles for EMG. The median values for the linear and polynomial models are similar, 2.00 Nm and 1.95 Nm respectively. The median absolute error for the neural network model is much lower at 1.34 Nm. The Interquartile Range (IQR), which provides a measure of variability of the absolute estimation errors, is greatest for the linear model (4.64 Nm), lower for the polynomial model (3.72 Nm), and lowest for the neural network model (2.63 Nm). The

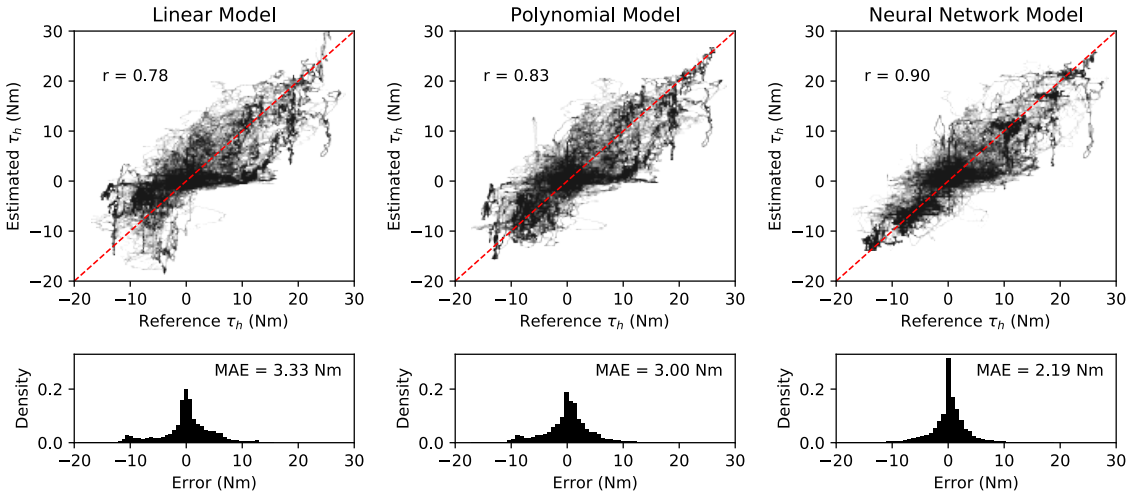


Figure 6.21: HTE Model Regression Results. Plots showing the correlation (top row) and error distribution (bottom row) of estimated human joint torque, $\hat{\tau}_h$, against the time-shifted reference human joint torque, τ_h, for the linear, polynomial and neural network models, estimated using input set F with EMG datasets for both exercises.

histogram of absolute error values, divided onto eight bins in Figure 6.22, shows the neural network model to have the greatest density of highly accurate observations, followed by the polynomial model, and then the linear model.

Human joint torque prediction by the linear, polynomial, and neural network models using MMG and EMG from both exercises is illustrated in Fig. 6.23. The sample is taken from dynamic exercise with torque in the direction of extension. Because the MMG and EMG tests were conducted independently, the reference

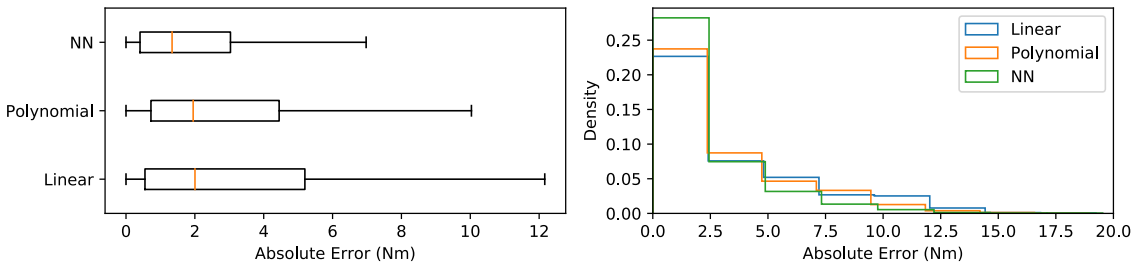


Figure 6.22: HTE Model Error Distributions for EMG. Plots showing the distribution of human joint torque estimation absolute errors for the linear, polynomial, and neural network models, estimated using EMG datasets across both exercises for input set F. The box plot (left) shows the distribution of quartiles and outlier ranges, and the histogram (right) shows the density distribution of absolute errors.

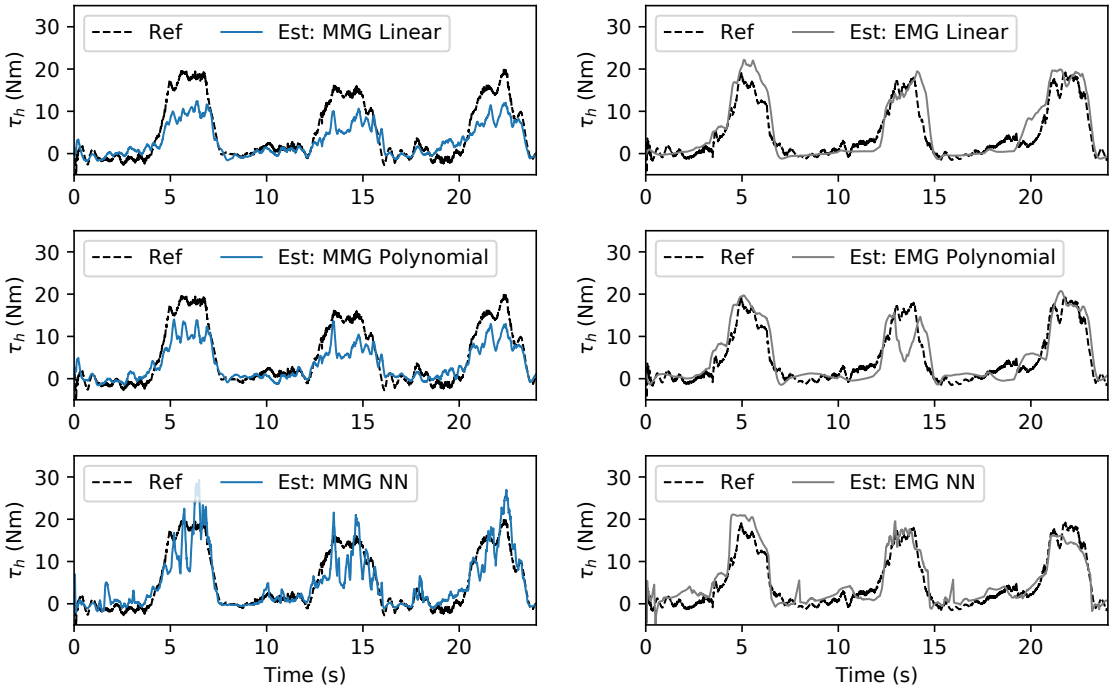


Figure 6.23: Human Joint Torque Prediction. Demonstration of human joint torque prediction with MMG and EMG using the training dataset with both exercises for linear (top row), polynomial (middle row), and neural network (bottom row) models.

torque profiles vary slightly. The estimation profile for EMG is closer to the reference torque and less noisy. All models demonstrate prediction from the onset of human joint torque. At the onset of contraction the EMG signals overshoot force prediction, in agreement with the literature [204]. Satisfactory agreement between reference torque and estimated torque is seen for the linear, polynomial, and neural network models. The neural network models have the least smooth torque estimation profiles, with sudden spikes and deviations. This is because they are nonlinear and the outputs can change more suddenly. This feature is more prominent in the MMG predictions. Filtering of the output signals may counteract this, however introduction of significant phase delay should be avoided.

6.3.3 Discussion

The linear and polynomial modelling approaches assume linear and quadratic relationships, respectively, between human joint torque and the input variables. This relationship contains nonlinearities which are not captured in these models. The neural network models capture some nonlinear features in the data. With neural networks it is possible to build complex models that outperform linear or quadratic approaches. However, it is important to avoid overfitting and candidate models should not be unnecessarily complicated.

Performance of neural networks models is dependent on having a large enough and reliable dataset for training. The recorded human joint torque estimation datasets meet this requirement so the models were trained effectively. Training neural networks can be computationally expensive and trained models can be difficult to implement in real-time compared to linear and polynomial models. Model complexity for real-time prediction at high sampling rates is limited and this was considered in the model design.

Comparison of MMG and EMG signals during muscle monitoring for isometric contraction has shown that MMG signal onset precedes EMG with a difference of 41 ± 181 ms [201]. MMG offset occurred after EMG at 202 ± 223 ms, meaning the signal has a significantly longer duration for the same muscle activity. MMG and EMG signals are produced prior to muscle contraction so both signals can provide advance prediction of muscle force [189].

For ANN rehabilitation, it is of great importance to accurately predict the human joint torque so that an accurately needed assistance can be provided by the exoskeleton. A recent work [182] has shown that EMG signals can be used to support an ANN control of a rehabilitation robot by using a linear regression method for human joint torque estimation. In this thesis, the experimental results show that using MMG with a neural network model achieved higher accuracy than using EMG with a linear regression method

for human joint torque estimation. Although the MMG is less mature than EMG systems, the advantages of the MMG for practical use (i.e., immunity to changes in skin impedance and elimination of the need of shaving and conductive gel) can largely benefit the development of lower limb exoskeletons, in particular for regular use. While further studies are necessary to parameterise the precise amount of support in ANN paradigms, formulation of assist based on robust muscle feedback offers a basis for such studies. In addition, MMG can be an effective means to estimate human joint torque for emerging hybrid electrical stimulation and exoskeleton systems. Note that EMG cannot be used during muscle stimulation. Moreover, the experimental results also show that more accurate and robust estimation of human joint torque can be expected by combining the advantages of EMG and MMG.

Chapter Summary

This chapter presented the method for estimating human joint torque from muscle activity signals and system states in advance of torque being produced at the knee joint. Background information was provided about previous usage of muscle activity signals for estimating muscle force. In the methods and materials section, the MMG and EMG muscle activity signal recording and processing procedures were described. The experimental procedure including isometric and dynamic exercises used for recording human joint torque estimation data were provided. The linear, polynomial, and neural network modelling approaches were presented. In the results and discussion section, the performance of MMG and EMG signals for estimating human joint torque were compared. As a more mature means of sensing muscle activity, EMG estimated human knee joint torque more accurately compared to the emerging microphone-based MMG. MMG performance was comparable when estimating human joint torques for dynamic exercises. The modelling approaches were evaluated using data for MMG and EMG signals from isometric and dynamic

exercises. The models and input signals were compared based on human joint torque prediction absolute error and correlation. The neural network models demonstrated the lowest estimation error and highest correlation, however these models have the highest number of parameters and require much more computational power for training and implementation. EMG is used in the control architecture evaluation.

Chapter 7

Controller Evaluation

The previous chapter explored the materials and procedures used for estimating human knee joint torque in advance using muscle activity signals from the thigh and human-exoskeleton system states. Various input datasets and modelling approaches were investigated and human joint torque estimation was evaluated. The human joint torque estimate is a critical part of the proposed control framework, which is evaluated in this chapter.

The proposed AAN-MPC control framework is evaluated with multiple healthy subjects for two dynamic exercises: seated extension and flexion, and walking. The controller is tested assisting each exercise in the human involvement conditions of neutral, assisting and resisting, in which the controller demonstrates the operation modes of passive, active-assist, and safety, respectively.

This chapter begins with a materials and methods section, which provides subject information, the human joint torque estimation method, and the experimental setup for the seated and walking exercises. The following section contains the results and discussion of the controller evaluation for the seated leg extension and flexion exercise and walking demonstration, considering trajectory tracking, human-robot interaction,

and assistance mode detection.

7.1 Materials and Methods

In this section, the test procedures for the controller evaluation are described, including the test subject details, choice of human joint torque estimation model, and experimental setup for the seated extension and flexion exercise and demonstration of the controller during the swing phase of walking.

7.1.1 Test Subjects

The seated tracking experiments were conducted with three healthy subjects (S1-3), able to execute complete flexion and extension movements of the knee joint without spasticity nor contracture. The upright walking experiments were conducted with one healthy subject (S1). Subject information is provided in Table 7.1. All subjects were able to actively follow or obstruct the knee joint trajectories while observing the progress on a monitor in real-time using the testing GUI shown in Figure 6.15. The subjects were given a patient information sheet regarding the assessment and gave written informed consent to all experimental procedures. The study received ethical approval from the Research Ethics Committee of Imperial College London. Constraints on the knee joint states and control input were imposed within the controller, and, for safety, device range of motion was securely limited between full extension (90 °) and full flexion (-30 °) mechanically.

7.1.2 Human Joint Torque Estimation

For the controller evaluation tests, the linear human joint torque estimation model, described in Section 6.2.3.1, was used with two inputs (input set A from Section 6.2.2.4). Using a minimal input set and low complexity model demonstrates the ability for the control approach to perform effectively without highly accurate prediction of human joint torque. A strength of the FLA developed in Chapter 5 is its ability to identify the most suitable mode of assistance using only the correct polarity of the estimated human joint torque. Additionally, for the standing exoskeleton configuration, shown in Figure 7.1a, EMG placement is more convenient with two sensors, as the electrodes must fit around the segment braces, as shown for the VM electrode pair and ground electrode in Figure 7.1b. As discussed in Section 6.3.2 and shown in Figure 6.18, the simple linear model has similar MAE to the more sophisticated polynomial model, which has a higher number of parameters, when trained using only the dynamic exercise datasets for a specific trajectory frequency.

7.1.2.1 Data Collection

The linear human torque estimation model was trained for each subject individually using input set A (see Section 6.2.2.6) with EMG muscle activity signals for data recorded during the dynamic extension and flexion exercise at medium frequency (see Section 6.2.2.1). This includes five human involvement conditions: Neutral (N), Extension Assist (EA), Extension Resist (ER), Flexion Assist (FA), Flexion Resist (FR). For three subjects, data was recorded over six repetitions of 4 s trials. The same exercise pattern

Table 7.1: Subject Information. Details about the three subjects included in the controller evaluation tests.

Subject	Sex	Age	Height (cm)	Weight (kg)
S_1	Male	27	188	85
S_2	Female	22	155	53
S_3	Male	27	179	65

(a) Full length profile view (without EMG electrodes). **(b)** Close-up front view (with EMG electrodes).

Figure 7.1: Portable Standing Exoskeleton Configuration. Knee joint exoskeleton with i) backpack containing rechargeable battery pack, portable EMG amplifier, and wireless router, ii) thigh segment braces, iii) shank segment braces, iv) EMG electrode pair for the VM muscle of the thigh, v) ground EMG electrode at the knee, and vi) actuator module connected to power (battery) and Ethernet (router). A short video demonstration of the standing exoskeleton configuration can be found here.

and repetition frequency is used for the seated controller evaluation tests.

7.1.2.2 Torque Estimation Performance

The linear human joint torque model parameters and model error for each subject are shown in Table 7.2.

The RMS torque estimation error can be written as

$$\text{RMSE}_{\hat{\tau}_h} = \sqrt{\frac{1}{n}\sum(\hat{\tau}_h - \tau_{hr})^2} \tag{7.1}$$

where $n = 60000$ is the total number of data samples across six repetitions of 4 s trials sampling at 500 Hz in the five human involvement conditions. The reference human torque τ_{hr} and estimated human torque $\hat{\tau}_h$ for all human involvement conditions are shown in Figure 7.2. Good agreement can be seen from the linear agonist-antagonist model. The estimation accuracy (mean $\text{RMSE}_{\hat{\tau}_h}$ = 2.82 Nm across all

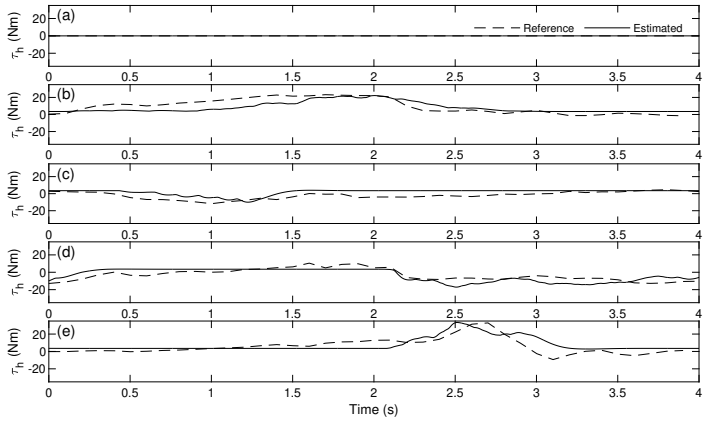


Figure 7.2: Human Joint Torque Estimation. Reference human torque, τ_{hr}, and estimated human torque, $\hat{\tau}_h$, from 4 s of linear human torque model parameter identification trials for subject 1 in the following involvement conditions: a) neutral (N), b) extension assist (EA), c) extension resist (ER), d) flexion assist (FA), and e) flexion resist (FR).

three subjects), ensures correct assistance mode detection using the FLA. The normalised RMS torque estimation error, $\text{NRMSE}_{\hat{\tau}_h}$ can be calculated by scaling $\text{RMSE}_{\hat{\tau}_h}$ by the range of reference human torque, $(\tau_{hr,\max} - \tau_{hr,\min}) = 40.6$ Nm, which gives $\text{NRMSE}_{\hat{\tau}_h}$ of 0.0695. This is comparable to previous knee joint torque estimation studies [173].

7.1.3 Experimental Setup

Performance of the AAN-MPC approach with HTE and FLA for assistance mode detection was evaluated for tracking during knee joint extension/flexion movements in a seated position and demonstrated for the swing phase of upright walking. Knee joint movements of the swinging leg during a gait cycle have a pendulum oscillatory extension/flexion behaviour that is close to the natural walking frequency [176]. This

Table 7.2: HTE Model Information. Linear human joint torque estimation parameters and RMS torque estimation error for each subject.

Subject	b_0	a_1	a_2	$\text{RMSE}_{\hat{\tau}_h}$ (Nm)
S_1	0.181	206.2	-90.5	2.96
S_2	0.127	163.8	-110.1	2.38
S_3	0.204	181.7	-132.8	3.11

makes seated leg extension and flexion exercise an effective proxy for robotic rehabilitation and mobility assistance.

The aim of the controller evaluation was to demonstrate tracking performance of the proposed controller and the adaptation capability for different human involvement conditions, i.e., adjusting the nature of human-robot interaction. The controller was tested operating in the following modes:

Passive Mode: The subjects are neutral, relaxing and contributing negligible torque, relying on the exoskeleton to generate torque to successfully complete the tracking task with minimal position error. This is a robot-dominant condition.

Active-Assist Mode: The subjects voluntarily generate joint torque to track the target extension/flexion movements. The exoskeleton should show compliance and only provide the deficient torque required to successfully complete the exercise task. This mode enables human-robot cooperation and is neither human nor robot-dominant. The extent to which human/robot is responsible for generating torque is influenced by the active-assist penalty parameter, $p_A = 0.5$, configured within the FLA, as described in Section 5.2.2.

Safety Mode: The subjects generate joint torque which obstructs the target extension/flexion movement. In practice this could be due to neurological patient pathology, limited range of motion, or subject discomfort, for example. This is a potentially dangerous and human-dominant condition in which the subject is not cooperating. For in-clinic and in-home assistance, assessment and rehabilitation, the safety of human–robot interactions, specifically patient–robot interaction can be a major concern [225]. It is beneficial for the controller to identify this and adapt appropriately. The exoskeleton should show high levels of compliance and not prioritise generating torque for trajectory tracking.

7.1.3.1 Seated Tracking

The first controller evaluation exercise, tested with all three subjects (S1-3), was seated leg extension and flexion. The same exercise was used for developing the HTE approach in Chapter 6 and described in detail in Section 6.2.2.1. For the reference movement profiles, a sinusoidal knee joint angle trajectory with amplitude of 1 rad can be written as

$$\theta_r(t) = 0.5\cos(2\pi f t - \pi) + 0.7 \tag{7.2}$$

where θ_r is the reference joint angle in radians, $f = 0.25$ is repetition frequency in Hz, and t is time in seconds. The frequency is chosen to approximate that of knee joint movement in normal walking gait [226]. The seated tracking controller evaluation trials were conducted over 24 s periods for each of the five human involvement conditions described in Section 6.2.2.1. The trials were kept relatively short to prevent subject fatigue and discomfort.

7.1.3.2 Walking Demonstration

The second controller evaluation test was a short walking demonstration tested with one subject (S1). This involved the standing exoskeleton configuration assisting knee extension and flexion during the swing phase of upright walking. Trials lasted 30 s for a number of self-selected speed gait cycles, with the subject instructed to behave in passive, assistive, and resistive human-robot interaction conditions.

For the walking reference movement profiles, the knee joint angle trajectory is recorded offline from either leg of the subject during self-selected speed walking. During assisted exercise, the standing exoskeleton configuration, presented in Section 4.2.3, provides assistance to the knee joint of the left leg of the subject. Therefore, the reference joint angle trajectory is recorded on the (opposite) right leg of the subject using

Figure 7.3: x-io Technologies x-IMU. Three x-IMU sensors (placed on the subject's foot, shank, and thigh segments) are used for recording walking reference trajectories, and a single x-IMU (placed on the subject's foot) is used during the controller evaluation walking tests to detect the swing phase of gait.

three x-IMU sensor boards (shown in Figure 7.3). The sensors are mounted on the subject using disposable electrical tape at the foot, shank, and thigh segments. Each IMU consists of a 3-D accelerometer, 3-D gyroscope, and 3-D magnetometer. The rotation quaternion of each IMU with respect to the global coordinate system G can be evaluated based on the measurements of these sensors. A built in algorithm of high accuracy with no drift is used to compute 3-D orientation for both static and dynamic movements. The quaternions provided by each IMU are used directly.

For accurate relative kinematics of the human body wearing the robot, a body segment coordinate system B is built based on the standards of the International Society of Biomechanics [227] as follows: origin, centre of rotation; X, forward in sagittal plane; Y, up; and Z, right in frontal plane. The global frame G is defined as: X, local magnetic north direction; Y, local magnetic west direction; and Z, vertical direction. The X-IMU sensors are all calibrated to provide an estimated transformation matrix from (shared) sensor frame S to body frame B denoted by

$$ {}^{B}\boldsymbol{R}_{S,j} = {}^{S}\boldsymbol{R}_{B,j}^{-1} = [{}^{S}\boldsymbol{X}_{B,j}\, {}^{S}\boldsymbol{Y}_{B,j}\, {}^{S}\boldsymbol{Z}_{B,j}]^{-1} \in \mathbb{R}^{3\times 3} \tag{7.3} $$

where ${}^{S}\boldsymbol{X}_{B,j}$, ${}^{S}\boldsymbol{Y}_{B,j}$, ${}^{S}\boldsymbol{Z}_{B,j}$ denote the segment frame's unit x-axis, y-axis, and z-axis, respectively, measured by the IMUs. $j \in \{1, 2, 3\}$ denotes the IMU number (index) for the foot, shank, and thigh segments, respectively [228]. Appendix A.3 gives the calibration procedure for estimating the transformation matrix

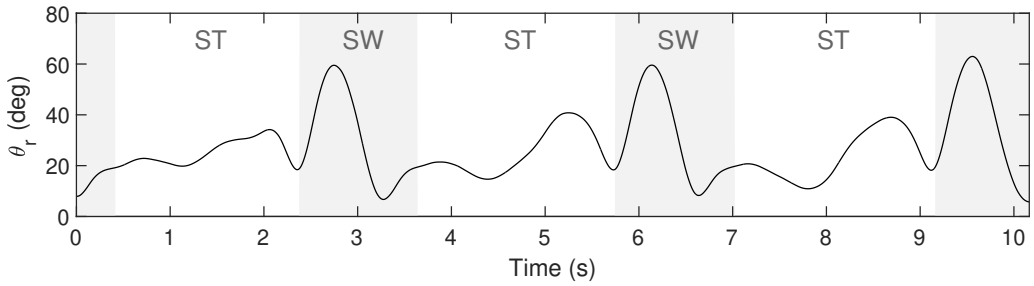


Figure 7.4: Walking Reference Trajectory. Reference knee joint angle trajectory for stance and swing phases of gait recorded from opposite leg of subject 1 during self-selected speed walking.

${}^{S}\boldsymbol{R}_{B,j}$ in detail.

For the walking reference, the subject was instructed to walk at a self-selected speed for five full gait cycles, staying still and standing upright for 10 s before and after walking to provide a stationary position reference. The knee joint reference trajectory for three gait cycles are shown in Figure 7.4. The stance (ST, white) and swing (SW, grey) phases of gait were identified using a gait phase detection algorithm from Huo et al. [176]. The algorithm uses an IMU sensor attached to the subject's foot to detect whether the leg is in swing or stance phase during walking.

For the walking demonstration, the AAN-MPC framework only provides assistance during the detected swing phase of walking. During swing, significant knee joint flexion and extension is observed. There is more potential deviation from normal walking gait, and various modes of human-robot interaction are more likely to occur. The magnitude of torque required for knee joint movement during the swing phase is typically within the ± 20 Nm actuator operating range (see Appendix A.1).

Due to observed high velocity extension and flexion movements during the swing (shown in the SW regions of Figure 7.4), the maximum and minimum velocity constraints of the MPC (see (4.11c) in Section 4.3.6) were increased from ± 2 rad/s to ± 8 rad/s. During the stance phases of the walking demonstration, the exoskeleton operates in transparent mode, applying zero torque at the knee joint. The subject is free to stand without impedance from the device.

7.2 Results and Discussion

In this section, the results from the controller evaluation tests are presented and discussed. Firstly, the seated tracking exercise results are explored. The performance metric of human torque ratio is defined for evaluating this exercise. The analysis looks at controller performance with three subjects in terms of trajectory tracking, human robot interaction, and assistance mode detection. Finally, the results of the walking demonstration with one subject are presented.

7.2.1 Seated Tracking

The AAN-MPC controller evaluation results are presented for three subjects performing a knee joint tracking task in five human involvement conditions: neutral (N), extension assist (EA), extension resist (ER), flexion assist (FA), and flexion resist (FR). N should invoke passive mode, EA and FA should invoke active-assist mode, and ER and FR should invoke safety mode.

Performance is evaluated during extension (E) and flexion (F) movement phases of the exercise in terms of suitable robot compliance during tracking (reflected in tracking error), human-robot cooperation/interaction (reflected in ratio of human torque to robot torque), and suitable assistance mode detection. The metric of human torque ratio, R_h, is defined as

$$R_h = \frac{\bar{\hat{\tau}}_h}{\bar{\hat{\tau}}_h + \bar{\tau}_e} \tag{7.4}$$

where $\bar{\hat{\tau}}_h$ and $\bar{\tau}_e$ denote the root-mean-square (RMS) of the estimated human torque and the exoskeleton torque, respectively.

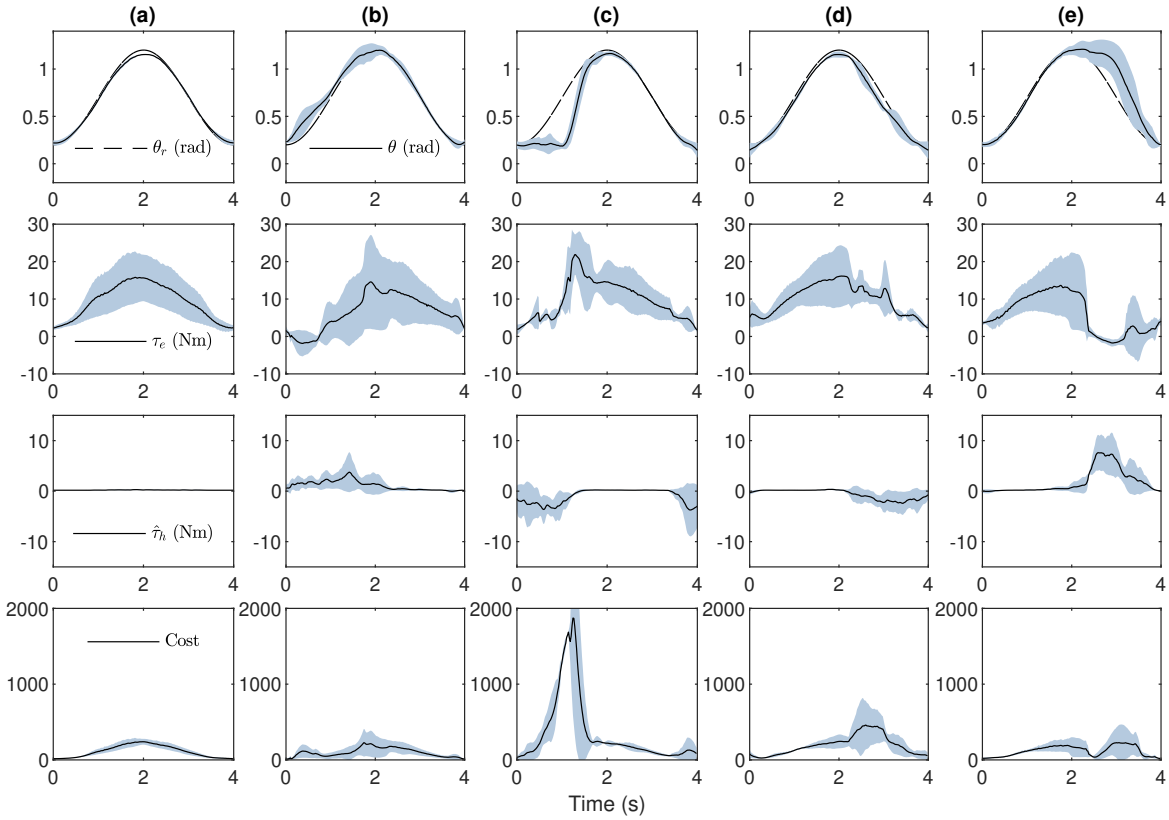


Figure 7.5: Seated Exercise Controller Evaluation Results. Tracking results from three subjects for the MPC-AAN controller with FLA assistance mode detection in five human involvement conditions. The plots show the average values and shaded Standard Deviation (SD). From left to right the columns show: a) neutral (N), b) extension assist (EA), c) extension resist (ER), d) flexion assist (FA), and e) flexion resist (FR). From top to bottom, rows display reference and actual trajectory, θ_r and θ, robot torque, τ_e, estimated human join torque, $\hat{\tau}_h$, and objective function cost.

7.2.1.1 Trajectory Tracking

Tracking results averaged across all three subjects can be seen for the five human involvement conditions in Figure 7.5, with Standard Deviation (SD) shown as a shaded region. The significant SD seen in exoskeleton torque across all involvement conditions is a result of varying subject properties, with smaller and lighter subjects requiring lower exoskeleton torque to complete the exercise. SD is also observed in the human joint torque estimation, $\hat{\tau}_h$, due to subjects applying different levels of effort for varying time during the trials.

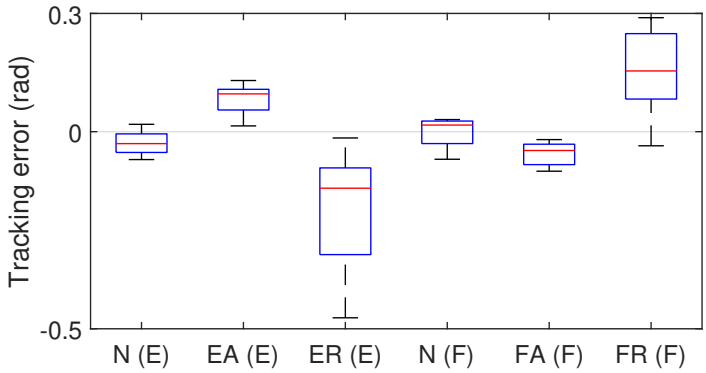


Figure 7.6: Tracking Error Box Plots. Box plots comparing percentiles of tracking error (rad) for extension (E), from 0 to 2 s, and flexion (F), from 2 to 4 s, in the five human involvement conditions (N, EA, ER, FA, and FR) averaged across all subjects. Large error values reflect robot compliance.

In the robot-dominant N condition the controller operates in passive mode. It closely follows the target trajectory with very low tracking error, as shown in Figure 7.6 for both extension and flexion (medians of -0.03 rad and 0.02 rad). There is minimal tracking SD across the relaxed trials. Low human torque is correctly estimated in this condition with minimal SD. The total objective function cost remains low and very little compliance is shown by the robot. This is reflected in very close trajectory tracking, despite a lack of human cooperation, with torque contributions very low for extension ($R_h = 0.036$) and flexion ($R_h = 0.042$).

During EA and FA involvement conditions the FLA recognises assistive human involvement and sets μ_A high, which shifts the controller to active-assist mode. In this mode it correctly permits greater tracking error. Figure 7.5b and Figure 7.5d show the controller behaving with greater compliance as the subjects begin to apply voluntary torque. This is shown in Figure 7.6 as significantly increased tracking error, with median errors of 0.10 rad and -0.05 rad during the extension and flexion phases respectively. These are a significant increase from the corresponding phases for N condition.

During ER and FR conditions, the controller shows even greater compliance, as instructed by setting the safety penalty $p_S = 1.0 > p_A = 0.5$. This can be seen in Figure 7.5c and Figure 7.5e where the joint angle significantly deviates from the reference trajectory. This leads to increased overall cost, caused partly by

significant tracking error (medians of -0.14 rad and 0.15 rad for extension and flexion phases respectively). For the ER condition, once the FLA no longer outputs high μ_S and detects human relaxation, the tracking error is minimised with high corrective around 1.25 s, reducing observed cost. The augmented Lagrangian optimisation method adjoins constraint equations to the overall MPC objective function. Cost levels exceeding 2000 are shown for ER in Figure 7.5c caused by activation of a minimum position constraint during relaxation and a maximum velocity constraint during trajectory recovery.

7.2.1.2 Human-Robot Interaction

The average human torque ratio, R_h, throughout the exercise across all subjects for each condition is shown is Figure 7.7. In N involvement condition, very little human cooperation is shown as the human torque ratio is very low ($R_h = 0.036$ for extension and $R_h = 0.042$ for flexion).

For EA and FA, increased human-robot cooperation is expected, and corresponding human torque ratio is observed, with $R_h = 0.46$ for the extension phase of EA and $R_h = 0.18$ for the flexion phase of FA. The human effort is shown in Figure 7.5b during EA, between 0.5 s and 2 s, as exoskeleton torque, τ_e, is much lower compared to N condition, despite noticeable tracking error.

For ER and FR, the calculated human torque ratio values are significantly greater ($R_h = 0.23$ and $R_h = 0.59$ for extension and flexion phases respectively) compared to N. However, the human effort observed during these phases obstructs the exercise, so does not signify human-robot cooperation. It is impeding the robot and is not helpful for the tracking task.

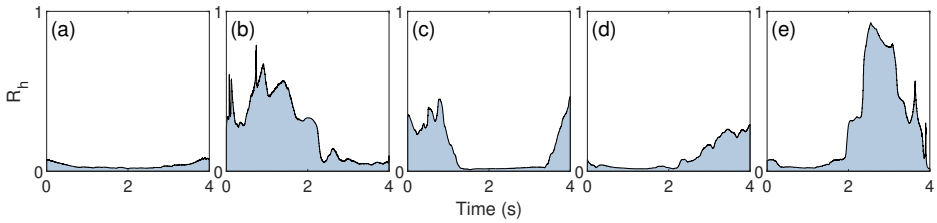


Figure 7.7: Human Torque Ratios. Human torque ratio, R_h, throughout the exercise tasks, averaged across all three subjects for the five human involvement conditions: a) neutral (N), b) extension assist (EA), c) extension resist (ER), d) flexion assist (FA), and e) flexion resist (FR).

7.2.1.3 Assistance Mode Detection

The RMS active-assist and safety assistance mode likelihood values, μ_A and μ_S, averaged across all three subjects are shown for the five human involvement conditions in Figure 7.8. The values show that the FLA successfully identifies the correct mode of operation for relaxed, human-assistive and human-resistive involvement conditions. The greatest RMS active-assist mode likelihood values, μ_A, occur during the extension and flexion phases of EA and FA respectively (0.50 and 0.45) and the greatest RMS safety mode likelihood values, μ_S, occur during the extension and flexion phases of ER and FR respectively (0.45 and 0.60).

The FLA input signals, $\dot{\theta}_r$ and $\hat{\tau}_h$, active-assist and safety mode likelihood values, μ_A and μ_S, and the assistance mode variable, m, for each human involvement condition are shown for subject 1 in Figure 5.5. The FLA performs well, consistently identifying the appropriate assistance mode throughout the task. For N condition, both μ_A and μ_S remain low, as the FLA recognises human relaxation and instructs the controller to operate in passive mode. The assistance mode function, m, behaves as expected, falling closer to zero for safety mode than for active-assist mode ($p_S = 1.0 > p_A = 0.5$). The assistance mode likelihoods μ_A and μ_S are activated on a mutually exclusive basis.

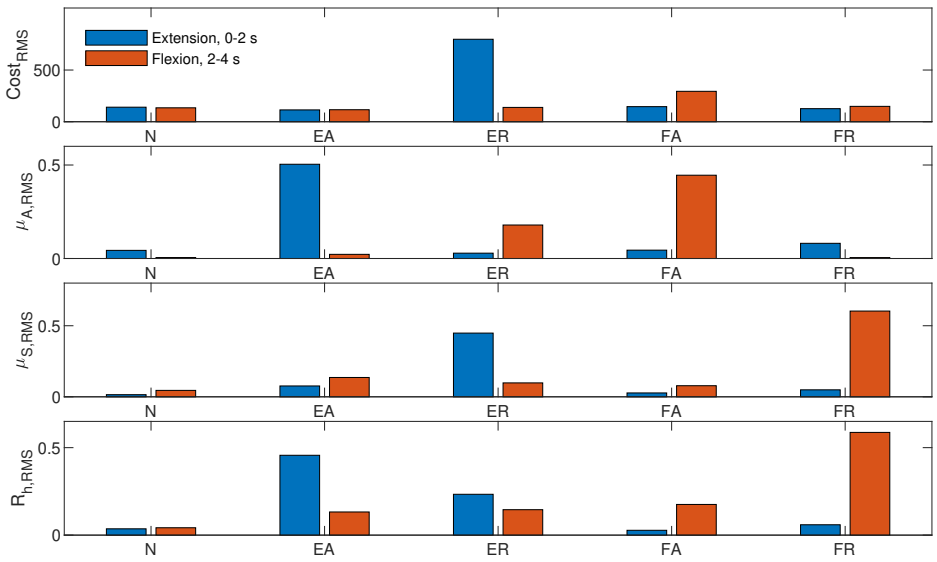


Figure 7.8: Controller Evaluation Metrics. Bar charts showing, from top to bottom, RMS values of cost, active-assist mode likelihood, μ_A, safety mode likelihood, μ_S, and human torque ratio, R_h, for all subjects. Results are divided into extension phase, from 0 to 2 s in the trials, and flexion phase, from 2 to 4 s, for neutral (N), extension assist (EA), extension resist (ER), flexion assist (FA), and flexion resist (FR) human involvement conditions.

7.2.2 Walking Demonstration

In this section of the controller evaluation, results of the walking demonstration are presented. This was an applied demonstration of the AAN-MPC control framework for a dynamic and practical exercise, which is used for many applications within mobility assistance and rehabilitation robotics. The aim here is to show that the same controller behaviour and functionality observed in the seated tracking experiments is present for a more extensive exercise implementation.

The controller provided assistance to one subject during upright walking in three human involvement conditions: neutral (N), (extension or flexion) assist (A), and (extension or flexion) resist (R). The N condition should invoke passive mode, A should invoke active-assist mode, and R should invoke safety mode.

The results presented here are a subset of the trials, chosen to represent the controller operating in each of the three operating modes. As described in Section 7.1.3.2, the exoskeleton operated in transparent mode

($\tau_e = 0$) during the stance phase of walking for exoskeleton side, and was intelligently controlled using during the swing phase. Therefore, the demonstration results show the system and controller behaviour for the swing phase only. Figure 7.9 shows a sample of walking demonstration results for the AAN-MPC control framework assisting knee extension and flexion in the three assistance modes of a) passive, b) active-assist, and c) safety.

In the example of passive mode (left column), the estimated human joint torque, $\hat{\tau}_h$, is low, so the FLA does not identify either active-assist, μ_A, or safety, μ_S, likelihood values. The FLA output value, m, is high, so the MPC cost function prioritises minimising tracking error over robot torque. Therefore, the controller assists the subject with low tracking error, showing little compliance.

In the example of active-assist mode, the estimated human joint torque, $\hat{\tau}_h$, during the flexion region of the swing phase, between 0.4 s and 0.6 s, is negative, assistive, and large enough in magnitude for the FLA to identify significant active-assist likelihood, μ_A. With $p_A = 0.5$, the FLA output, m, reduces significantly below 1 to around 0.6. This change in m causes the MPC cost function to reduce the weighting of tracking error so the controller displays compliance and the tracking error increases over this period.

In the example of safety mode, the estimated human torque during the extension region of the swing phase, between 0.1 s and 0.3 s, is negative, resistive, and very large in magnitude, at almost -3 Nm. This resistive human involvement causes the FLA to identify very high safety likelihood, μ_S, tending towards 1.0. With $p_S = 1.0$, the FLA output, m, falls close to zero. This changes the MPC cost function, causing the controller to neglect the weight of tracking error and instead reduce robot torque. This is seen in, τ_e, between 0.1 s and 0.4 s. Reduced robot torque causes the robot to display compliance and results in increased tracking error.

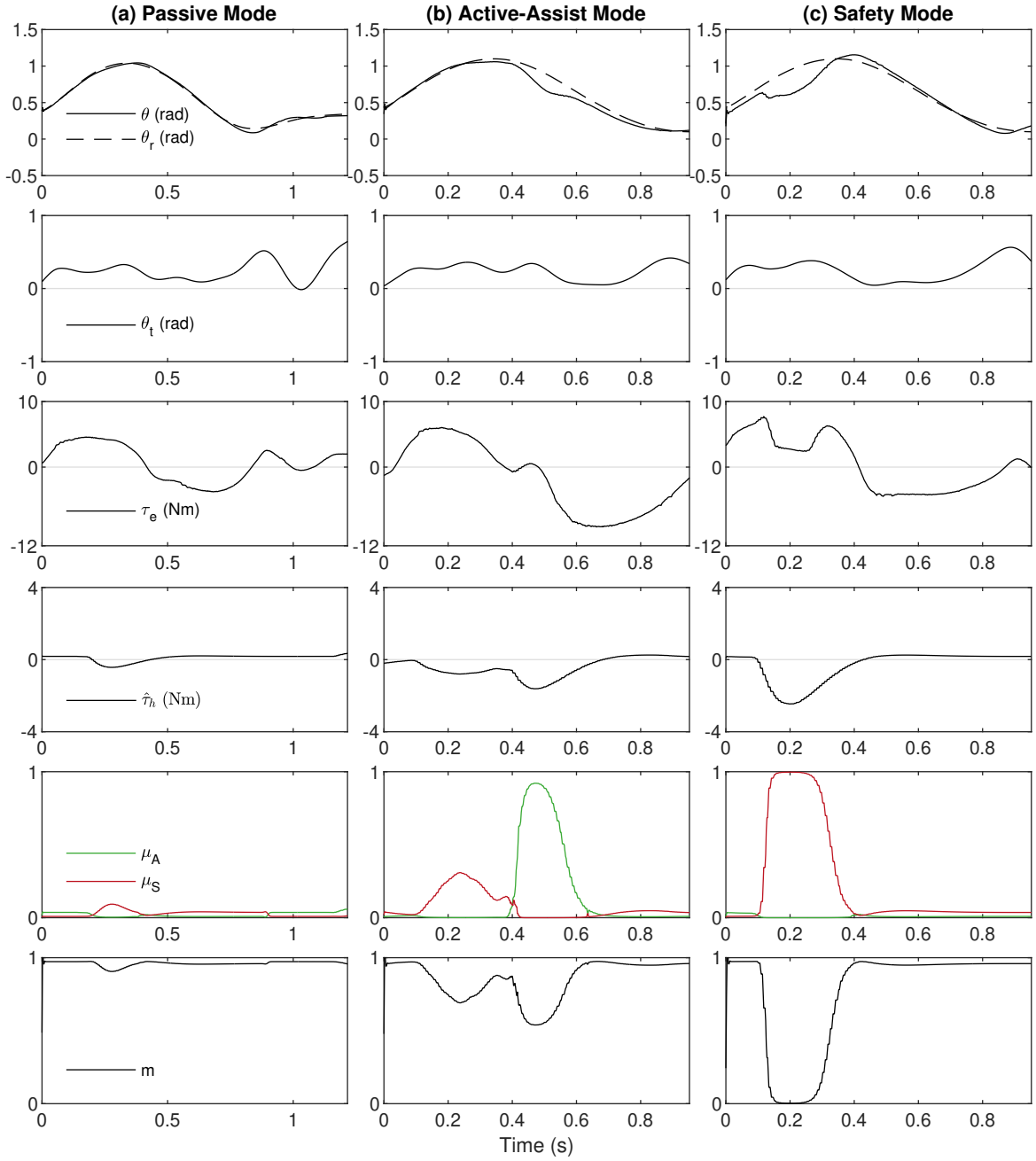


Figure 7.9: Walking Demonstration Results. Controller evaluation walking demonstration results, showing the AAN-MPC controller operating in a) passive, b) active-assist, and c) safety modes. From top to bottom, the rows show actual and reference knee joint angle, θ and θ_r, thigh inclination angle, θ_t, robot torque, τ_e, estimated human torque, $\hat{\tau}_h$, active-assist mode likelihood, μ_A, safety mode likelihood, μ_S, and assistance mode function output, m.

Chapter Summary

In this chapter, the proposed AAN-MPC control framework was evaluated with multiple healthy subjects for two dynamic exercises: seated leg extension and flexion, and the swing phase of overground walking. The controller was tested assisting tracking for each exercise in the human involvement conditions of neutral, assisting, and resisting, in which the controller demonstrated autonomous operation in the assistance modes of passive, active-assist, and safety. The chapter began with the materials and methods section, which described the test subjects, the implemented human joint torque estimation method, and the experimental setup for seated tracking and the walking demonstration. The latter required offline recording of the knee joint angle reference trajectory for the swing phase and an Inertial Measurement Unit (IMU) sensor orientation calibration procedure. The following results and discussion section presented the controller evaluation results for both exercises. For the seated tracking exercise, this included quantitative analysis summarising the controller performance in terms of trajectory tracking, human-robot interaction, and assistance mode detection. For the walking demonstration, samples of assistance for each operating mode were presented.

Chapter 8

Conclusion

In this chapter, findings from the research presented in this thesis are summarised. Features of the novel AAN-MPC control framework are considered including the MPC, FLA for determining assistance mode, and HTE method. Achievements are discussed in relation to the control framework aims set out in Section 4.1 which seek to address some of the prominent problems with existing AAN control approaches, outlined in Section 3.2.4. There will be a discussion of the applications of the work and finally directions in which this research can be built upon.

8.1 Main Conclusions

In this thesis, an AAN control approach is proposed which combines constrained nonlinear MPC with a fuzzy logic-based algorithm for assistance mode detection. Both use human joint torque estimated from processed EMG signals using a predictive model. It is used in the MPC for deriving the robot torque required to successfully follow a target trajectory and in the FLA for evaluating human involvement condition and selecting the most suitable assistance mode.

Although the EMG-based method depends on muscle activity signal measurement and model training to extract the torque output, the method benefits from advanced prediction of human behaviour. It can also distinguish disturbance caused by human effort from other external factors. This is an advantage over sensorless approaches such as nonlinear disturbance observers.

The FLA enables the controller to change behaviour fluidly, operating in assistance modes that fall on a continuous spectrum. Human-robot interaction changes on a spectrum with human-assisting at one extreme, human-relaxed in the middle, and human-resisting at the other extreme. For convenient analysis, human involvement conditions are discretely defined for knee joint tracking as relaxed (R), extension assist (EA), extension resist (ER), flexion assist (FA), and flexion resist (FR). Without human torque, the controller operates in passive mode, which is a servo-like robot-dominant condition. This is achieved by setting high tracking weight, w_θ, in the MPC cost function. In this mode the exoskeleton generates nearly all of the required torque for the task. In active-assist mode, the controller shows compliance and enables robot-human cooperation, contributing less torque to the exercise task, with the subject applying greater effort. In safety mode, the controller exhibits very high levels of compliance for the safety and comfort of the human, and trajectory tracking is given low priority, with very little robot torque produced for the task.

To further improve subject safety, a reference trajectory adaptation method (see Section 5.2.3) can halt the reference trajectory when the likelihood of the safety mode is higher than the adjustable human resistance threshold, R_{th}. This is a utilisation of high level human involvement information from the FLA. Trajectory adaption can prevent the subject experiencing uncomfortable levels of resistance to the rehabilitation task and prevent the subject falling behind with the task. This enables them to safely pick up at the reference point where they were last comfortable and remain engaged with the exercise.

The controller is demonstrated tracking a 0.25 Hz sinusoidal reference trajectory in a seated leg extension and flexion exercise in the five representative human involvement conditions. The controller is then

demonstrated for the swing phase of a walking exercise. The FLA is shown to identify the appropriate assistance mode during each involvement condition.

For the recovery of motor function, active patient participation in generating a target exercise movement is crucial. The proposed controller can intelligently encourage this by identifying subject engagement on-the-fly without the need for repetitive training trials, which can use up valuable clinic time. Practitioners can easily configure the controller for the subject and exercise by changing the value of the MPC cost function tracking weight w_θ, constraints on joint states (or (non)linear combinations of them), active-assist penalty parameter, p_A, and human resistance threshold, R_{th}.

8.2 Achievements

The achievements of this thesis are summarised in relation to the AAN-MPC control approach aims, provided in Section 3.2.4, which address the significant problems identified with existing AAN control approaches:

A – *Adaptation Latency*

Advance Adaptation: This was addressed by adjusting human-robot interaction based on future subject behaviour. Human joint torque was predicted using muscle activity signals that preceded muscle force output. This is demonstrated in Figure 7.2. Adaptation of the control policy using the FLA was demonstrated for various human interaction conditions in Section 7.2.

B – *Singular Modality*

Assistance Modes: The FLA, presented in Section 5.2, infers the human-robot interaction condition on-the-fly and changes the mode of assistance. The output operates directly on the MPC cost function, as described in Section 4.3.7. The ability of the FLA to detect neutral, assistive, and resistive interaction conditions is shown in Figure 5.5 and the passive, active-assist, and safety modes of operation are demonstrated in Figure 7.5.

C – *Unconstrained Behaviour*

Constrained MPC: The nonlinear constrained MPC method directly addresses this problem by adjoining system state constraints to the MPC cost function, making the controller optimise for tracking error, robot torque, and system constraints simultaneously. The constraints enable multiple joint positions and velocities to be constrained in (non)linear relationships or (in)equalities. Simple cases of knee joint position and velocity constraints were demonstrated in simulation in Section 4.4.1.

D – *Disturbance Identification*

Human Torque in MPC: The problem of external disturbance being indistinguishable from human torque contribution, occurring in sensorless human torque estimation approaches, is addressed by estimating human joint torque from muscle activity signals and system states, as described in Section 6.2, and including it in the system models, as detailed in Section 4.2.4. The controller can effectively reject external disturbance, while recognising and adapting to human joint torque.

E – *Human Torque Estimation Latency*

Predictive Human Torque Estimation: Human joint torque estimation latency, present in sensorless approaches, is addressed by estimating human joint torque from muscle activity measurement, enabling

advance prediction, as shown in Figure 7.2. The controller can react rapidly to changes in neurological signals, instead of responding to the effects of human joint torque long after it is produced.

F – *Flawed Human Torque Estimation Assumptions*

Agnostic Human Torque Estimation Approach: A number of methods for estimating human joint torque from muscle activity signals are based on biomechanical models which assume healthy and conventional muscle form and behaviour. The HTE method developed in Chapter 6 does not assume conventional biomechanics and muscle condition. It uses machine learning to build and train a predictive model of human joint torque, using a carefully recorded dataset specific to the subject. The agnostic nature of this approach makes it suitable for neurological disorder and elderly patients who may not retain conventional muscle function.

8.3 Limitations

Whilst this work has made significant contributions to the field of AAN control for robotic assistance and rehabilitation, it is not without limitations. The main limitations are as follows:

Single Joint Actuation: The work in this thesis is limited to single joint actuation. This was necessary because only one actuator was included in the exoskeleton configurations. This limited the human joint torque estimation approach to considering only the main rotational degree of freedom in the knee joint. However, the approach can be applied to multi-joint and multi DOF human-exoskeleton systems, provided a representative system of dynamic equations can be derived.

Planar Systems: The human-exoskeleton system models for the exercises in this work are restricted to the sagittal plane. This simplifies modelling, reducing the number of DOFs and makes analysis of functionality and performance easier for each part of the control framework. The method can be applied with the same principles to system models that include out-of-plane DOFs.

Lower Body Exercises: The exercise tasks considered in this work are limited to the lower body. Although this application area was targeted, the control framework and related analysis are applicable to upper-body robotic assistance. A direct parallel to the knee joint is the elbow joint, as it has a single DOF, and the biceps and triceps muscles in the front and back of the upper arm behave with an agonist-antagonist relationship similar to muscles in the thigh.

Parameter Identification: The models for the human-exoskeleton system and human joint torque estimation require parametric identification to find model parameters that are representative of the real world. These processes are time-consuming and require informed cooperation from the human subject. The MPC component of the controller requires little tuning compared to traditional control approaches such as PID.

Limited Number of Test Subjects: Three subjects were recruited for the testing in this work. One subject was tested for the human joint torque estimation study in Chapter 6, three for the seated controller evaluation in Chapter 7, and one for the walking controller demonstration in Chapter 7. Because individual parametric identification is required for each subject, the process of preparing the control framework takes some time. Enough tests were conducted and data collected to provide a valid demonstration of the control framework and evidence of the key arguments presented in this thesis.

8.4 Future Work

There are a number of ways in which the presented work can be taken forward. In future work, controller tests could involve more general and complex trajectories. The approach could be applied to a more advanced scenario with a multi-DOF system model, such as application to multiple lower limb joints during walking or sit-to-stand exercises. Clinical testing with survivors of stroke or elderly subjects with mobility problems could provide interesting validation. Efficacy of the p_A parameter for variable assistance and the trajectory adaptation method could be investigated further. Controller variables of μ_A and μ_S from the FLA could provide metrics for assessing patient progress during rehabilitation. Routines for autonomously adjusting controller parameters p_A and R_{th} could be developed, optimising online to help the controller handle subject-specific range of movement and fatigue. The model-based nature of the controller makes it adaptable with relatively little tuning of parameters. The approach could be applied to a wide variety of upper and lower limb rehabilitation devices, given a representative system model can be derived. Assistance situations with complex muscle group involvements could be investigated and there is opportunity to develop more advanced models for human joint torque estimation.

Bibliography

[1] Q. Yang, X. Tong, L. Schieb, *et al.*, "Vital signs: Recent trends in stroke death rates - United States, 2000-2015," *Morb. Mortal. Wkly. Rep.*, 2017.

[2] R. Teasell, M. Mcrae, N. Foley, and A. Bhardwaj, "The incidence and consequences of falls in stroke patients during inpatient rehabilitation: Factors associated with high risk," *Arch. Phys. Med. Rehabil.*, 2002.

[3] R. Burge, B. Dawson-hughes, D. H. Solomon, J. B. Wong, A. King, and A. Tosteson, "Incidence and economic burden of osteoporosis-related fractures in the United States, 2005-2025," *J. Bone Miner. Res.*, 2007.

[4] J. Verghese, A. Levalley, C. B. Hall, M. J. Katz, A. F. Ambrose, and R. B. Lipton, "Epidemiology of gait disorders in community-residing older adults," *J. Am. Geriatr. Soc.*, 2006.

[5] Ö. Saka, A. Mcguire, and C. Wolfe, "Cost of stroke in the United Kingdom," *Age Ageing*, 2009.

[6] B. Indredavik, F. Bakke, S. A. Slørdahl, R. Rokseth, and L. L. Håheim, "Treatment in a combined acute and rehabilitation stroke unit: Which aspects are most important?" *Stroke*, 1999.

[7] G. Kwakkel, R. Van Peppen, R. C. Wagenaar, *et al.*, "Effects of augmented exercise therapy time after stroke: A meta-analysis," in *Stroke*, 2004.

[8] W. H. Gage, K. F. Zabjek, K. M. Sibley, A. Tang, D. Brooks, and W. E. Mcilroy, "Ambulatory monitoring of activity levels of individuals in the sub-acute stage following stroke: A case series," *J. Neuroeng. Rehabil.*, 2007.

[9] S. Levy, *SAFE: The Stroke Alliance for Europe*, 2010.

[10] S. H. Smith, P. Reilly, and A. M. Bull, "A musculoskeletal modelling approach to explain sit-to-stand difficulties in older people due to changes in muscle recruitment and movement strategies," *J. Biomech.*, 2020.

[11] M. A. Hughes, B. S. Myers, and M. L. Schenkman, "The role of strength in rising from a chair in the functionally impaired elderly," *J. Biomech.*, 1996.

[12] Y. R. Mao, X. Q. Wu, J. Li Zhao, *et al.*, "The crucial changes of sit-to-stand phases in subacute stroke survivors identified by movement decomposition analysis," *Front. Neurol.*, 2018.

[13] S. F. Atashzar, I. G. Polushin, and R. V. Patel, "Networked teleoperation with non-passive environment: Application to tele-rehabilitation," in *IEEE Int. Conf. Intell. Robot. Syst.*, 2012.

[14] J. H. Burridge, A. C. W. Lee, R. Turk, *et al.*, "Telehealth, Wearable Sensors, and the Internet: Will They Improve Stroke Outcomes Through Increased Intensity of Therapy, Motivation, and Adherence to Rehabilitation Programs?" In *J. Neurol. Phys. Ther.*, 2017.

[15] A. M. Dollar and H. Herr, "Lower Extremity Exoskeletons and Active Orthoses:Challenges and State-of-the-Art," *IEEE Trans. Robot.*, vol. 24, no. 1, pp. 144–158, 2008.

[16] W. Huo, S. Mohammed, J. C. Moreno, and Y. Amirat, "Lower Limb Wearable Robots for Assistance and Rehabilitation: A State of the Art," *IEEE Syst. J.*, vol. 10, no. 3, pp. 1068–1081, 2016.

[17] A. J. Young and D. P. Ferris, "State of the art and future directions for lower limb robotic exoskeletons," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 2, pp. 171–182, 2017.

[18] S. Hesse and D. Uhlenbrock, "A mechanized gait trainer for restoration of gait," *J. Rehabil. Res. Dev.*, 2000.

[19] H. Schmidt, C. Werner, R. Bernhardt, S. Hesse, and J. Krüger, "Gait rehabilitation machines based on programmable footplates," *J. Neuroeng. Rehabil.*, 2007.

[20] S. Maeshima, A. Osawa, D. Nishio, *et al.*, "Efficacy of a hybrid assistive limb in post-stroke hemiplegic patients: A preliminary report," *BMC Neurol.*, 2011.

[21] K. A. Strausser, T. A. Swift, A. B. Zoss, and H. Kazerooni, "Prototype medical exoskeleton for paraplegic mobility: First experimental results," in *ASME 2010 Dyn. Syst. Control Conf. DSCC2010*, 2010.

[22] M. Bortole, A. Venkatakrishnan, F. Zhu, *et al.*, "The H2 robotic exoskeleton for gait rehabilitation after stroke: Early findings from a clinical study Wearable robotics in clinical testing," *J. Neuroeng. Rehabil.*, vol. 12, no. 1, pp. 1–14, 2015.

[23] P. O. Riley, D. E. Krebs, and R. A. Popat, "Biomechanical analysis of failed sit-to-stand," *IEEE Trans. Rehabil. Eng.*, 1997.

[24] V. L. Feigin, G. A. Roth, M. Naghavi, *et al.*, "Global burden of stroke and risk factors in 188 countries, during 1990-2013: a systematic analysis for the Global Burden of Disease Study 2013," *Lancet Neurol.*, vol. 15, no. 9, pp. 913–924, 2016.

[25] W. H. Organization, *The atlas of heart disease and stroke / Judith Mackay and George Mensah ; with Shanthi Mendis and Kurt Greenland*, 2004.

[26] M. Nichols, N. Townsend, and M. Rayner, *European cardiovascular disease statistics*. British Heart Foundation, 2012, pp. 10–123.

[27] J. J. Day and J. D. Sweatt, *Epigenetic treatments for cognitive impairments*, 1. World Health Organization, 2012, vol. 37, pp. 247–260.

[28] S. Srivastava, P. C. Kao, S. H. Kim, *et al.*, "Assist-as-Needed Robot-Aided Gait Training Improves Walking Function in Individuals Following Stroke," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 23, no. 6, pp. 956–963, 2015.

[29] L. E. Kahn, M. L. Zygman, W. Z. Rymer, and D. J. Reinkensmeyer, "Robot-assisted reaching exercise promotes arm movement recovery in chronic hemiparetic stroke: A randomized controlled pilot study," *J. Neuroeng. Rehabil.*, vol. 3, 2006.

[30] W. H. Chang and Y.-h. Kim, "Robot-assisted Therapy in Stroke Rehabilitation," *J. Stroke*, vol. 15, no. 3, p. 174, 2013.

[31] R. Huang, H. Cheng, J. Qiu, and J. Zhang, "Learning Physical Human-Robot Interaction With Coupled Cooperative Primitives for a Lower Exoskeleton," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 4, pp. 1566–1574, 2019.

[32] J. Huang, W. Huo, W. Xu, S. Mohammed, and Y. Amirat, "Control of Upper-Limb Power-Assist Exoskeleton Using a Human-Robot Interface Based on Motion Intention Recognition," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 4, pp. 1257–1270, 2015.

[33] X. Wu, D.-x. Liu, M. Liu, C. Chen, and H. Guo, "Individualized Gait Pattern Generation for Sharing Lower Limb Exoskeleton Robot," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1459–1470, 2018.

[34] L. L. Cai, A. J. Fong, L. Yongqiang, J. Burdick, and V. R. Edgerton, "Assist-as-needed training paradigms for robotic rehabilitation of spinal cord injuries," in *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2006, IEEE, 2006, pp. 3504–3511.

[35] N. Hogan and H. I. Krebs, "Interactive robots for neuro-rehabilitation.," *Restor. Neurol. Neurosci.*, vol. 22, no. 3-5, pp. 349–58, 2004.

[36] J. L. Emken, R. Benitez, and D. J. Reinkensmeyer, "Human-robot cooperative movement training: Learning a novel sensory motor transformation during walking with robotic assistance-as-needed," *J. Neuroeng. Rehabil.*, vol. 4, no. 1, p. 8, 2007.

[37] S. Hussain, P. K. Jamwal, M. H. Ghayesh, and S. Q. Xie, "Assist-as-needed control of an intrinsically compliant robotic gait training orthosis," *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 1675–1685, 2017.

[38] J. Garrido, W. Yu, and A. Soria, "Modular design and modeling of an upper limb exoskeleton," in *5th IEEE RAS/EMBS Int. Conf. Biomed. Robot. Biomechatronics*, IEEE, 2014, pp. 508–513.

[39] R. Levi, *Neurological Rehabilitation*, 13-14. 2014, vol. 10, pp. 401–409.

[40] T. Proietti, V. Crocher, A. Roby-brami, and N. Jarrasse, "Upper-limb robotic exoskeletons for neurorehabilitation: A review on control strategies," *IEEE Rev. Biomed. Eng.*, vol. 9, pp. 4–14, 2016.

[41] A. M. Khan, D. W. Yun, M. A. Ali, J. Han, K. Shin, and C. Han, "Adaptive impedance control for upper limb assist exoskeleton," in *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2015-June, IEEE, 2015, pp. 4359–4366.

[42] J. C. Perez-ibarra, A. A. Siqueira, and H. I. Krebs, "Assist-As-needed ankle rehabilitation based on adaptive impedance control," *IEEE Int. Conf. Rehabil. Robot.*, vol. 2015-Septe, pp. 723–728, 2015.

[43] J. C. Pérez-ibarra, A. A. G. Siqueira, M. A. Silva-couto, *et al.*, "Adaptive Impedance Control Applied to Robot-Aided Neuro-Rehabilitation of the Ankle," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 185–192, 2019.

[44] S. Balasubramanian, R. Wei, M. Perez, *et al.*, "RUPERT: An exoskeleton robot for assisting rehabilitation of arm functions," in *2008 Virtual Rehabil.*, IEEE, 2008, pp. 163–167.

[45] J. L. Pons, Ed., *Wearable Robots*. Chichester, UK: John Wiley & Sons, Ltd, 2008.

[46] A. B. Zoss, H. Kazerooni, and A. Chu, "Biomechanical design of the Berkeley Lower Extremity Exoskeleton (BLEEX)," *IEEE/ASME Trans. Mechatronics*, vol. 11, no. 2, pp. 128–138, 2006.

[47] E. Guizzo and H. Goldstein, "The rise of the body bots: robotic exoskeletons," *IEEE Spectr.*, vol. 42, no. 10, pp. 50–56, 2005.

[48] R. Bogue, "Exoskeletons and robotic prosthetics: a review of recent developments," *Ind. Robot An Int. J.*, vol. 36, no. 5, pp. 421–427, 2009.

[49] W. Kim, H. Lee, D. Kim, J. Han, and C. Han, "Mechanical design of the Hanyang Exoskeleton Assistive Robot(HEXAR)," in *2014 14th Int. Conf. Control. Autom. Syst. (ICCAS 2014)*, IEEE, 2014, pp. 479–484.

[50] G. Zeilig, H. Weingarden, M. Zwecker, I. Dudkiewicz, A. Bloch, and A. Esquenazi, "Safety and tolerance of the ReWalk™ exoskeleton suit for ambulation by people with complete spinal cord injury: a pilot study.," *J. Spinal Cord Med.*, vol. 35, no. 2, pp. 96–101, 2012.

[51] S. A. Kolakowsky-hayner, "Safety and Feasibility of using the EksoTM Bionic Exoskeleton to Aid Ambulation after Spinal Cord Injury," *J. Spine*, 2013.

[52] M. Bernhardt, M. Frey, G. Colombo, and R. Riener, "Hybrid Force-Position Control Yields Cooperative Behaviour of the Rehabilitation Robot Lokomat," in *9th Int. Conf. Rehabil. Robot. 2005. ICORR 2005.*, IEEE, 2005, pp. 536–539.

[53] M. Fontana, R. Vertechy, S. Marcheschi, F. Salsedo, and M. Bergamasco, "The body extender: A full-body exoskeleton for the transport and handling of heavy loads," *IEEE Robot. Autom. Mag.*, vol. 21, no. 4, pp. 34–44, 2014.

[54] M. Vukobratović, B. Borovac, D. Surla, and D. Stokić, *Biped Locomotion*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990.

[55] S. Toyama and G. Yamamoto, "Development of Wearable-Agri-Robot mechanism for agricultural work," in *2009 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, IEEE, 2009, pp. 5801–5806.

[56] K. S. Stadler, W. J. Elspass, and H. W. VAN DE VENN, "ROBO-MATE: EXOSKELETON TO ENHANCE INDUSTRIAL PRODUCTION," in *Mob. Serv. Robot.*, WORLD SCIENTIFIC, 2014, pp. 53–60.

[57] W. Honig, S. Kiesel, A. Tinka, J. W. Durham, and N. Ayanian, "Persistent and Robust Execution of MAPF Schedules in Warehouses," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1125–1131, 2019.

[58] D. Sykes and G. Keighren, "Industrial-scale environments with bounded uncertainty," *2018 IEEE/ACM 1st Int. Work. Robot. Softw. Eng.*, pp. 29–32, 2018.

[59] B. Dynamics, *Home — Boston Dynamics*, 2020. (visited on 04/21/2020).

[60] D. N. Condie, "The modern era of orthotics," *Prosthet. Orthot. Int.*, vol. 32, no. 3, pp. 313–323, 2008.

[61] A. Esquenazi, M. Talaty, A. Packel, and M. Saulino, "The Rewalk powered exoskeleton to restore ambulatory function to individuals with thoracic-level motor-complete spinal cord injury," *Am. J. Phys. Med. Rehabil.*, vol. 91, no. 11, pp. 911–921, 2012.

[62] R. J. Farris, H. A. Quintero, and M. Goldfarb, "Preliminary evaluation of a powered lower limb orthosis to aid walking in paraplegic individuals.," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 19, no. 6, pp. 652–9, 2011.

[63] A. Kilicarslan, S. Prasad, R. G. Grossman, and J. L. Contreras-vidal, "High accuracy decoding of user intentions using EEG to control a lower-body exoskeleton," in *2013 35th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, IEEE, 2013, pp. 5606–5609.

[64] J. Gancet, M. Ilzkovitz, E. Motard, *et al.*, "MINDWALKER: Going one step further with assistive lower limbs exoskeleton for SCI condition subjects," in *2012 4th IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatronics*, IEEE, 2012, pp. 1794–1800.

[65] G. R. Andrews, "Care of older people: Promoting health and function in an ageing population," *BMJ*, vol. 322, no. 7288, pp. 728–729, 2001.

[66] G. S. Virk, U. Haider, I. N. Indrawibawa, R. K. Thekkeparampumadom, and N. Masud, "EXO-LEGS FOR ELDERLY PERSONS," in *Mob. Serv. Robot.*, WORLD SCIENTIFIC, 2014, pp. 85–92.

[67] B. S. Rupal, S. Rafique, A. Singla, E. Singla, M. Isaksson, and G. S. Virk, "Lower-limb exoskeletons: Research trends and regulatory guidelines in medical and non-medical applications," *Int. J. Adv. Robot. Syst.*, vol. 14, no. 6, pp. 1–27, 2017.

[68] C. Morawietz and F. Moffat, "Effects of locomotor training after incomplete spinal cord injury: A systematic review," *Arch. Phys. Med. Rehabil.*, vol. 94, no. 11, pp. 2297–2308, 2013.

[69] P. Sale, M. Franceschini, A. Waldner, and S. Hesse, *Use of the robot assisted gait therapy in rehabilitation of patients with stroke and spinal cord injury*, 2012.

[70] D. S. Nichols-larsen, P. C. Clark, A. Zeringue, A. Greenspan, and S. Blanton, "Factors influencing stroke survivors' quality of life during subacute recovery," *Stroke*, vol. 36, no. 7, pp. 1480–1484, 2005.

[71] G. Chen, C. K. Chan, Z. Guo, and H. Yu, "A Review of Lower Extremity Assistive Robotic Exoskeletons in Rehabilitation Therapy," *Crit. Rev. Biomed. Eng.*, vol. 41, no. 4-5, pp. 343–363, 2013.

[72] S. Jezernik, G. Colombo, T. Keller, H. Frueh, and M. Morari, "Robotic Orthosis Lokomat: A Rehabilitation and Research Tool," *Neuromodulation*, vol. 6, no. 2, pp. 108–115, 2003.

[73] A. Mayr, M. Kofler, E. Quirbach, H. Matzak, K. Fröhlich, and L. Saltuari, "Prospective, blinded, randomized crossover study of gait rehabilitation in stroke patients using the Lokomat gait orthosis," *Neurorehabil. Neural Repair*, vol. 21, no. 4, pp. 307–314, 2007.

[74] J. Hidler, D. Nichols, M. Pelliccio, *et al.*, "Multicenter randomized clinical trial evaluating the effectiveness of the Lokomat in subacute stroke," *Neurorehabil. Neural Repair*, vol. 23, no. 1, pp. 5–13, 2009.

[75] M. Alcobendas-maestro, A. Esclarín-ruz, R. M. Casado-lópez, *et al.*, "Lokomat Robotic-Assisted Versus Overground Training Within 3 to 6 Months of Incomplete Spinal Cord Lesion: Randomized Controlled Trial," *Neurorehabil. Neural Repair*, vol. 26, no. 9, pp. 1058–1063, 2012.

[76] K. Y. Nam, H. J. Kim, B. S. Kwon, J. W. Park, H. J. Lee, and A. Yoo, "Robot-assisted gait training (Lokomat) improves walking function and activity in people with spinal cord injury: a systematic review," *J. Neuroeng. Rehabil.*, vol. 14, no. 1, pp. 1–13, 2017.

[77] K. Raab, K. Krakow, F. Tripp, and M. Jung, "Effects of training with the ReWalk exoskeleton on quality of life in incomplete spinal cord injury: a single case study," *Spinal Cord Ser. Cases*, vol. 2, no. 1, pp. 1–3, 2016.

[78] J. Pransky, "The Pransky interview: Russ Angold, Co-Founder and President of Ekso(TM) Labs," *Ind. Rob.*, vol. 41, no. 4, p. 329, 2014.

[79] P. Milia, F. De Salvo, M. Caserio, *et al.*, "Neurorehabilitation in paraplegic patients with an active powered exoskeleton (Ekso)," *Digit. Med.*, vol. 2, no. 4, p. 163, 2016.

[80] A. Esquenazi, M. Talaty, and A. Jayaraman, "Powered Exoskeletons for Walking Assistance in Persons with Central Nervous System Injuries: A Narrative Review," *PM&R*, vol. 9, no. 1, pp. 46–62, 2017.

[81] P. Sale, E. F. Russo, M. Russo, *et al.*, "Effects on mobility training and de-adaptations in subjects with Spinal Cord Injury due to a Wearable Robot: A preliminary report," *BMC Neurol.*, vol. 16, no. 1, pp. 2–9, 2016.

[82] C. Buesing, G. Fisch, M. O'donnell, *et al.*, "Effects of a wearable exoskeleton stride management assist system (SMA) on spatiotemporal gait characteristics in individuals after stroke: a randomized controlled trial," *J. Neuroeng. Rehabil.*, vol. 12, no. 1, p. 69, 2015.

[83] A. Nilsson, K. Vreede, V. Häglund, H. Kawamoto, Y. Sankai, and J. Borg, "Gait training early after stroke with a new exoskeleton – the hybrid assistive limb: a study of safety and feasibility," *J. Neuroeng. Rehabil.*, vol. 11, no. 1, p. 92, 2014.

[84] J. Stein, L. Bishop, D. J. Stein, and C. K. Wong, "Gait Training with a Robotic Leg Brace After Stroke," *Am. J. Phys. Med. Rehabil.*, vol. 93, no. 11, pp. 987–994, 2014.

[85] D. R. Louie and J. J. Eng, "Powered robotic exoskeletons in post-stroke rehabilitation of gait: A scoping review," *J. Neuroeng. Rehabil.*, vol. 13, no. 1, pp. 1–10, 2016.

[86] I. Díaz, J. J. Gil, and E. Sánchez, "Lower-Limb Robotic Rehabilitation: Literature Review and Challenges," *J. Robot.*, vol. 2011, no. i, pp. 1–11, 2011.

[87] S. Dehem, T. Lejeune, G. Stoquart, *et al.*, "Robotic-assisted serious game for motor and cognitive post-stroke rehabilitation," *2017 IEEE 5th Int. Conf. Serious Games Appl. Heal. SeGAH 2017*, no. ii, pp. 1–8, 2017.

[88] M. Ghassemi, K. Triandafilou, A. Barry, *et al.*, "Development of an EMG-Controlled Serious Game for Rehabilitation," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 2, pp. 283–292, 2019.

[89] H. Yu, S. Huang, G. Chen, Y. Pan, and Z. Guo, "Human-Robot Interaction Control of Rehabilitation Robots with Series Elastic Actuators," *IEEE Trans. Robot.*, vol. 31, no. 5, 2015.

[90] C. Jagger, C. Gillies, F. Moscone, *et al.*, "Inequalities in healthy life years in the 25 countries of the European Union in 2005: a cross-national meta-regression analysis," *Lancet*, vol. 372, no. 9656, pp. 2124–2131, 2008.

[91] E. J. Benjamin, P. Muntner, A. Alonso, *et al.*, "Heart Disease and Stroke Statistics-2019 Update: A Report From the American Heart Association," *Circulation*, vol. 139, no. 10, 2019.

[92] Y. Béjot, H. Bailly, J. Durier, and M. Giroud, "Epidemiology of stroke in Europe and trends for the 21st century," *Press. Medicale*, vol. 45, no. 12, e391–e398, 2016.

[93] S. Wist, J. Clivaz, and M. Sattelmayer, "Muscle strengthening for hemiparesis after stroke: A meta-analysis," *Ann. Phys. Rehabil. Med.*, vol. 59, no. 2, pp. 114–124, 2016.

[94] R. N. Kalaria, R. Akinyemi, and M. Ihara, "Stroke injury, cognitive impairment and vascular dementia," *Biochim. Biophys. Acta - Mol. Basis Dis.*, vol. 1862, no. 5, pp. 915–925, 2016.

[95] W. Johnson, O. Onuma, M. Owolabi, and S. Sachdev, "Stroke: a global response is needed," *Bull. World Health Organ.*, vol. 94, no. 9, 634–634A, 2016.

[96] W. H. Organization, *International Perspectives on Spinal Cord Injury*. 2003.

[97] M. J. Devivo, "Epidemiology of traumatic spinal cord injury: Trends and future implications," *Spinal Cord*, vol. 50, no. 5, pp. 365–372, 2012.

[98] N. B. Jain, G. D. Ayers, E. N. Peterson, *et al.*, "Traumatic Spinal Cord Injury in the United States, 1993-2012," *JAMA*, vol. 313, no. 22, p. 2236, 2015.

[99] M. J. Eckert and M. J. Martin, "Trauma: Spinal Cord Injury," *Surg. Clin. North Am.*, vol. 97, no. 5, pp. 1031–1045, 2017.

[100] R. B. Van Dijsseldonk, L. A. F. De Jong, B. E. Groen, M. Vos-van der Hulst, A. C. H. Geurts, and N. L. W. Keijsers, "Gait Stability Training in a Virtual Environment Improves Gait and Dynamic Balance Capacity in Incomplete Spinal Cord Injury Patients," *Front. Neurol.*, vol. 9, 2018.

[101] A. O. Sasmita, J. Kuruvilla, and A. P. K. Ling, "Harnessing neuroplasticity: modern approaches and clinical future," *Int. J. Neurosci.*, vol. 128, no. 11, pp. 1061–1077, 2018.

[102] T. K. Hensch and P. M. Bilimoria, "Re-opening Windows: Manipulating Critical Periods for Brain Development.," *Cerebrum*, vol. 2012, no. August, p. 11, 2012.

[103] A. K. H. Miller, R. L. Alston, and J. A. N. Corsellis, "VARIATION WITH AGE IN THE VOLUMES OF GREY AND WHITE MATTER IN THE CEREBRAL HEMISPHERES OF MAN: MEASUREMENTS WITH AN IMAGE ANALYSER," *Neuropathol. Appl. Neurobiol.*, vol. 6, no. 2, pp. 119–132, 1980.

[104] M. Maier, B. R. Ballester, and P. F. Verschure, "Principles of Neurorehabilitation After Stroke Based on Motor Learning and Brain Plasticity Mechanisms," *Front. Syst. Neurosci.*, vol. 13, no. December, pp. 1–18, 2019.

[105] M. O. Krucoff, S. Rahimpour, M. W. Slutzky, V. R. Edgerton, and D. A. Turner, "Enhancing Nervous System Recovery through Neurobiologics, Neural Interface Training, and Neurorehabilitation," *Front. Neurosci.*, vol. 10, 2016.

[106] K. Ganguly, N. N. Byl, and G. M. Abrams, "Neurorehabilitation: motor recovery after stroke as an example," *Ann. Neurol.*, vol. 74, no. 3, pp. 373–381, 2013.

[107] S. C. Cramer, "Repairing the human brain after stroke: I. Mechanisms of spontaneous recovery," *Ann. Neurol.*, vol. 63, no. 3, pp. 272–287, 2008.

[108] T. Kitago, J. Liang, V. S. Huang, *et al.*, "Improvement after constraint-induced movement therapy: Recovery of normal motor control or task-specific compensation?" *Neurorehabil. Neural Repair*, vol. 27, no. 2, pp. 99–109, 2013.

[109] K. Doya, *Complementary roles of basal ganglia and cerebellum in learning and motor control*, 2000.

[110] A. J. Dinnerstein, T. Frigyesi, and M. Lowenthal, "Delayed Feedback as a Possible Mechanism in Parkinsonism," *Percept. Mot. Skills*, vol. 15, no. 3, pp. 667–680, 1962.

[111] K. P. Tee, D. W. Franklin, M. Kawato, T. E. Milner, and E. Burdet, "Concurrent adaptation of force and impedance in the redundant muscle system," *Biol. Cybern.*, 2010.

[112] L. Piron, A. Turolla, M. Agostini, *et al.*, "Motor learning principles for rehabilitation: A pilot randomized controlled study in poststroke patients," *Neurorehabil. Neural Repair*, vol. 24, no. 6, pp. 501–508, 2010.

[113] S. L. Wolf, C. J. Winstein, J. P. Miller, *et al.*, "Effect of constraint-induced movement therapy on upper extremity function 3 to 9 months after stroke: The EXCITE randomized clinical trial," *J. Am. Med. Assoc.*, 2006.

[114] N. Dancause, S. Barbay, S. B. Frost, *et al.*, "Extensive cortical rewiring after brain injury," *J. Neurosci.*, 2005.

[115] S. M. Hatem, G. Saussez, M. Della Faille, *et al.*, "Rehabilitation of motor function after stroke: A multiple systematic review focused on techniques to stimulate upper extremity recovery," *Front. Hum. Neurosci.*, 2016.

[116] J. W. Krakauer, *Motor learning: Its relevance to stroke recovery and neurorehabilitation*, 2006.

[117] S. C. Cramer, "Intense rehabilitation therapy produces very large gains in chronic stroke," *J. Neurol. Neurosurg. Psychiatry*, vol. 90, no. 5, pp. 497–497, 2019.

[118] E. Taub, *New discovery equals change in clinical practice*, 1999.

[119] N. A. Bayona, J. Bitensky, K. Salter, and R. Teasell, "The role of task-specific training in rehabilitation therapies," *Top. Stroke Rehabil.*, vol. 12, no. 3, pp. 58–65, 2005.

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

[121] P. Langhorne, J. Bernhardt, and G. Kwakkel, *Stroke rehabilitation*, 2011.

[122] E. Taub, G. Uswatte, and T. Elbert, "New treatments in neurorehabiliation founded on basic research," *Nat. Rev. Neurosci.*, 2002.

[123] M. G. Bowden, M. L. Woodbury, and P. W. Duncan, "Promoting neuroplasticity and recovery after stroke : future directions for rehabilitation clinical trials," pp. 37–42, 2013.

[124] B. H. Dobkin, *Motor rehabilitation after stroke, traumatic brain, and spinal cord injury: Common denominators within recent clinical trials*, 2009.

[125] J. Perry, M. Garrett, J. K. Gronley, and S. J. Mulroy, "Classification of Walking Handicap in the Stroke Population," *Stroke*, vol. 26, no. 6, pp. 982–989, 1995.

[126] L. F. Teixeira-salmela, S. J. Olney, S. Nadeau, and B. Brouwer, "Muscle strengthening and physical conditioning to reduce impairment and disability in chronic stroke survivors," *Arch. Phys. Med. Rehabil.*, vol. 80, no. 10, pp. 1211–1218, 1999.

[127] N. F. Gordon, M. Gulanick, F. Costa, *et al.*, "Physical Activity and Exercise Recommendations for Stroke Survivors," *Circulation*, vol. 109, no. 16, pp. 2031–2041, 2004.

[128] P. W. Duncan, L. B. Goldstein, D. Matchar, G. W. Divine, and J. Feussner, "Measurement of motor recovery after stroke outcome assessment and sample size requirements," *Stroke*, 1992.

[129] G. E. Berrios and M. Grli, "Abulia and impulsiveness revisited: A conceptual history," *Acta Psychiatrica Scandinavica*, vol. 92, no. 3, pp. 161–167, 1995.

[130] P. W. Duncan, K. J. Sullivan, A. L. Behrman, *et al.*, "Body-weight - Supported treadmill rehabilitation after stroke," *N. Engl. J. Med.*, 2011.

[131] A. C. Lo, P. D. Guarino, L. G. Richards, *et al.*, "Robot-assisted therapy for long-term upper-limb impairment after stroke," *N. Engl. J. Med.*, 2010.

[132] S. E. Nadeau, S. S. Wu, B. H. Dobkin, *et al.*, "Effects of task-specific and impairment-based training compared with usual care on functional walking ability after inpatient stroke rehabilitation: Leaps trial," *Neurorehabil. Neural Repair*, vol. 27, no. 4, pp. 370–380, 2013.

[133] M. Brainin and R. D. Zorowitz, "Advances in Stroke," *Stroke*, vol. 44, no. 2, pp. 311–313, 2013.

[134] E. T. Wolbrecht, V. Chan, D. J. Reinkensmeyer, and J. E. Bobrow, "Optimizing compliant, model-based robotic assistance to promote neurorehabilitation," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 16, no. 3, pp. 286–297, 2008.

[135] N. Hogan, "Impedance Control: An Approach to Manipulation," in *1984 Am. Control Conf.*, IEEE, 1984, pp. 304–313.

[136] A. Roy, H. Krebs, D. Williams, *et al.*, "Robot-Aided Neurorehabilitation: A Novel Robot for Ankle Rehabilitation," *IEEE Trans. Robot.*, vol. 25, no. 3, pp. 569–582, 2009.

[137] H. I. Krebs, N. Hogan, M. L. Aisen, and B. T. Volpe, "Robot-Aided Neurorehabilitation," vol. 6, no. 1, pp. 75–87, 1998.

[138] S. Jezernik, G. Colombo, and M. Morari, "Automatic Gait-Pattern Adaptation Algorithms for Rehabilitation With a 4-DOF Robotic Orthosis," *IEEE Trans. Robot. Autom.*, vol. 20, no. 3, pp. 574–582, 2004.

[139] Y. H. Tsoi and S. Q. Xie, "Design and Control of a Parallel Robot for Ankle Rehabiltation," in *2008 15th Int. Conf. Mechatronics Mach. Vis. Pract.*, IEEE, 2008, pp. 515–520.

[140] H. I. Krebs, J. J. Palazzolo, L. Dipietro, *et al.*, "Rehabilitation robotics: Performance-based progressive robot-assisted therapy," *Auton. Robots*, vol. 15, no. 1, pp. 7–20, 2003.

[141] T. Nef, M. Mihelj, and R. Riener, "ARMin: A robot for patient-cooperative arm therapy," *Med. Biol. Eng. Comput.*, vol. 45, no. 9, pp. 887–900, 2007.

[142] W. Yu, J. Rosen, and X. Li, "PID admittance control for an upper limb exoskeleton," *Proc. Am. Control Conf.*, pp. 1124–1129, 2011.

[143] P. Staubli, T. Nef, V. Klamroth-marganska, and R. Riener, "Effects of intensive arm training with the rehabilitation robot ARMin II in chronic stroke patients: Four single-cases," *J. Neuroeng. Rehabil.*, vol. 6, no. 1, 2009.

[144] B. C. Tsai, W. W. Wang, L. C. Hsu, L. C. Fu, and J. S. Lai, "An articulated rehabilitation robot for upper limb physiotherapy and training," *IEEE/RSJ 2010 Int. Conf. Intell. Robot. Syst. IROS 2010 - Conf. Proc.*, pp. 1470–1475, 2010.

[145] A. Frisoli, F. Salsedo, M. Bergamasco, B. Rossi, and M. C. Carboncini, "A force-feedback exoskeleton for upper-limb rehabilitation in virtual reality," *Appl. Bionics Biomech.*, vol. 6, no. 2, pp. 115–126, 2009.

[146] M. Mihelj, T. Nef, and R. Riener, "A novel paradigm for patient-cooperative control of upper-limb rehabilitation robots," *Adv. Robot.*, vol. 21, no. 8, pp. 843–867, 2007.

[147] H. J. Asl, T. Narikiyo, and M. Kawanishi, "An assist-as-needed control scheme for robot-assisted rehabilitation," in *Proc. Am. Control Conf.*, IEEE, 2017, pp. 198–203.

[148] C. Carignan, J. Tang, and S. Roderick, "Development of an exoskeleton haptic interface for virtual task training," *2009 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS 2009*, pp. 3697–3702, 2009.

[149] A. Gupta and M. K. O'malley, "Design of a haptic arm exoskeleton for training and rehabilitation," *IEEE/ASME Trans. Mechatronics*, vol. 11, no. 3, pp. 280–289, 2006.

[150] L. M. Miller and J. Rosen, "Comparison of multi-sensor admittance control in joint space and task space for a seven degree of freedom upper limb exoskeleton," *2010 3rd IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatronics, BioRob 2010*, pp. 70–75, 2010.

[151] J. C. Ibarra and A. A. Siqueira, "Impedance control of rehabilitation robots for lower limbs. Review," *Proc. - 2nd SBR Brazilian Robot. Symp. 11th LARS Lat. Am. Robot. Symp. 6th Rob. Work. Appl. Robot. Autom. SBR LARS Rob. 2014 - Part Jt. Conf. Robot. Intell. Syst.*, pp. 235–240, 2015.

[152] S. Jezernik, K. Jezernik, and M. Morari, "Impedance Control Based Gait-Pattern Adaptation for a Robotic Rehabilitation Device," *IFAC Proc. Vol.*, 2002.

[153] J. Yoon, J. Ryu, and K. B. Lim, "Reconfigurable ankle rehabilitation robot for various exercises," *J. Robot. Syst.*, 2006.

[154] J. C. Pérez-ibarra, A. A. Siqueira, J. C. Perez-ibarra, A. A. Siqueira, J. C. Pérez-ibarra, and A. A. Siqueira, "Comparison of kinematic and EMG parameters between unassisted, fixed- and adaptive-stiffness robotic-assisted ankle movements in post-stroke subjects," *IEEE Int. Conf. Rehabil. Robot.*, pp. 461–466, 2017.

[155] Y. Du, H. Wang, S. Qiu, W. Yao, P. Xie, and X. Chen, "An advanced adaptive control of lower limb rehabilitation robot," *Front. Robot. AI*, vol. 5, no. OCT, pp. 1–11, 2018.

[156] A. Duschau-wicke, J. Von Zitzewitz, A. Caprez, L. Lünenburger, and R. Riener, "Path control: A method for patient-cooperative robot-aided gait rehabilitation," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 18, no. 1, pp. 38–48, 2010.

[157] S. K. Banala, S. H. Kim, S. K. Agrawal, and J. P. Scholz, "Robot assisted gait training with active leg exoskeleton (ALEX)," *Proc. 2nd Bienn. IEEE/RAS-EMBS Int. Conf. Biomed. Robot. Biomechatronics, BioRob 2008*, vol. 17, no. 1, pp. 653–658, 2009.

[158] S. K. Banala, S. K. Agrawal, and J. P. Scholz, "Active Leg Exoskeleton (ALEX) for gait rehabilitation of motor-impaired patients," *2007 IEEE 10th Int. Conf. Rehabil. Robot. ICORR'07*, vol. 00, no. c, pp. 401–407, 2007.

[159] P. Stegall, D. Zanotto, and S. K. Agrawal, "Variable Damping Force Tunnel for Gait Training Using ALEX III," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1495–1501, 2017.

[160] P. Agarwal and A. D. Deshpande, "Subject-Specific Assist-as-Needed Controllers for a Hand Exoskeleton for Rehabilitation," *IEEE Robot. Autom. Lett.*, vol. 3, no. 1, pp. 508–515, 2018.

[161] P. Y. Li and R. Horowitz, "Passive velocity field control of mechanical manipulators," *IEEE Trans. Robot. Autom.*, vol. 15, no. 4, pp. 751–763, 1999.

[162] H. J. Asl, T. Narikiyo, and M. Kawanishi, "An Assist-as-Needed Velocity Field Control Scheme for Rehabilitation Robots," in *IEEE Int. Conf. Intell. Robot. Syst.*, IEEE, 2018, pp. 3322–3327.

[163] A. Martinez, B. Lawson, C. Durrough, and M. Goldfarb, "A Velocity-Field-Based Controller for Assisting Leg Movement during Walking with a Bilateral Hip and Knee Lower Limb Exoskeleton," *IEEE Trans. Robot.*, vol. 35, no. 2, pp. 307–316, 2019.

[164] I. Cervantes, R. Kelly, J. Alvarez-ramirez, and J. Moreno, "A robust velocity field control," *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 6, pp. 888–894, 2002.

[165] H. J. Asl, T. Narikiyo, and M. Kawanishi, "Neural network velocity field control of robotic exoskeletons with bounded input," *IEEE/ASME Int. Conf. Adv. Intell. Mechatronics, AIM*, pp. 1363–1368, 2017.

[166] H. J. Asl and T. Narikiyo, "An assistive control strategy for rehabilitation robots using velocity field and force field," *IEEE Int. Conf. Rehabil. Robot.*, vol. 2019-June, pp. 790–795, 2019.

[167] H. Zhang, H. Austin, S. Buchanan, R. Herman, J. Koeneman, and J. He, "Feasibility study of robot-assisted stroke rehabilitation at home using RUPERT," *2011 IEEE/ICME Int. Conf. Complex Med. Eng. C. 2011*, pp. 604–609, 2011.

[168] F. Meng and Y. Dai, "Adaptive Inverse Optimal Control for Rehabilitation Robot Systems Using Actor-Critic Algorithm," *Math. Probl. Eng.*, vol. 2014, 2014.

[169] T. Proietti, N. Jarrassé, A. Roby-brami, and G. Morel, "Adaptive control of a robotic exoskeleton for neurorehabilitation," *Int. IEEE/EMBS Conf. Neural Eng. NER*, vol. 2015-July, no. 1, pp. 803–806, 2015.

[170] A. U. Pehlivan, D. P. Losey, and M. K. Omalley, "Minimal Assist-as-Needed Controller for Upper Limb Robotic Rehabilitation," *IEEE Trans. Robot.*, vol. 32, no. 1, pp. 113–124, 2016.

[171] A. U. Pehlivan, F. Sergi, and M. K. Omalley, "A subject-adaptive controller for wrist robotic rehabilitation," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 3, pp. 1338–1350, 2015.

[172] R. Pérez-rodríguez, C. Rodríguez, Ú. Costa, *et al.*, "Anticipatory assistance-as-needed control algorithm for a multijoint upper limb robotic orthosis in physical neurorehabilitation," *Expert Syst. Appl.*, vol. 41, no. 8, pp. 3922–3934, 2014.

[173] J. Chen, X. Zhang, L. Gu, and C. Nelson, "Estimating Muscle Forces and Knee Joint Torque using Surface Electromyography: A Musculoskeletal Biomechanical Model," *J. Mech. Med. Biol.*, vol. 17, no. 04, p. 1750 069, 2017.

[174] M. A. Gomes, A. A. G. Siqueira, and R. G. Gobbo, "Improving the parameters of neural oscillators to generate joint trajectories of an exoskeleton for lower limbs," *IEEE Int. Conf. Control Autom. ICCA*, pp. 286–291, 2011.

[175] D. G. Lloyd and T. F. Besier, "An EMG-driven musculoskeletal model to estimate muscle forces and knee joint moments in vivo," *J. Biomech.*, 2003.

[176] W. Huo, S. Mohammed, and Y. Amirat, "Impedance Reduction Control of a Knee Joint Human-Exoskeleton System," *IEEE Trans. Control Syst. Technol.*, vol. PP, pp. 1–16, 2018.

[177] R. Huang, H. Cheng, Y. Chen, Q. Chen, X. Lin, and J. Qiu, "Optimisation of Reference Gait Trajectory of a Lower Limb Exoskeleton," *Int. J. Soc. Robot.*, vol. 8, no. 2, pp. 223–235, 2016.

[178] H. Rifai, S. Mohammed, K. Djouani, and Y. Amirat, "Toward Lower Limbs Functional Rehabilitation Through a Knee-Joint Exoskeleton," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 2, pp. 712–719, 2017.

[179] R. Dumas, L. Chèze, and J.-p. Verriest, "Adjustments to McConville et al. and Young et al. body segment inertial parameters," *J. Biomech.*, vol. 40, no. 3, pp. 543–553, 2007.

[180] O. Ozen, F. Traversa, S. Gadi, K. A. Buetler, T. Nef, and L. Marchal-crespo, "Multi-purpose Robotic Training Strategies for Neurorehabilitation with Model Predictive Controllers," in *2019 IEEE 16th Int. Conf. Rehabil. Robot.*, IEEE, 2019, pp. 754–759.

[181] S. Y. Raza, S. F. Ahmed, A. Ali, *et al.*, "Model Predictive Control for Upper Limb Rehabilitation Robotic System Under Noisy Condition," in *2018 IEEE 5th Int. Conf. Smart Instrumentation, Meas. Appl.*, IEEE, 2018, pp. 1–4.

[182] T. Teramae, T. Noda, and J. Morimoto, "EMG-Based Model Predictive Control for Physical Human–Robot Interaction: Application for Assist-As-Needed Control," *IEEE Robot. Autom. Lett.*, vol. 3, no. 1, pp. 210–217, 2018.

[183] T. Englert, A. Völz, F. Mesmer, S. Rhein, and K. Graichen, "A software framework for embedded nonlinear model predictive control using a gradient-based augmented Lagrangian approach (GRAMPC)," *Optim. Eng.*, vol. 20, no. 3, pp. 769–809, 2019.

[184] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[185] L. Grüne, "Economic receding horizon control without terminal constraints," *Automatica*, vol. 49, no. 3, pp. 725–734, 2013.

[186] K. Graichen, "A fixed-point iteration scheme for real-time model predictive control," *Automatica*, vol. 48, no. 7, pp. 1300–1305, 2012.

[187] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, Inc., 1996, p. 410.

[188] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II*. Springer Series in Computational Mathematics, 1996.

[189] E. J. Vos, J. Harlaar, and I. S. Van, "Electromechanical delay during knee extensor contractions. / Reponse electromecanique lors de contractions des muscles extenseurs du genou," *Med. & Sci. Sport. & Exerc.*, 1991.

[190] A. E. Jackson, R. J. Holt, P. R. Culmer, *et al.*, "Dual robot system for upper limb rehabilitation after stroke: The design process," *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.*, 2007.

[191] V. Novák, I. Perfilieva, and J. Močkoř, *Mathematical Principles of Fuzzy Logic*. 1999.

[192] M. S. Ju, C. C. K. Lin, D. H. Lin, I. S. Hwang, and S. M. Chen, "A rehabilitation robot with force-position hybrid fuzzy controller: Hybrid fuzzy control of rehabilitation robot," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 13, no. 3, pp. 349–358, 2005.

[193] T. Watanabe, T. Masuko, A. Arifin, and M. Yoshizawa, "A feasibility study of fuzzy FES controller based on cycle-to-cycle control: An experimental test of knee extension control," *IEICE Trans. Inf. Syst.*, vol. E91-D, no. 3, pp. 865–868, 2008.

[194] R. Jailani, M. O. Tokhi, S. C. Gharooni, and M. Jogtaei, "Finite state control of FES-assisted walking with spring brake orthosis," *Proc. - 2011 UKSim 13th Int. Conf. Model. Simulation, UKSim 2011*, pp. 183–188, 2011.

[195] Z. Li, C. Y. Su, G. Li, and H. Su, "Fuzzy approximation-based adaptive backstepping control of an exoskeleton for human upper limbs," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 3, pp. 555–566, 2015.

[196] B. O. Mushage, J. C. Chedjou, and K. Kyamakya, "Fuzzy neural network and observer-based fault-tolerant adaptive nonlinear control of uncertain 5-DOF upper-limb exoskeleton robot for passive rehabilitation," *Nonlinear Dyn.*, pp. 1–17, 2016.

[197] K. Kong and M. Tomizuka, "A Gait Monitoring System Based on Air Pressure Sensors Embedded in a Shoe," *IEEE/ASME Trans. Mechatronics*, vol. 14, no. 3, pp. 358–370, 2009.

[198] G. Shi, G. Xu, H. Wang, N. Duan, and S. Zhang, "Fuzzy-adaptive Impedance Control of Upper Limb Rehabilitation Robot Based on sEMG *," in *2019 16th Int. Conf. Ubiquitous Robot.*, IEEE, 2019, pp. 745–749.

[199] K. Kiguchi and Y. Hayashi, "An EMG-based control for an upper-limb power-assist exoskeleton robot," *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 42, no. 4, pp. 1064–1071, 2012.

[200] P. Martin Larsen, "Industrial applications of fuzzy logic control," *Int. J. Man. Mach. Stud.*, vol. 12, no. 1, pp. 3–10, 1980.

[201] R. B. Woodward, S. J. Shefelbine, and R. Vaidyanathan, "Pervasive Monitoring of Motion and Muscle Activation: Inertial and Mechanomyography Fusion," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 5, pp. 2022–2033, 2017.

[202] J. P. Van Dijk, M. M. Lowery, B. G. Lapatki, and D. F. Stegeman, "Evidence of potential averaging over the finite surface of a bioelectric surface electrode," *Ann. Biomed. Eng.*, 2009.

[203] I. I. Christov and I. K. Daskalov, "Filtering of electromyogram artifacts from the electrocardiogram," *Med. Eng. Phys.*, 1999.

[204] D. Staudenmann, K. Roeleveld, D. F. Stegeman, and J. H. Van Dieën, "Methodological aspects of SEMG recordings for force estimation – A tutorial and review," *J. Electromyogr. Kinesiol.*, vol. 20, no. 3, pp. 375–387, 2010.

[205] D. Staudenmann, I. Kingma, A. Daffertshofer, D. F. Stegeman, and J. H. Van Dieën, "Heterogeneity of muscle activation in relation to force direction: A multi-channel surface electromyography study on the triceps surae muscle," *J. Electromyogr. Kinesiol.*, 2009.

[206] R. M. Enoka and A. J. Fuglevand, *Motor unit physiology: Some unresolved issues*, 2001.

[207] A. Ridderikhoff, C. L. E. Peper, R. G. Carson, and P. J. Beek, "Effector dynamics of rhythmic wrist activity and its implications for (modeling) bimanual coordination," *Hum. Mov. Sci.*, 2004.

[208] D. Staudenmann, I. Kingma, A. Daffertshofer, D. F. Stegeman, and J. H. Van Dieën, "Improving EMG-based muscle force estimation by using a high-density EMG grid and principal component analysis," *IEEE Trans. Biomed. Eng.*, 2006.

[209] W. Meng, B. Ding, Z. Zhou, Q. Liu, and Q. Ai, "An EMG-based Force Prediction and Control Approach for Robot-assisted Lower Limb Rehabilitation," *2014 IEEE Int. Conf. Syst. Man, Cybern.*, pp. 2198–2203, 2014.

[210] C. Loconsole, S. Dettori, A. Frisoli, C. A. Avizzano, and M. Bergamasco, "An EMG-based approach for on-line predicted torque control in robotic-assisted rehabilitation," *IEEE Haptics Symp. HAPTICS*, pp. 181–186, 2014.

[211] R. A. Gopura, K. Kiguchi, and Y. Yi, "SUEFUL-7: A 7DOF upper-limb exoskeleton robot with muscle-model-oriented EMG-based control," *2009 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS 2009*, pp. 1126–1131, 2009.

[212] K. T. Ebersole, T. J. Housh, G. O. Johnson, T. K. Evetovich, D. B. Smith, and S. R. Perry, "MMG and EMG responses of the superficial quadriceps femoris muscles," *J. Electromyogr. Kinesiol.*, 1999.

[213] R. L. Drake, A. W. Vogl, and A. W. Mitchell, "Gray's Anatomy for Students, Third Edition," *Gray's Anat. Students*, 2015.

[214] M. J. Stokes and P. A. Dalton, "Acoustic myography for investigating human skeletal muscle fatigue," in *J. Appl. Physiol.*, 1991.

[215] M. B. I. Reaz, M. S. Hussain, and F. Mohd-yasin, "Techniques of EMG signal analysis: detection, processing, classification and applications," *Biol. Proced. Online*, vol. 8, no. 1, pp. 11–35, 2006.

[216] A. Rainoldi, G. Melchiorri, and I. Caruso, "A method for positioning electrodes during surface EMG recordings in lower limb muscles," *J. Neurosci. Methods*, 2004.

[217] B. Gerdle, S. Karlsson, S. Day, and M. Djupsjöbacka, "Acquisition, Processing and Analysis of the Surface Electromyogram," in *Mod. Tech. Neurosci. Res.* 1999.

[218] C. M. Smith, T. J. Housh, E. C. Hill, G. O. Johnson, and R. J. Schmidt, "Dynamic versus isometric electromechanical delay in non-fatigued and fatigued muscle: A combined electromyographic, mechanomyographic, and force approach," *J. Electromyogr. Kinesiol.*, vol. 33, pp. 34–38, 2017.

[219] W. Van Drongelen, *Signal processing for neuroscientists*. 2018.

[220] A. K. Blangsted, G. Sjøgaard, P. Madeleine, H. B. Olsen, and K. Søgaard, "Voluntary low-force contraction elicits prolonged low-frequency fatigue and changes in surface electromyography and mechanomyography," *J. Electromyogr. Kinesiol.*, 2005.

[221] M. A. Alouane, H. Rifai, Y. Amirat, and S. Mohammed, "Cooperative Control for Knee Joint Flexion-Extension Movement Restoration," in *2018 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, IEEE, 2018, pp. 5175–5180.

[222] F. Guenzkofer, F. Engstler, H. Bubb, and K. Bengler, "Joint torque modeling of knee extension and flexion," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6777 LNCS, pp. 79–88, 2011.

[223] T. Seel, J. Raisch, and T. Schauer, "IMU-based joint angle measurement for gait analysis," *Sensors (Basel).*, vol. 14, no. 4, pp. 6891–6909, 2014.

[224] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *J. Mach. Learn. Res.*, 2011.

[225] S. F. Atashzar, I. G. Polushin, and R. V. Patel, "A Small-Gain Approach for Nonpassive Bilateral Telerobotic Rehabilitation: Stability Analysis and Controller Synthesis," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 49–66, 2017.

[226] D. Xu, X. Liu, and Q. Wang, "Knee Exoskeleton Assistive Torque Control Based on Real-Time Gait Event Detection," *IEEE Trans. Med. Robot. Bionics*, vol. 1, no. 3, pp. 158–168, 2019.

[227] G. Wu, S. Siegler, P. Allard, *et al.*, *ISB recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motion - Part I: Ankle, hip, and spine*, 2002.

[228] W. Huo, S. Mohammed, Y. Amirat, and K. Kong, "Fast Gait Mode Detection and Assistive Torque Control of an Exoskeletal Robotic Orthosis for Walking Assistance," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1035–1052, 2018.

[229] H. J. Luinge, P. H. Veltink, and C. T. Baten, "Ambulatory measurement of arm orientation," *J. Biomech.*, 2007.

Appendix A

Exoskeletons

A.1 Actuator Specifications

The detailed specifications for the HEBI X-Series X8-9 Series Elastic Actuator (SEA) are summarised in Table A.1. The torque-speed performance curves are shown in Figure A.1.

Table A.1: Actuator Module Specifications. HEBI X-Series Actuator technical specifications.

Configuration	X8-9
Peak Torque	20 Nm
Continuous Torque	8 Nm
Maximum Speed	30 rpm
Mass	475 g
Dimensions	110 mm x 73 mm x 45 mm 15mm hollow bore
Power	24 - 48 V
Temperature	-10 °C to 50 °C ambient
Communication	100 Mbps Ethernet (dual port)
Angular Resolution	0.005 °
Torque Resolution	0.01 Nm
Backlash	$\pm$ 0.25 °
Sensing	Angular Position Angular Velocity Output Torque 3-axis Accelerometer / Gyro Temperature Voltage Current
API Support	Matlab ROS Python C/C++

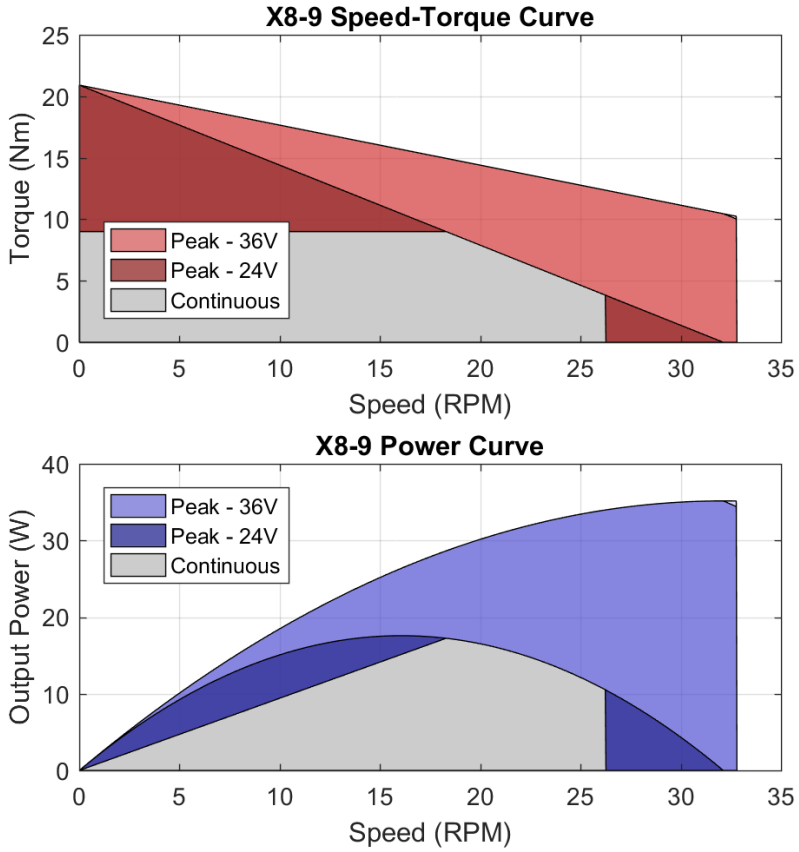
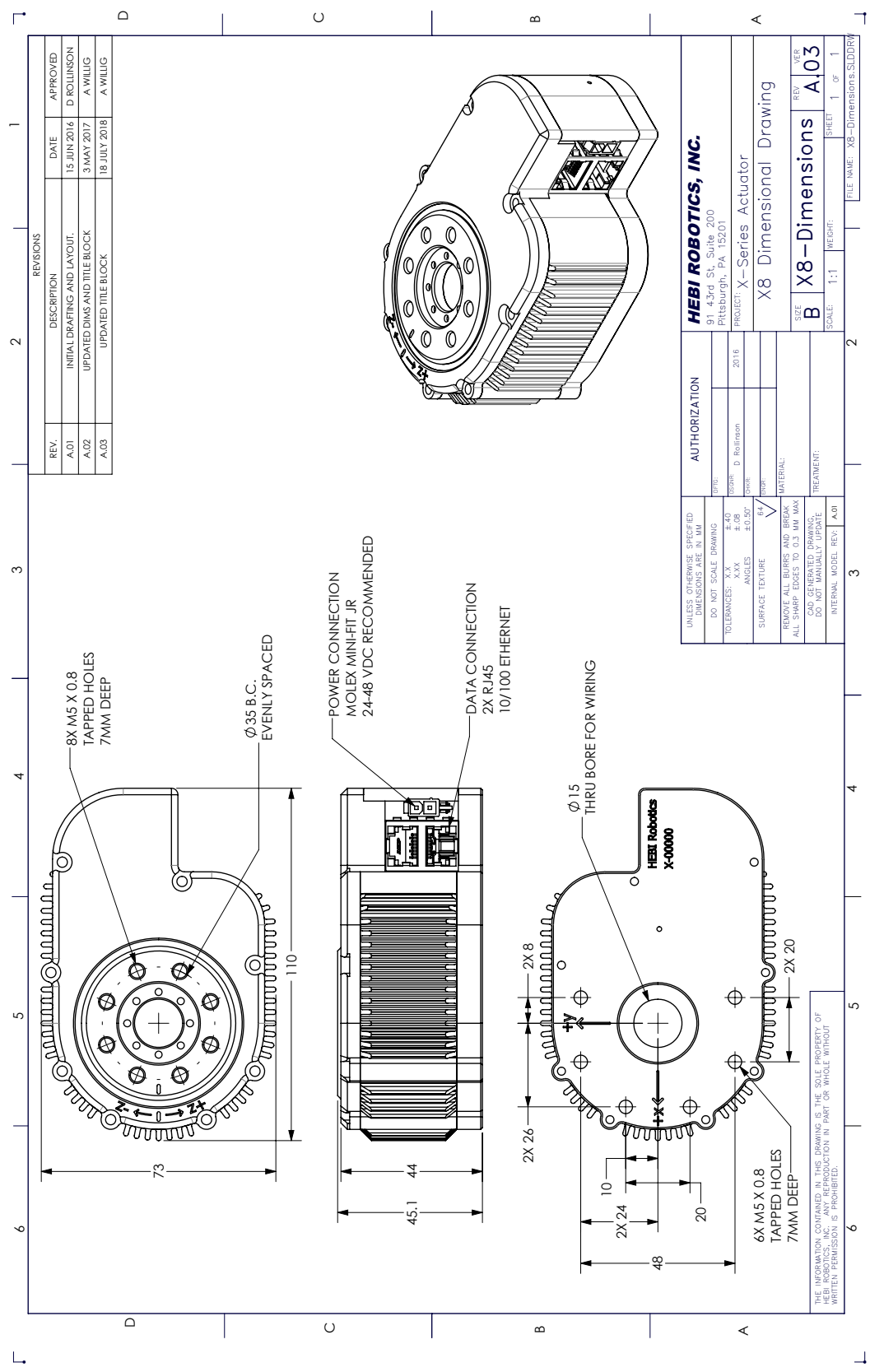
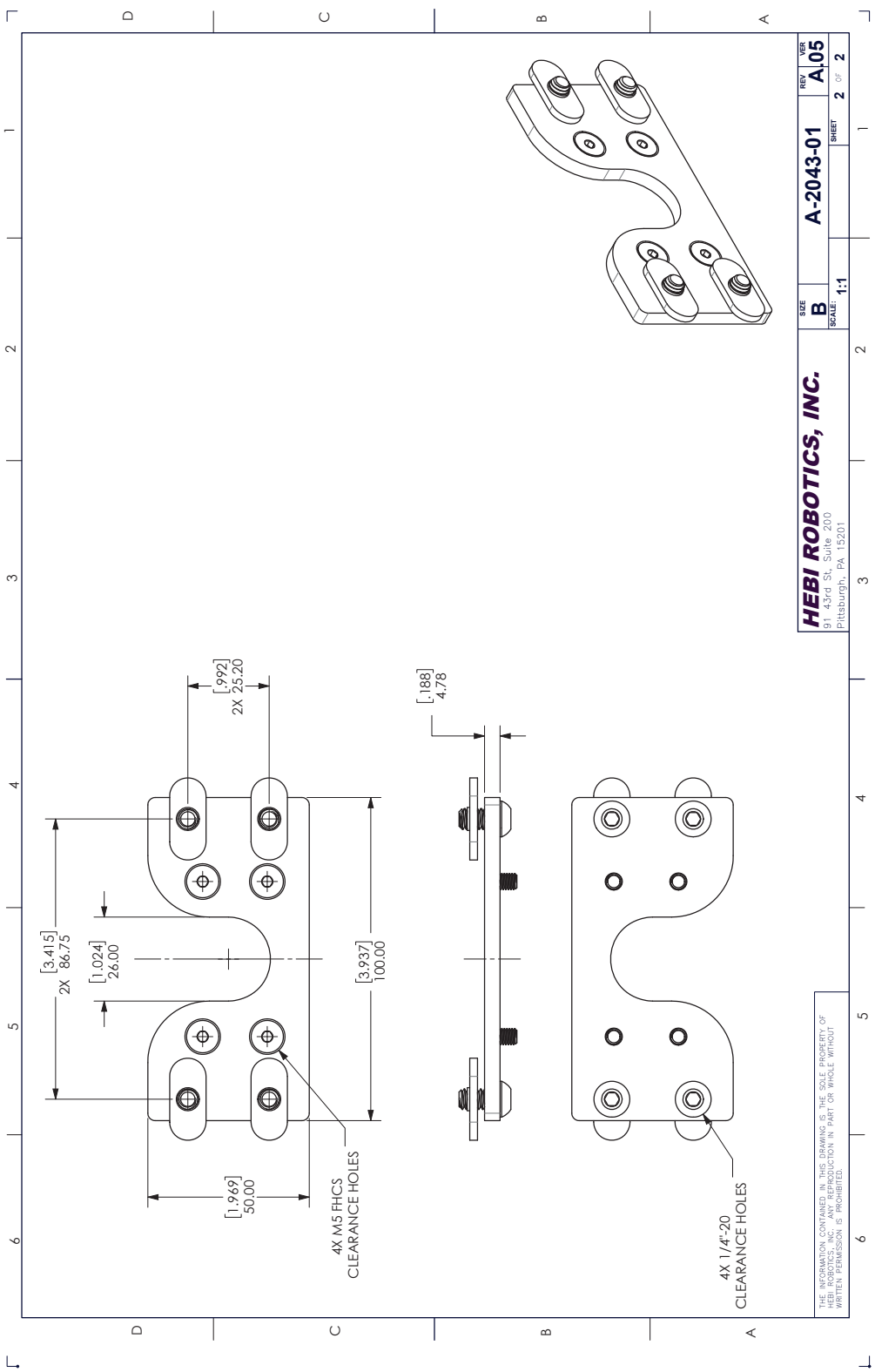
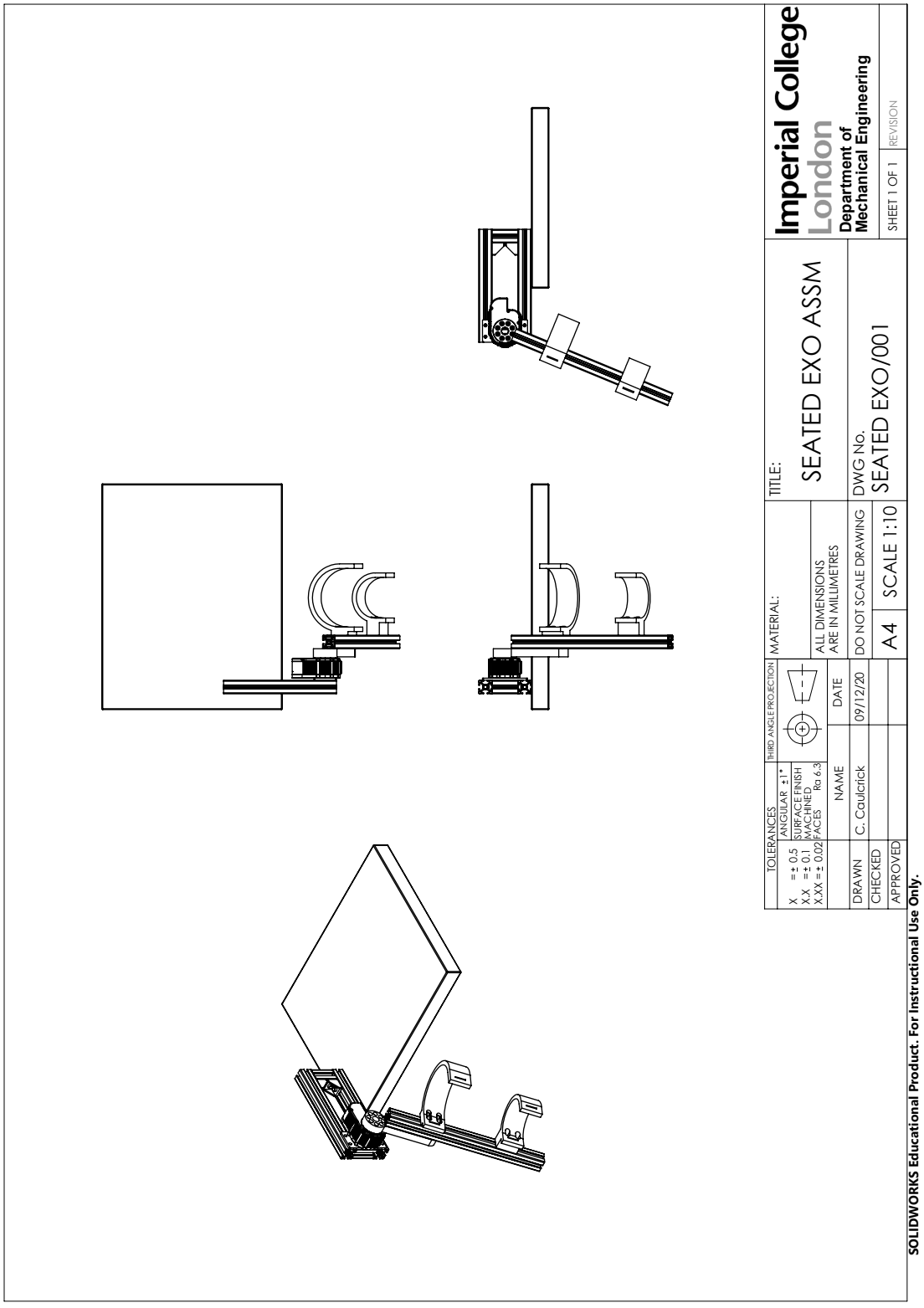


Figure A.1: Actuator Performance Curves. Performance curves for the HEBI X-Series X8-9 Actuator. The operating voltage for the experiments in this thesis is 30 V.

A.2 Technical Drawings


REVISIONS
REV. DESCRIPTION DATE APPROVED
A.01 INITIAL DRAFTING AND LAYOUT. 15 JUN 2016 D ROLLINSON
A.02 UPDATED DIMS AND TITLE BLOCK 3 MAY 2017 A WILLIG
A.03 UPDATED TITLE BLOCK 18 JULY 2018 A WILLIG
8X M5 X 0.8 TAPPED HOLES 7MM DEEP
Ø35 B.C. EVENLY SPACED
110
73
POWER CONNECTION MOLEX MINI-FIT JR 24-48 VDC RECOMMENDED
DATA CONNECTION 2X RJ45 10/100 ETHERNET
44
45.1
Ø15 THRU BORE FOR WIRING
HEBI Robotics X-00000
2X 8
2X 26
2X 20
10
2X 24
20
48
6X M5 X 0.8 TAPPED HOLES 7MM DEEP
+y
+x
+z
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN MM
DO NOT SCALE DRAWING
TOLERANCES: X.X ±.40 X.XX ±.08 ANGLES ±0.50°
SURFACE TEXTURE 64
REMOVE ALL BURRS AND BREAK ALL SHARP EDGES TO 0.3 MM MAX
CAD GENERATED DRAWING. DO NOT MANUALLY UPDATE
INTERNAL MODEL REV.: A.01
AUTHORIZATION
DFTR:
ISSUER: D Rollinson 2016
CHKR:
ENGR:
MATERIAL:
TREATMENT:
HEBI ROBOTICS, INC.
91 43rd St, Suite 200
Pittsburgh, PA 15201
PROJECT: X-Series Actuator
X8 Dimensional Drawing
SIZE: B
X8-Dimensions
REV: A.03
VER
SCALE: 1:1
WEIGHT:
SHEET 1 OF 1
FILE NAME: X8-Dimensions.SLDDRW
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF HEBI ROBOTICS, INC. ANY REPRODUCTION IN PART OR WHOLE WITHOUT WRITTEN PERMISSION IS PROHIBITED.


[.992]
2X 25.20
[.188]
4.78
[3.415]
2X 86.75
[1.024]
26.00
[3.937]
100.00
[1.969]
50.00
4X M5 FHCS
CLEARANCE HOLES
4X 1/4"-20
CLEARANCE HOLES
HEBI ROBOTICS, INC.
91 43rd St. Suite 200
Pittsburgh, PA 15201
SIZE B
A-2043-01
REV A.05
SCALE: 1:1
SHEET 2 OF 2
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF HEBI ROBOTICS, INC. ANY REPRODUCTION IN PART OR WHOLE WITHOUT WRITTEN PERMISSION IS PROHIBITED.


TOLERANCES
ANGULAR ±1°
X = ± 0.5
X.X = ± 0.1
X.XX = ± 0.02
SURFACE FINISH
MACHINED Ra 6.3
FACES
THIRD ANGLE PROJECTION
MATERIAL:
ALL DIMENSIONS ARE IN MILLIMETRES
NAME
DATE
DRAWN
C. Caulcrick
09/12/20
CHECKED
APPROVED
DO NOT SCALE DRAWING
A4
SCALE 1:10
TITLE:
SEATED EXO ASSM
DWG No.
SEATED EXO/001
Imperial College London
Department of Mechanical Engineering
SHEET 1 OF 1
REVISION
SOLIDWORKS Educational Product. For Instructional Use Only.

A.3 IMU Calibration Procedure

First, the wearer is instructed to stay in a standing position, where gravity vectors $\boldsymbol{g}_{\text{stand}}$ measured by the IMUs are considered to be aligned with the vertical axes ${}^{B}\boldsymbol{Y}_{j}$ of each segment. Next, the wearer is instructed to perform a swinging leg flexion/extension movement using the hip, knee and ankle joints in the sagittal plane. The axis ${}^{B}\boldsymbol{Z}_{j}$ of the foot is defined using the rotation axis of the angular velocity $\boldsymbol{\omega}_{joint}$ measured by the IMUs. The three unit vectors of orientation ${}^{S}\boldsymbol{R}_{B,j}$ can be expressed as follows [229]:

$$\begin{cases} {}^{S}\boldsymbol{Y}_{B,j} = \frac{\boldsymbol{g}_{\text{stand},j}}{|\boldsymbol{g}_{\text{stand},j}|} \\ {}^{S}\boldsymbol{Z}_{B,j} = d\frac{\boldsymbol{\omega}_{\text{joint},j}}{|\boldsymbol{\omega}_{\text{joint},j}|} \\ {}^{S}\boldsymbol{X}_{B,j} = \frac{{}^{S}\boldsymbol{Z}_{B,j}\times{}^{S}\boldsymbol{Y}_{B,j}}{|{}^{S}\boldsymbol{Z}_{B,j}\times{}^{S}\boldsymbol{Y}_{B,j}|} \end{cases} \tag{A.1}$$

where $\boldsymbol{g}_{\text{stand},j}$ denotes the measured gravity vector of each IMU in the standing posture, d the rotation direction of each joint ($d = 1$ during flexion movements, and $d = -1$ during extension movements), and $\omega_{\text{joint},j}$, the measured angular velocity.

Consider that the measured vectors ${}^{S}\boldsymbol{Y}_{B,j}$ and ${}^{S}\boldsymbol{Z}_{B,j}$ in A.1 may not be orthogonal due to measurement error. The orientation ${}^{S}\boldsymbol{R}_{B,j}$ is modified as follows [228]:

$${}^{S}\boldsymbol{R}_{B,j} = [{}^{S}\boldsymbol{X}_{B,j}{}^{S}\boldsymbol{Y}_{B,j}{}^{S}\boldsymbol{X}_{B,j} \times {}^{S}\boldsymbol{Y}_{B,j}]. \tag{A.2}$$

The measured ${}^{S}\boldsymbol{Z}_{B,j}$ axis is replaced by the term ${}^{S}\boldsymbol{X}_{B,j} \times {}^{S}\boldsymbol{Y}_{B,j}$. This is due to the fact that it its not possible to guarantee perfectly aligned flexion/extension movements in the sagittal plane.

Appendix B

Human Torque Estimation

B.1 Human Torque Estimation Datasets

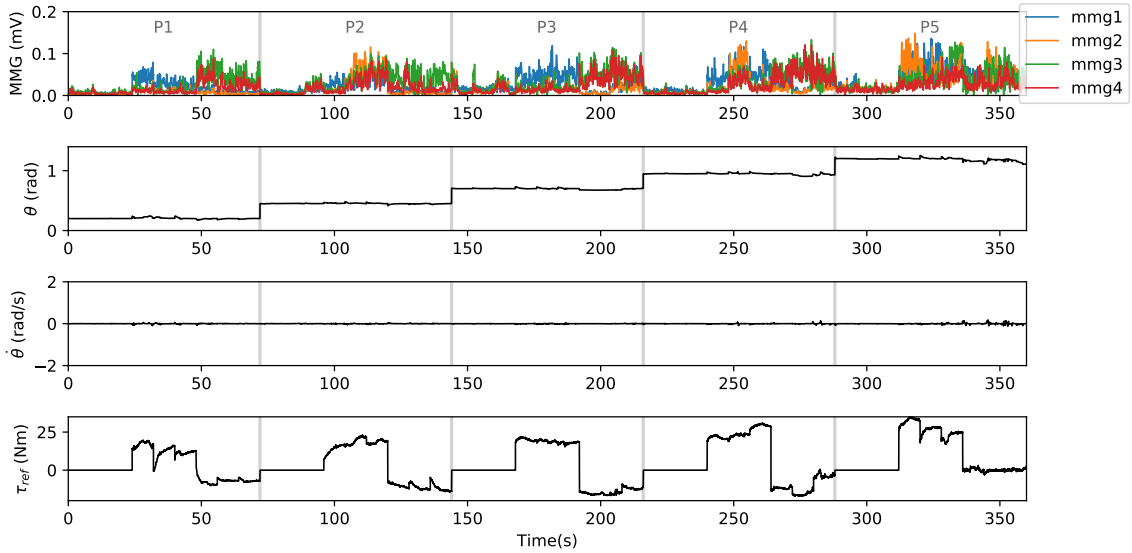


Figure B.1: HTE Dataset – Test 1. Human torque estimation dataset for test 1 with processed MMG muscle activity, system states of knee joint angle, θ, and angular velocity, $\dot{\theta}$, and reference human joint torque, τ_{ref}, recorded during exercise 1: isometric contraction.

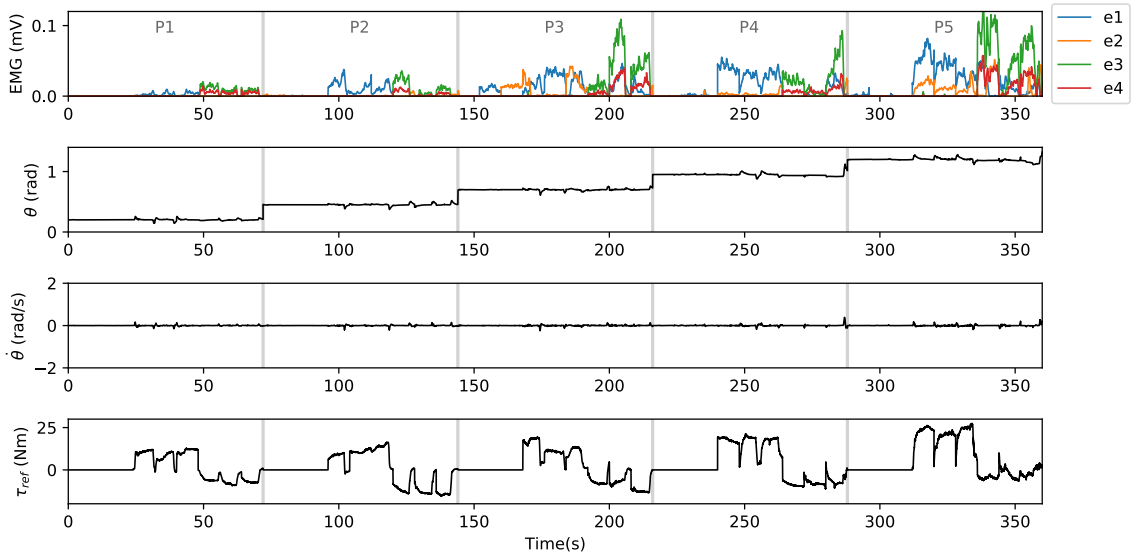


Figure B.2: HTE Dataset – Test 2. Human torque estimation dataset for test 2 with processed EMG muscle activity, system states of knee joint angle, θ, and angular velocity, $\dot{\theta}$, and reference human joint torque, τ_{ref}, recorded during exercise 1: isometric contraction.

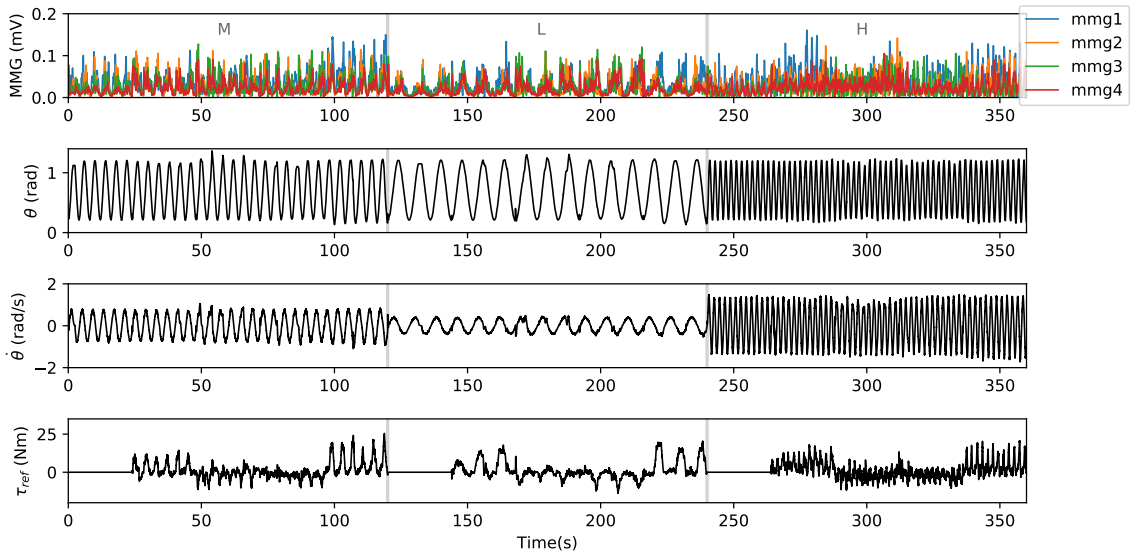


Figure B.3: HTE Dataset – Test 3. Human torque estimation dataset for test 3 with processed MMG muscle activity, system states of knee joint angle, θ, and angular velocity, $\dot{\theta}$, and reference human joint torque, τ_{ref}, recorded during exercise 2: dynamic extension and flexion.

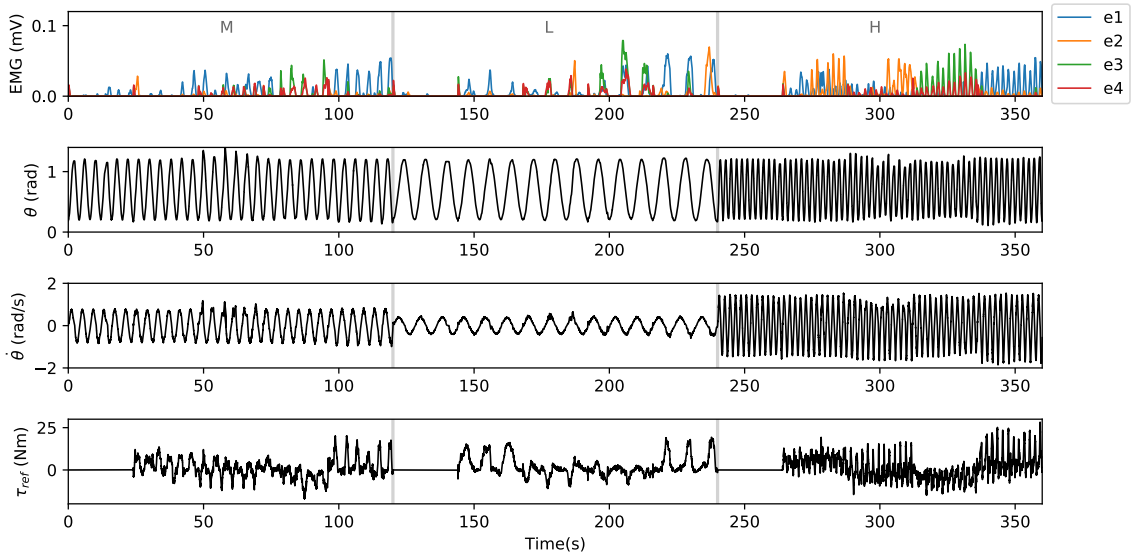


Figure B.4: HTE Dataset – Test 4. Human torque estimation dataset for test 4 with processed EMG muscle activity, system states of knee joint angle, θ, and angular velocity, $\dot{\theta}$, and reference human joint torque, τ_{ref}, recorded during exercise 2: dynamic extension and flexion.

B.2 Linear and Polynomial Model Fitting Code

```
import numpy as np
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, mean_absolute_error
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.pipeline import Pipeline


def linear_fit(X,y,dtheta_r):
    model = linear_model.LinearRegression()
    model.fit(X,y)
    y_pred = model.predict(X)
    sqr_err = mean_squared_error(y,y_pred)
    abs_err = mean_absolute_error(y,y_pred)
    abs_err_ar = np.abs(y-y_pred)
    std_err = np.std(abs_err_ar)
    fla = fla_predict(dtheta_r,y,y_pred)
    return model, y_pred, sqr_err, abs_err, abs_err_ar, std_err, model.coef_,
       model.intercept_, fla


def poly_fit(X,y,dtheta_r):
    model = Pipeline([('poly', PolynomialFeatures(degree=2)),
                      ('linear', LinearRegression())])
    model = model.fit(X,y)
    coef = model.named_steps['linear'].coef_
    intercept = model.named_steps['linear'].intercept_
    y_pred = model.predict(X)
    sqr_err = mean_squared_error(y,y_pred)
    abs_err = mean_absolute_error(y,y_pred)
    abs_err_ar = np.abs(y-y_pred)
    std_err = np.std(abs_err_ar)
    fla = fla_predict(dtheta_r,y,y_pred)
    return model, y_pred, sqr_err, abs_err, abs_err_ar, std_err, coef, intercept
       , fla
```

B.3 Neural Network Model Training Code

```
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers


def build_model(n_neurons):
    model = keras.Sequential([
                              layers.Dense(n_neurons, activation='relu',
                                 input_shape=[len(train_dataset.keys())]),
                              layers.Dense(n_neurons, activation='relu'),
                              layers.Dense(1)])
    optimizer = tf.keras.optimizers.RMSprop(0.001)
    model.compile(loss='mse',
                  optimizer=optimizer,
                  metrics=['mae','mse'])
    return model


# Create a callback that saves the model's weights
cp_callback = tf.keras.callbacks.ModelCheckpoint(filepath=checkpoint_path,
                                                 save_weights_only=True,
                                                 verbose=1,
                                                 save_freq='epoch')

# The patience parameter is the amount of epochs to check for improvement
early_stop = keras.callbacks.EarlyStopping(monitor='val_loss', patience=10)

EPOCHS = 1000 # Max number of epochs

history = model.fit(
    normed_train_data, train_labels,
    epochs=EPOCHS, validation_split = 0.2, verbose=1,
    callbacks=[tfdocs.modeling.EpochDots(),cp_callback,early_stop])
```

B.4 Mean Absolute Error Comparison of MMG and EMG

Table B.1: Mean Absolute Error Values. MAE for human joint torque estimation from MMG and EMG muscle activity signals using the linear (L) and polynomial (P) models across input sets A to F various isometric and dynamic exercise sets.

	Inputs	P1	P2	P3	P4	P5	M	L	H	All	Iso	Dyn
MMG - Linear	A	2.93	5.15	5.75	6.93	7.18	3.05	3.03	3.21	6.0	6.21	3.13
	B	2.9	4.67	5.21	6.91	5.99	3.07	3.05	3.23	6.0	6.14	3.16
	C	2.89	4.64	5.12	6.87	5.88	3.06	3.04	3.23	6.0	6.11	3.16
	D	2.7	4.03	5.25	6.57	6.96	3.02	3.02	3.15	5.71	5.65	3.13
	E	2.67	3.77	4.98	6.58	5.56	2.94	3.05	3.17	5.64	5.5	3.13
	F	2.67	3.72	4.89	6.56	5.43	2.94	3.04	3.15	5.65	5.44	3.13
MMG - Poly.	A	2.58	4.54	4.93	6.36	7.23	2.79	3.02	3.21	5.86	5.93	3.08
	B	2.53	4.24	4.22	6.33	5.08	2.78	2.97	3.19	5.83	5.78	3.07
	C	2.52	4.22	4.14	6.18	4.99	2.72	2.94	3.1	5.69	5.71	3.06
	D	2.27	3.21	4.09	5.09	6.64	2.66	2.91	3.08	5.53	5.19	3.03
	E	2.23	3.0	3.79	4.87	4.61	2.66	2.79	3.01	5.41	4.84	2.99
	F	2.17	2.94	3.75	4.76	4.5	2.56	2.76	2.94	5.24	4.79	2.97
EMG - Linear	A	2.84	4.14	3.53	2.92	3.9	2.79	2.53	3.3	3.46	3.98	2.88
	B	2.64	4.17	3.48	2.7	3.91	2.8	2.35	3.34	3.55	3.99	2.85
	C	2.6	4.14	3.45	2.65	3.92	2.81	2.38	3.33	3.54	3.99	2.85
	D	2.75	4.13	3.18	2.62	3.27	2.47	2.3	3.08	3.27	3.88	2.65
	E	2.6	4.17	3.14	2.38	3.26	2.47	2.16	3.16	3.33	3.89	2.64
	F	2.55	4.14	3.13	2.3	3.25	2.47	2.21	3.16	3.33	3.86	2.64
EMG - Poly.	A	1.72	3.04	3.1	2.18	2.4	2.69	2.5	3.25	3.16	3.26	2.86
	B	1.49	2.64	2.98	1.61	2.32	2.6	2.3	3.17	3.08	3.31	2.8
	C	1.37	2.58	2.86	1.46	2.3	2.58	2.21	3.09	3.07	3.3	2.77
	D	1.56	2.94	2.67	1.6	2.12	2.35	2.21	2.95	3.01	3.19	2.59
	E	1.35	2.58	2.63	1.22	2.05	2.34	2.11	2.91	2.97	3.19	2.56
	F	1.25	2.47	2.57	1.13	1.88	2.27	1.98	2.82	3.0	3.14	2.55

B.5 MMG and EMG Signal Filters

B.5.1 Signal Filter Code

```
#include "daq.h"

double DAQ::lowpass(double X_in, int signal)
{
        low_para.x[0][signal] = X_in;

        low_para.y[0][signal] = k*(low_para.x[0][signal] + 2 * low_para.x[1][
           signal] + low_para.x[2][signal]) - a1*low_para.y[1][signal] - a2*
           low_para.y[2][signal];

        for (int i = 2; i>0; i--)
        {
                low_para.x[i][signal] = low_para.x[i - 1][signal];
                low_para.y[i][signal] = low_para.y[i - 1][signal];
        }
        return low_para.y[0][signal];
}

double DAQ::highpass(double X_in, int signal)
{
        high_para.x[0][signal] = X_in;

        high_para.y[0][signal] = k * (high_para.x[0][signal] - 2 * high_para.x
           [1][signal] + high_para.x[2][signal]) - a1 * high_para.y[1][signal] -
            a2 * high_para.y[2][signal];

        for (int i = 2; i > 0; i--)
        {
                high_para.x[i][signal] = high_para.x[i - 1][signal];
                high_para.y[i][signal] = high_para.y[i - 1][signal];
        }
        return high_para.y[0][signal];
}

double DAQ::clipLimSignal(double signal, double lim) {
        if (abs(signal) > lim) {
                signal = 0;
        }
        return signal;
}

double DAQ::noiseLimSignal(double signal, double lim) {
        if (abs(signal) < lim) {
                signal = 0;
        }
        return signal;
}
```

B.5.2 Filter Parameters

Table B.2: Muscle Activity Filter Information. Table showing muscle activity filter information and parameters for the MMG and EMG signals.

Filter	Signal	n	f_s (Hz)	b_1	b_2	b_3	a_1	a_2	a_3
HP1	MMG	2	1000	0.9565	-1.9131	0.9565	1.0000	-1.9112	0.9150
LP1				0.0675	0.1349	0.0675	1.0000	-1.1430	0.4128
LP2				3.9130e-05	7.8260e-05	3.9130e-05	1.0000	-1.9822	0.9824
HP1	EMG	2	2048	0.9575	-1.9151	0.9575	1.0000	-1.9133	0.9169
LP1				0.2822	0.5644	0.2822	1.0000	-0.0431	0.1720
LP2				2.3480e-06	4.6960e-06	2.3480e-06	1.0000	-1.9957	0.9957

Appendix C

Model Predictive Control

C.1 Controller Code

The branches of the Github repository at https://github.com/cic12/exo_control contain the code used for the controller experiments in this work.