

sEMG-Based Lower Limb Intention Detection using Artificial Intelligence and its Impact on Assistive Human-Robot Interaction

by

Hasti Khiabani

A Thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfilment of
the requirements for the degree of
Master of Applied Science

Ottawa-Carleton Institute for
Mechanical and Aerospace Engineering

Department of Mechanical and Aerospace Engineering
Carleton University
Ottawa, Ontario, Canada
August 2021

Copyright ©

2021 - Hasti Khiabani

Abstract

People with mobility-related disabilities, including individuals with Cerebral Palsy (CP), can observe improvements in their quality of life by incorporating various types of robotic devices, e.g., rehabilitation, assistive, and Human-Computer Interaction (HCI) devices. To be compliant to users' needs, such robots should be intelligent in regards to their human partners' intention to be able to adapt themselves to their needs and provide beneficial feedback in executing a desired motion. For this purpose, a Surface Electromyography (sEMG)-based intelligent lower limb intention-detection system is studied to augment Human-Robot Interaction (HRI) by detecting subjects' walking direction prior-to or during walking. Ten Classical Machine Learning (C-ML)-based models with two different implementation strategies (Subject-Exclusive and Generalized) are employed to detect direction intentions and evaluate inter-subject robustness in one knee/foot-gesture and three walking-related scenarios. In each, sEMG signals are collected from eight muscles of nine subjects during five trials of at least nine distinct gestures/activities. Four combinations of features are extracted using the feature engineering method to be evaluated as C-ML inputs. Linear Discriminant Analysis (LDA) and Random Forest (RF) classifiers, applied to the Time-Domain (TD) feature set, provided the best accuracy. Convolutional Neural Network (CNN) approach (from the family of Deep Learning (DL) methods) is further investigated to automatically learn discriminant features. To

enhance the detection accuracy and reliability of the detecting system further, an advanced transfer learning methodology (Subject-Adaptive) is employed which leverages the learning over previous participants and adapts the network parameters according to the new participant's particular manner of doing a task. DL approach shows better results than the C-ML approach, particularly when using the Subject Exclusive strategy. Finally, the advanced DL model is employed in a computer-simulated environment with the direct Inertial Measurement Unit (IMU) and sEMG data collected from the human. The proper method for augmenting an HRI controller is evaluated, and a proof of concept for the feasibility of this ideology is presented. A combination of Impedance and Velocity Adapted (IA+VA) augmentation methodology is identified as the most effective method to improve the control of a gait training robotic system, particularly when assisting people experiencing disability.

Acknowledgments

I'd like to thank my supervisor, Dr. Mojtaba Ahmadi, for giving me the opportunity to work on this project, and for all the assistance navigating grad school. I would like to thank my co-supervisor Dr. Anna McCormick and her team for providing helpful consultation along my path.

I would also like to thank all the members of Advanced Biomechatronics Locomotion Lab (ABL) for their support and advice. Some special words of gratitude go to my friends who have always been a major source of support when things would get a bit discouraging, especially Borna, Aida, and many others. Finally, my deep and sincere gratitude to my family for their continuous and unparalleled love, help and support.

Table of Contents

Abstract	ii
Acknowledgments	iv
Table of Contents	v
List of Tables	ix
List of Figures	x
Nomenclature	xiii
1 Introduction	1
1.1 Project Background	5
1.2 Thesis Objectives	7
1.3 Thesis Contributions	9
1.4 Thesis Outline	10
2 Background	12
2.1 Introduction	12
2.2 Human-Robot Communication Means	14
2.3 Intention Detection	15
2.3.1 Feature Engineering and Classical Machine Learning approach	16

2.3.2	Feature Learning and Deep Learning approach	20
2.4	Human-Robot Interaction Control	23
2.5	Rehabilitation and Assistive Devices Application	25
3	Experiments	30
3.1	Introduction	31
3.1.1	Scenarios	31
3.2	Experiment	34
3.2.1	Processing Steps	34
3.2.2	Graphical User Interface	35
3.2.3	Data Acquisition	36
3.3	Signal Pre-processing	39
3.3.1	Data Segmentation	40
3.3.2	Filtering	41
4	Lower Limb Intention Detection Using Classical Machine Learning	42
4.1	Introduction	43
4.2	Feature Engineering	43
4.2.1	Feature Selection and Dimensionality Reduction	45
4.2.2	Feature Sets	46
4.3	Classic Machine Learning (C-ML) Fundamentals	50
4.3.1	Primary Models	50
4.3.2	Ensemble Models	54
4.4	Hyperparameter Tuning	56
4.5	Different Implementation Algorithms	58
4.6	Intention Detection Evaluation Metrics	59
4.7	Results	60
4.8	Discussion	62

4.8.1	Confusion Matrix	68
4.9	Chapter Summary	74
5	Lower Limb Intention Detection Using Deep Learning	75
5.1	Introduction	75
5.2	Deep Learning Fundamentals	77
5.2.1	Deep Neural Network Basic Structure	77
5.2.2	Convolutional Neural Network (CNN) Basic Structure	79
5.2.3	Network Optimization	81
5.2.4	Regularization	82
5.3	Proposed CNN Architecture	84
5.3.1	Transfer Learning	85
5.4	Different Implementation Algorithms	88
5.5	Results	89
5.6	Discussion	90
5.7	Comparison of C-ML and Deep Learning	94
5.8	Identifying the Optimal Configuration of Sensors	95
5.9	Chapter Summary	98
6	Application to the Control of Assistive Robotic Systems	99
6.1	Introduction	100
6.2	Admittance Control vs Impedance Control	103
6.3	Simulation	108
6.3.1	Model	109
6.4	Methods for Augmenting Admittance Control	113
6.5	Tests	114
6.6	Results	118
6.7	Discussion	125

6.7.1	Observations over different Simulation Modes (SM1-4)	125
6.7.2	Comparison of different augmentation methodologies (SM4-5)	127
6.8	Chapter Summary	131
7	Conclusion and Future Work	132
7.1	Conclusion	132
7.2	Recommendations and Future Work	135
	List of References	138
	Appendix A Experiments	151
A.1	Gestures Performed during Different Scenarios	151
A.2	Ethics Clearance	156
	Appendix B Custom-made 3d-printed parts	159
	Appendix C IMU Calibration and Fusion Methodology	166
C.1	IMU Calibration	166
C.2	IMU Sensor Fusion	167
	Appendix D Simulation Environment	169

List of Tables

4.1	List of features extracted for C-ML	47
4.2	C-ML models' hyperparameters tuning-search range	57
4.3	C-ML models' accuracy score in Sitting Position scenario	63
4.4	C-ML Models' accuracy score in Standing Position scenario	63
4.5	C-ML Models' accuracy score in Attempted Motion scenario	64
4.6	C-ML models' accuracy score in Dynamic Motion scenario	64
5.1	Parameters of the proposed CNN architecture	85
5.2	DL models' accuracy in Sitting Position scenario	89
5.3	DL models' accuracy in Standing Position scenario	90
5.4	DL models' accuracy in Attempted Motion scenario	90
5.5	DL models' accuracy in Dynamic Motion scenario	90
6.1	Designed and tuned parameters of simulation	112
6.2	Simulation results of each subject in each Simulation Mode	120
A.1	List of Gestures and Activities performed in each Scenario	151

List of Figures

1.1	Project Background: SoloWalk Robot	6
2.1	Samples of intention detection used in biomedical applications	29
3.1	Sample of gestures/activities performed	33
3.2	Machine Learning Process-General Diagram	35
3.3	Sample of designed GUI Window	35
3.4	Data Acquisition setup	36
3.5	Sample of wearable setup used by a participant	39
4.1	C-ML models' performance - different feature sets - Sitting Position	60
4.2	C-ML models' performance - different feature sets - Standing Position	61
4.3	C-ML models' performance - different feature sets - Attempted Motion	61
4.4	C-ML models' performance - different feature sets - Dynamic Motion	62
4.5	Best Model's Confusion Matrix in Sitting Position scenario	69
4.6	Best Model's Confusion Matrix in Standing Position scenario	70
4.7	Best Model's Confusion Matrix in Attempted Motion scenario	72
4.8	Best Model's Confusion Matrix in Dynamic Motion scenario	73
5.1	Deep Neural Network Basic Structure	78
5.2	Convolutions in CNNs	80
5.3	Samples of modern Activation Functions	81
5.4	Designed CNN architecture for eight channel sEMG input	84
5.5	Advanced TL architecture.	87

5.6	Designed CNN architecture for four channel sEMG input	96
5.7	Performance comparison of different sensor configurations	97
6.1	Block diagram of generic impedance control.	105
6.2	Block diagram of a generic admittance control.	107
6.3	Block diagram of the developed simulation model	109
6.4	Final tests	115
6.5	Subject1- test1- trajectory following in each SM	121
6.6	Subject1- test2 - trajectory following in each SM	122
6.7	Subject2- test1 -trajectory following in each SM	123
6.8	Subject2 -test2- trajectory following in each SM	124
A.1	Static gestures to be performed in Sitting Position Scenario	152
A.2	Static gestures to be performed in Standing Position Scenario	153
A.3	Self-prevented gestures to be performed in Attempted Motion Scenario	154
A.4	Dynamic activities to be performed in Dynamic Motion Scenario	155
B.1	IMU base box	159
B.2	IMU calibration box	160
B.3	First gear connector	161
B.4	Second gear connector	162
B.5	Calibration box and gear connectors assembly	163
B.6	strap holder component	164
B.7	Motor holder components	165
D.1	Simulation block of admittance-controller	169
D.2	Simulation block of Human Model and Robot Model	170
D.3	Simulation block of Contact Model	171
D.4	Simulation block of Coordinate Transformation	172
D.5	Simulation block of PID controller on human model	173
D.6	sEMG-based command generation block	174

D.7 Simulation Block of Human and Robot Differential Kinematics	175
D.8 All simulation blocks together	176

Nomenclature

List of Abbreviations

Name	Acronym	Name	Acronym
Ensemble Adaptive Boosting	Ada	Integrated Mean Absolute Value	iMAV
Adaptive Batch Normalization	AdaBN	Inertial Measurement Unit	IMU
Adaptive Moment Estimation	Adam	K-Nearest Neighbors	KNN
Actuated Mobile Gait Trainers	AMGT	Linear Discriminant Analysis	LDA
Auto-Regression	AR	Mean Absolute Value	MAV
Ensemble Bagging	Bag	Multi-Model HRI	MHRI
Biceps Femoris	BF	Multi-Layer Perceptron	MLP
Batch Normalization	BN	Motor Units	MU
Body-Weight Supported	BWS	Motor Unit Action Potential	MUAP
BWS Treadmill training	BWSTT	Neural Network	NN
Classical Machine Learning	C-ML	Principle Component Analysis	PCA
Convolutional Neural Network	CNN	Progressive Neural Network	PNN
Convolutional Layer	CONV	Parametric Rectified Linear Unit	PReLU
Cerebral palsy	CP	Quadratic Discriminant Analysis	QDA
Continuous Wavelet Transform	CWT	Rectified Linear Unit	ReLU

Deep Learning	DL	Ensemble Random Forest	RF
Deep Neural Network	DNN	Root Mean Square	RMS
Degree of Freedom	DoF	Root Mean Square Error	RMSE
Decision Tree	DT	Recurrent Neural Network	RNN
Electroencephalogram	EEG	Surface Electromyogram	sEMG
Electromyogram	EMG	Skewness	SKEW
Force Adapted	FA	Simulation Mode	SM
Fully connected Layers	FC	Slope Sign Change	SSC
Frequency Domain	FD	Standard Deviation	STD
Functional Electrical Stimulation	FES	Support Vector Machine	SVM
Ensemble Gradient Boosted DT	GBDT	Tibialis Anterior	TA
Gastrocnemius Lateralis	GL	Time Domian	TD
Gastrocnemius Medialis	GM	Time-Frequency Domain	TFD
Gaussian Naive Bayes	GNB	Transfer Learning	TL
Graphical User Interface	GUI	Velocity Adapted	VA
Human Computer Interaction	HCI	Vastus Lateralis	VL
Histogram	HIST	Vastus Medialis	VM
Human Robot Interaction	HRI	Waveform Length	WL
Impedance Adapted	IA	Zero Crossing	ZC

Chapter 1

Introduction

As robotic devices get more involved in our lives, the commodification of robots creates new opportunities to integrate autonomy into our day-to-day life. This is particularly beneficial for people who experience some level of disability (i.e. impairments of motor functioning, vision, hearing, cognition and mental health) [1]. Based on the statistics published by the Statistics Canada, in 2017, one in five (22%) of the Canadian population aged 15 years and over- or about 6.2 million individuals- had one or more disabilities, 9.6% of which is related to mobility [2]. People with mobility-related disabilities can observe improvements in their quality of life through incorporation of various types of robotic assets, e.g. rehabilitation devices, assistive devices, and human computer interaction (HCI) devices [3–7]. These robots and devices aim to enhance the patients’ engagement and interaction with their environment and eventually improve their independence [1]. Rehabilitation robots offer high intensity exercise with the intense and precise repetition to enhance patient’s motor capabilities as much as possible and decrease caregiver burden [3, 5, 8]. Assistive devices compensate impairments by providing an alternative means to the patient to perform certain activities [6]. Accessibility-based HCI devices can utilize the patients capabilities to help them engage with the environment in their daily activities in a more sufficient way [7,9]. As an example, cerebral palsy (CP) is a neurological disorder

that emerges in infancy or early childhood. CP, caused by a non-progressive injury in a developing brain, is considered the most common cause of childhood physical disability with a global prevalence of approximately 1.5 to 4 per 1000 live births [10]. CP cases are classified into four distinct types based on the region in the brain that is damaged, which includes: (i) Spastic (ii) Dyskinetic (iii) Ataxic (iv) Mixed. Spastic is by far the most common type of cerebral palsy, accounting for over 75% of all cases. Spastic CP is primarily characterized by spastic stiffness and rigidity of the muscles, which impairs normal movements. Spastic CP cases are further classified into four subtypes based on the location of the impairment as: (i) Monoplegia, where one limb is affected, (ii) Hemiplegia, a form of unilateral CP where one side of the body is affected, (iii) Diplegia, a form of bilateral cerebral palsy where most of the times both legs are affected, (iv) Quadriplegia, a form of bilateral cerebral palsy where both arms and legs are affected [11]. CP impacts body movement and posture. This difficulty with movement and the resultant movement patterns may be highly variable moment to moment and across the life-span. Therefore, assistance is usually provided to individuals with CP through utilizing the mentioned robotic means for maintaining their muscular capabilities, facilitating the ability to exercise, enhancing functioning, and promoting independence; all with the goal of improving quality of life for the person and providers of care [1, 12].

Rehabilitation robots and assistive devices can target either the upper extremities or the lower extremities. The focus of this research is on the rehabilitation robots for lower extremities. One very important forms of lower-limb rehabilitation is gait training, where in its classical format, the patient walks with a therapist's assistance. Lower-limb rehabilitation robots were introduced to provide intense and focused training with a high repetition to invoke neuroplasticity, promote motor learning and positively influence functional motor trajectory of the users in a more progressive manner than traditional rehabilitation solely done by a therapist [3]. Body weight

support (BWS) and fall prevention mechanisms are incorporated into these devices to have control over training intensity (via controlling body-weight unloading), and ensure safety. These gait training robots, can be divided into three categories of (i) End-effector-type robots that attach to the user's feet and guide the motion of the user's feet, (ii) exoskeleton-type robots that attach parallel to the patient's lower extremities, and (iii) actuated mobile gait trainers (AMGT) that enable overground gait training. The structure of AMGTs allows users to practice over ground walking which can further facilitate balance training. Gait trainers have the potential to dramatically improve the walking capability of millions of individuals with gait impairments.

In any Human Robot Interaction (HRI), a key concern is ensuring the users' safety. A proper rehabilitation robots' control law precisely guides the patient through a smooth trajectory required for their therapy while considering their level of effort and engagement, to ensure that the safety measurements are satisfied and the patient does not get injured. Thus, the choice of the type of the controller is a crucial factor in their design, as it will determine the interaction behaviour and stability. It has been signified that, the interaction should be optimized such that the patient is actively engaged in the therapy [13]. Active engagement has the potential of tapping into their neural plasticity to improve motor functions or to lead to improved functional motor trajectory [14]. In order to ensure engagement, a feedback from the user can be utilized to identify their participation level and the assistance they require to do a specific task. This feedback can help the main controller detect users' true intentions and their states, and react properly. (e.g., the robot can either (i) utilize this information along with a virtual contact model and the users inputs to generate a reference trajectory that keeps the patient engaged, or (ii) actively adapt its contact model parameters to tune assistance level, and enhance the users' engagement [15,16]). Such a feedback

can be obtained by collecting and processing data from either biological or non-biological sensors that go into contact with human. Examples of such sensors include electroencephalogram (EEG), Electrooculography (EOG), Electromyogram (EMG), or camera-based head and eye motion tracking, etc., [17, 18]. Force/torque sensors are generally used in HRI to measure the force/torque exchanged at the user/robot interface and accordingly the state of interaction to activate the robot [19].

An issue with the physically interactive robots may arise when the patients are not capable of initiating the required motion or generating the required force to activate the robot. Therefore, the robot would not be able to rapidly detect the user's true states and intentions, and respond accordingly at the correct time. Moreover, people with disability might exhibit involuntarily movements, and input misleading commands due to muscle spasm, weakness, stiffness, unbalanced contractions, etc. Therefore, the user might not experience enough support to initiate the motion or might experience impulsive response due to the sudden state change which are not preferable from a therapeutic perspective. Thus, relying on just a single source of information might not be sufficient to predict user's intention/state, especially if that source is located in a distant muscle relative to the source of motion.

Electromyograms (EMG) are a family of biological sensors measuring muscular activity that can convey user's intention. These intention data, if correctly interpreted, can be utilized as further communication means between the human and the robot. By processing EMG signals before or during the early stages of the motion initiation, the controller can provide continuous and non-impulsive commands without requiring the patient to exert the full force for initiation of the motion, resulting in an improved gait training performance. To correctly interpret the EMG signals, intelligent learning methods can be utilized to distinguish, categorize, and relate EMG measurements to appropriate corresponding intended motions [4, 20]. Intent recognition can be considered as a supervised classification learning problem with known inputs and outputs,

solving which provides the best model to identify the user’s intention. At the first step of this work, Classical Machine Learning (C-ML) models are explored as solutions to this learning problem. The utilized C-ML models employ “Feature Engineering” as the feature extraction methodology, which is built upon manually selected features of windows of EMG signals. To enhance the detection performance, generalization and adaptation capabilities, Deep Learning (DL) models are further investigated. The utilized DL models employ the “Feature Learning” structure to automatically select differentiating features in each layer of learning procedure. In order to demonstrate the performance of such EMG-based intention detection systems in an HRI scenario, a computer-simulation environment with real humans’ data is developed. In this simulation a gait training robot is modeled and the interaction control is regulated using the detected intents and states from sEMG and IMU data collected from a human. Multiple methods of augmenting the interaction control system with the intention detection system are investigated and the best method that improves the main controller in a most intuitive way is identified.

1.1 Project Background

An advanced gait training robot, was originally designed and developed at Carleton University’s Advanced Biomechanics and Locomotion Laboratory (ABL). Fig. 1.1 shows the latest generation of this robot called “SoloWalk”, which was further developed and designed by Gaittronics. Inc [21]. SoloWalk facilitates overground and balance training by providing assistance as demanded by the user through its admittance control system. Using a virtual mass-spring-damper model, this controller translates the force applied on its user-robot interface to a reference position on a target trajectory that then the inner position control loop of the robot follows. SoloWalk is also specially designed to ensure stable user-robot interactions, particularly when

the robot lowers its apparent mass to the user in order to make it easier to move the robot [8]. The mentioned displaying of a small mass and application of pushing forces to make the device easy to move may not be appropriate for some patients or from a therapeutic perspective. Thus, in earlier studies on this device, it was proposed to augment the controller by a logic based on information from sEMG modules that can provide a better understanding of the user's intention or exertion level.

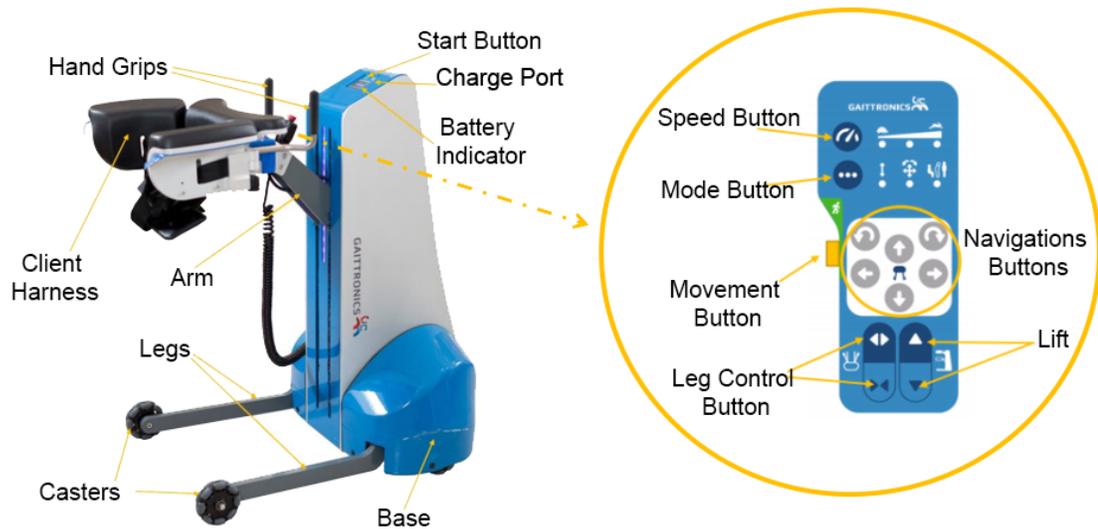


Figure 1.1: SoloWalk's layout, image from [21] - Permission to include the image was obtained from GaitTronics Inc.

The research at hand investigates the integration of the intention-detection system with a model of the Solowalk. This integration can improve upon the main controller by augmenting robot's intelligence through detecting user's intentions. Another proposal for developing next generations of the SoloWalk is to use this intention detection system to facilitate the initiation or continuation of motion by adding an exoskeleton to guide patients' gait movement in a systematic manner.

Because of the COVID19 outbreak in the year 2019-2020, the author was inevitably constrained to simulate this experiment instead of performing it empirically using SoloWalk. The methodology to do so is detailed in Chapter 6. To prove the

proposed concept and to ensure safety of everyone in the unprecedented situation of Coronavirus outbreak, all the experiments and data are collected from healthy subjects.

In addition, outcomes of the experiments lead one to believe that the designed detecting system can be adopted by many robotic platforms that aim to integrate the users' lower limb intention data into their controller for a more intuitive mode adaptation.

1.2 Thesis Objectives

SoloWalk is used to provide assistance for rehabilitation of several forms of lower-limb motor disabilities, e.g. individuals diagnosed with CP. Individuals with CP or severe motor disabilities often have difficulty initiating and selectively controlling motion. On its own, Solowalk or any other rehabilitation robot with admittance controller that relies only on the force sensor to detect the states and intentions of users' with disability cannot direct the initiation of the motion smoothly due to the challenges explained in Section 1. Therefore, an intention detection system is proposed to facilitate the robot's interaction with patients with severe disability. The biological signal coming from EMG sensors can be integrated into an admittance controller to address this issue. Moreover, at the moment the constraints of motion are dictated by a care-giver operating the robot using an attached control pad which can be seen in Fig. 1.1. One of the beneficial effects of this research is to transfer this task from the care-giver to the intelligent intention detector logic.

Through several interviews with physiotherapists and medical experts, the project at hand was defined and developed on a conceptual level. The following decisions on which improvements to focus on and how to set up the experiment were made on the basis of the aforementioned consultations: (i) to enhance the SoloWalk robot via

adding the initial intention detection capability, (ii) the phase of motion in which the patients requires the most assistance (onset of of motion), (iii) the manner in which the robot provides assistance to the patient, (iv) the means of collection of information from the patient, and (v) the muscles to monitor. The evaluation of these conversations resulted in establishing the requirements, features and design of the experiment and the expected functionalities of the robot that are addressed throughout this work. This thesis uses an EMG-based intention detection strategy to augment an admittance controller that assists the rehabilitation process of lower limbs. This approach can make the robot smart and responsive to users intentions prior to the initiation of the motion and during the motion.

This research aims to:

1. Design an intention detection algorithm to detect intention from lower limb using surface Electromyogram (sEMG), assess the reliability and performance of C-ML and DL methods, and improve the robustness and adaptability of the models when applied to new subjects it has not seen before.
2. Find the best scenarios that simulate the patients' real-life circumstances and warrant well-founded results in real world applications and identify the most desirable configuration and number of sensors for highest comfort and performance by training the intention detection algorithm through those scenarios.
3. Utilize the designed, trained and optimized intelligent intention detection algorithm to improve the interaction controller, and integrate the detected direction data into an admittance controller optimally.
4. Develop a computer simulated environment, and a real human data in the loop simulation to validate the performance of the intention detection algorithm and the robot controller.

1.3 Thesis Contributions

The major contributions presented in this thesis include:

- Formulating a model of the HRI of an admittance controlled assistive device (SoloWalk) supporting the walk of patients with motor impairment, plus an sEMG-based intention detection module. This research, though at the preliminary level, will provide the basis for future experiments with the target population.
- Application of C-ML with a feature engineering structure to detect the intention to move in various directions from only EMG signals during several walking scenarios. The presented work, to the best of knowledge of the author, is the first work on walk direction detection based on sEMG sensors.
- Application of DL with a feature learning structure to detect the intention to move in various directions from only EMG signals during several walking scenarios. Incorporation of an advanced transfer learning based DL method to ensure robustness of the model to inter-user data and its adaptation to the new users. Identification of the optimum configuration of muscles recruited for the walking intention detection.
- Development of the experimental setup and plans to investigate the accuracy of the above methods, running and analysis of the results obtained from 9 participants.
- Development and analysis of an interactive simulation environment of the robotic experiment to connect the DL based intention detection to a dynamic environment (including real data of the Subject, IMU and EMG).

- Identification of the best augmentation methodology that can augment an admittance controller to provide more efficient assistance for patients with motor impairment to follow a desired therapeutic path.

1.4 Thesis Outline

- **Chapter 2:** presents a thorough literature review on several families of control methodologies applied to rehabilitation robots, intention detection methodologies based on C-ML and DL, and biomedical applications of intention detection.
- **Chapter 3:** details the experiment and scenarios specifically designed for the lower limb rehabilitation task. The fundamentals of sEMG signal pre-processing steps to prepare data are described.
- **Chapter 4:** introduces the C-ML frameworks and provides insight into the procedure to choose significant features. The most well known C-ML algorithms are implemented on the scenarios based on the SoloWalk setup. A comparison of various chosen feature sets and an investigation of different classifiers are detailed. A comparative analysis of the level of generalization of different implementation algorithms is performed.
- **Chapter 5:** covers the fundamentals of DL, Convolutional Neural Networks (CNNs) and the designed model's architecture. A comparative study of different implementation algorithms in terms of generalization and adaptation is presented.
- **Chapter 6:** covers the development procedure of the controller and the simulation environment. The general admittance control augmented with the intention detection system that utilizes IMU and sEMG signals are described. Tests on

the simulation environment, including on the functionality of the built model and the controller are covered. A discussion on the advantages of various forms of integration of the sEMG intention detection information into the control algorithm is presented.

- **Chapter 7:** concludes with a summary of the contributions and recommendations for future work. This includes essential features that have not yet been implemented due to time constraints and ideas for future directions of the project.

Chapter 2

Background

In this chapter, at first, the notion behind the demand for a proper and intelligent HRI system is described, and the required infrastructure for such a system is explained in Section 2.1. Then, Section 2.2 goes through the process to choose a proper communication means between human and robot. In Section 2.3, a thorough literature review on intention detection approaches is presented. In Section 2.4, the most common methodologies for a stable HRI in co-activity mode are discussed. Finally, in Section 2.5, a broad overview of the rehabilitation and assistive device applications that incorporate different HRI methodologies and the available samples integrating intention detection into these applications are explained.

2.1 Introduction

With the growth of rehabilitation robotics in which robots work closely with humans, it is necessary to have complex and intelligent HRI that enables the human and the robot to share an environment and their roles in performing their tasks in the most safe and intuitive manner. In any shared environment, humans and robots can interact through various means with varying levels of interaction and sophistication [22].

Three distinct modes are recognized when a human and a robot interact: (i) manual, (ii) autonomous, and (iii) shared or co-activity. In both manual or autonomous modes, a robot or human completes a given task independently. In the manual mode, the human takes the leading role, and the robot follows/emulates the human's actions closely in performing a task. The robot becomes the executor and not an active planner, governor, or controller of the task. In the autonomous mode, the robot takes the lead, governs, plans, and controls the execution of the task, while the human remains passive during interaction (although it has the authority to impose high-level commands). The shared mode has an edge over the two previous approaches because it takes into consideration the cooperation or collaboration between the human and robot through various means of communication. HRI in a co-activity mode is a research field that can be applied in diverse biomedical areas, such as assistive, rehabilitation technology, and robot-assisted surgery [23]. This is particularly essential for rehabilitation tasks where the user both needs support to execute a task and needs to be actively engaged (lead the action) during performing that task so as to promote motor functioning to receive maximum recovery. The third mode needs at least a single means of exchange of information between the robot and the patient to ensure that they can intuitively understand and communicate with each other. Sensing technologies can be used to obtain the communication means required for collaboration [24]. The performance of the HRI in a co-activity mode relies on: (i) the means of communication, (ii) the method used to interpret the collected data and estimate human's intentions, and (iii) the method that the robot utilizes to react to human's intentions while preserving the safety of its human partner. The following Sections (2.2, 2.3, 2.4) go through the process to choose a proper communication means, a thorough literature review on intention detection approaches and the most common methodologies for a stable HRI in co-activity mode.

2.2 Human-Robot Communication Means

Researchers have used various signs in human behavior to collect data from that can be interpreted as the human's intentions and states, e.g. head pose, eye gaze, hand position and orientation, speech, kinematic parameters (such as velocity and force), and biological signals (e.g., Electrooculogram (EOG), Electrocorticogram (ECoG), EEG, Magnetoencephalography (MEG), and Electromyogram (EMG)) [24].

Initially, EEG sensors, which record the electrical activity of the brain, were considered for this study. Although EEG signals have shown their potential in detecting the intentions of the users for particular classification tasks, they face various challenges in applications with a more complex nature [25]. Since the brain is responsible for processing an immense amount of information coming from multiple sensory and cognitive sources in the human body, the signals relevant to the intended direction are closely correlated with many other signals. As a result, decomposition of EEG signals to the relevant and irrelevant to the intended direction is challenging. There are some activities involved in this study which introduce complexity to the mentioned decomposition. Examples include: dynamic walking, thinking (for interacting with a robotic platform), physical and cognitive rehabilitation, and everyday normal behaviors (e.g. facial expression, blinking, etc.). EEG sensors, especially those with non-invasive electrodes, are particularly susceptible to noise. Therefore, it is required to use high-quality and consequently high-cost EEG devices with more electrodes. These take longer to adjust and often end up having aesthetically unfavorable geometric shapes. All of these factors acted as further downsides of using EEG sensors for the study at hand.

In contrast, muscles, as the executors of human actions, are reliable sources of conveying the relevant brain signals that contain information about the execution of the task. EMG, also known as the myoelectrical signal, is a bio-potential signal

resulting from muscular activity. A muscle is composed of many motor units (MU). The “discharge” or “firing” of each MU generates a signal that contributes to “Motor Unit Action Potential” (MUAP). EMG identifies the generation and propagation of these MUAPs through sensing muscle’s electrical activity from the skin surface. It has been observed that the time delay between action-potential initiation and muscle force buildup, referred to “Electro-Mechanical Delay” (EMD), can reach 100 ms depending on the muscle size. This EMD phenomenon suggests that the movement from humans can be anticipated by detecting the signal with sEMG prior to the actual movement [26]. This way, muscle activities conveyed through sEMGs signals can be interpreted as the users’ intentions [27]. This signal can be sensed by utilizing either non-invasive or invasive electrodes. Although the invasive electrodes provide less noisy data, the comfort of the non-invasive ones while having an acceptable signal quality has given these surface EMG (sEMG) electrodes a significant advantage. Moreover, the correlation between signals relevant to the motion and other irrelevant signals (e.g. facial expressions, thinking, etc.) is lower for sEMG sensors in comparison to EEG sensors. Thus, EMG sensors are better capable of capturing/predicting gait movement and intended motion through sensing MUAPs. A selective number of muscles can be monitored to detect the relevant intended direction. This means that sEMG sensors end up with a lower cost and are more easily adjustable. In addition, their smaller shape and portability make them a reasonable alternative to brain signal sensors for the intention detection task. Therefore, this research aims to utilize sEMG sensors to detect the users’ intentions.

2.3 Intention Detection

The EMG-based methodologies for human intention identification are broadly broken down into classification and regression models. Upper limb intention detection

through pattern recognition and classification has been thoroughly studied in the past decades. These studies typically involve sensing isometric contraction, defined as an active contraction in which the muscle’s length stays constant. A walking task, on the contrary, mainly incorporates the lower limbs. In comparison to the upper limb, the research around lower limb pattern recognition is multifaceted. There is a broad and growing body of literature on lower limb gesture recognition and intention detection to control assistive robots using sEMG signals [4, 20, 28, 29]. In addition to studies on isometric contraction of lower limb [30], the lower limb pattern recognition applications are also concerned with isotonic contractions. Isotonic contraction happens when the muscle changes length while it maintains constant tension. Isotonic contraction can be either concentric (the muscle shortens) or eccentric (muscle lengthens) [29].

2.3.1 Feature Engineering and Classical Machine Learning approach

Classical machine learning approaches (C-ML), also referred to as traditional or conventional machine learning approaches, are a broad family of classifiers that consist of K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA), Decision Trees and Random Forests (RF), Multi-Layer Perceptron (MLP) (a Neural Network (NN) with only a single hidden layer), and ensemble models. C-ML’s fundamental ideology is to establish specific features that can help the aforementioned classifiers differentiate between different gestures when dealing with the intention detection problem. This approach is called feature engineering since it explicitly requires algorithms to look for particular features. Although this approach has shown promising performance, it struggles to look for features that have not been explicitly prognosticated in advance.

The main factors that affect the accuracy of the classification of sEMG signals include the quality of the raw sEMG signal itself and the used processing techniques. Correct electrode location, appropriate selection of signal channels, and proper selection of experiment procedures to recognize limb motion are other noteworthy factors that can improve the signal quality and lead to higher classification accuracy. sEMG signals are inherently subject to noise. Due to their type of adhesion to the body, they can easily be disturbed by a shift in the electrode location, loose electrode-skin contact, and environmental noises. These flaws are more severe when the signal is collected from a person with some degree of disability because of sudden spastic movements and muscles' extensive contractions. To mitigate these issues and improve the accuracy, C-ML methods employ pre-processing steps. Pre-processing includes filtering the acquired data and segmenting data into windows. Afterward, these sample windows can be used in the feature extraction step to convert sEMG signals to a compact and informative set of features. The most significant features can be selected either manually or by using statistical approaches like Principal Component Analysis (PCA). PCA also has been used as a dimensionality reduction technique throughout the literature.

The process of choosing the best features that help the classifier differentiate between different tasks has a deep root in pattern recognition studies. Many researchers use the road map provided by Phinyomark et al. to select their features of choice. They examined thirty-seven sEMG features collected from both Time-Domain (TD) and Frequency-Domain (FD). They concluded that many of these features are redundant; therefore, they provided a set of recommendations based on the features' effectiveness in distinguishing different gesture classes. As they suggested, mean absolute value (MAV), Wavelength(WL), Willison Amplitude (WAMP), Auto-Regressive coefficients (AR), Mean absolute value slope (MAVS) are the most significant features from the TD. They also indicated that FD features are not decisive enough

for EMG signal classification. However, they recognize some of the FD features like Mean Frequency (MNF) and Power Spectrum Ratio (PSR) for their ability to assist in classification via EMG input vector [31].

Study of human pattern recognition for upper limb based on C-ML methods roots back to 1993 with Hudgin's study on classifying four hand gestures from two sEMG channels with MLP classifier based on five features from TD [32]. Researchers afterward investigated other classifiers with either the same or different sets of features to improve the classification accuracy. In 1999, Englehart et al. stated that the performance of feature extraction and dimensionality reduction is highly dependent on the structure of a classifier [33]. They indicated this by illustrating that when using Time-Frequency Domain (TFD) features subject to PCA, the LDA exhibits better generalization performance than the MLP classifier, although MLP is known to be superior in describing nonlinear class boundaries. This is the result of diminishing the nonlinearity in the higher dimensional feature set of TFD. Later, they demonstrated that four channels of myoelectric data considerably enhance the classification accuracy compared to one or two channels [34]. Further, they reduced the error in prediction by applying a post-processing technique, called majority vote window, to the classifiers' predictions to reduce the number of false predictions [35]. The first study that successfully achieved a high classification accuracy (70-80%) on a large number of hand gestures (52 gestures on the Ninapro dataset [36]) was proposed by Kuzborskij et al. in 2012. They used an SVM classifier with Radial Basis Function (RBF) kernel on a set of TD and FD features extracted from eight-channel of myoelectric signals [37]. Later, Atzori et al. improved the previous researches by using an RF classifier. They achieved an average accuracy of 75.32% on a linear combination of features [38]. In the same year, Gijssberts improved classification performance by 5% through combining extracted features from sEMG and Acceleration signals [39]. More

recent studies focused on using DL to classify hand gestures, which is summarized in the next section.

There is a broad and growing body of literature on lower limb gesture recognition and intention detection to control assistive robots using sEMG signals [4, 20, 28, 29]. Lyons et al. illustrated that they could achieve similar performance for lower body pattern recognition tasks by applying a similar pattern recognition algorithm with a one-to-one correspondence to hand gestures [30]. Although this study considered an isometric contraction in the patterns of the Lower limb, most of the other studies on the topic of lower leg intention detection deal with isotonic contractions in walking practices [29], e.g., gait phase recognition and mode predictions (sitting, standing, ramp ascent/descent, etc.) [28, 40]. Different researchers also investigated the effects of the choice of sensors, their fusion, and their configuration (unilateral and bilateral). Kilicarslan et al. achieved high accuracy by utilizing EEG sensors to detect a paraplegic subject's intentions in two types of tasks. Zhang et al. fused mechanical measurement data (from Ground Reaction Force (GRF) and Kinematic information) and neuromuscular data to identify three activities (level-ground walking, sitting, and standing) in a study on patients with multiple sclerosis. Their designed system was capable of predicting the users' intention with a high accuracy rate in static states while also being able to correctly predict the activity transitions about 100 to 130 ms before the actual transition [28]. Hu et al. used data collected bilaterally from lower limb EMG and joint and limb kinematics to detect the smooth transition between locomotor activities. They were able to demonstrate that fusing different sensors while taking into account the manner in which bilateral sensor data is collected can enhance the classifiers' ability to predict the true activity [41].

However, to the extent of the knowledge of the author, further investigation is

needed to incorporate C-ML Algorithms to detect the user’s intended direction of motion based on the sEMG signal. In Chapter 4, this study aims to explore the best-founded C-ML classifiers that can detect the user’s at least nine intended directions of motion based on the sEMG both during and before the onset of motion. The inter-subject robustness of the models are evaluated by employing two different implementation strategies.

2.3.2 Feature Learning and Deep Learning approach

Lately, DL methods have attracted the spotlight in sEMG-based gesture recognition, relying on their significant performance in computer vision and speech recognition tasks. The main advantage of DL methods in comparison to C-ML methods is their ability to incorporate feature learning; a consistent part of the traditional pipeline can be entrusted to the algorithm’s training, which has enough capacity to learn beneficial feature representations on its own. This helps remove the reliance on pre-processing steps and the precise design of discriminative features. Existing DL architectures are mainly built based on two well-known architectures: the Convolutional Neural Network (CNN), which works best for detecting spatial information of the signal, and the Recurrent Neural Network (RNN), which allows exploiting the sequential nature of the data and temporal information. However, recent research has pointed out that CNN can achieve what Long Short-Term Memory (LSTM) (a well-known family of RNN) has been great at, namely predicting sequences, but in a much faster, more computationally efficient manner [42]. Due to the RNN models’ need for very high-level embedded hardware, their complex architecture that results in longer training time, and their less value per required effort, CNN approach has been chosen for this

study.

The first end-to-end DL structure was proposed to solve the inter-user variability in detecting six common hand gestures. They used a CNN model consisting of five convolutional layers and evaluated it with the SVM algorithm of C-ML approach in the adaptive and nonadaptive modes. They demonstrated that CNN could outperform the SVM model in both modes [43,44]. Atzori et al. provided a baseline for the classification of hand gestures collected from sEMG sensors based on CNN models on a large number of gestures and subjects. Their results showed that their proposed simple CNN classification model’s accuracy is comparable on average to those obtained via the C-ML approach. Although, it does not perform better than the best performance achieved on the same problem using an RF classifier of C-ML approach [45].

Further, Geng et al. and Wei et al. incorporated dropout and Batch Normalization (BN) methods in their CNN structure to enhance the results across different datasets by avoiding the problems of overfitting [46,47]. Geng et al. utilized two fully connected layers, while Wei et al. used a multi-stream CNN architecture. Wei et al. proposed the multi-stream CNN architecture to evaluate the validity of the empirical theory that a small group of muscles plays a significant role in some movements. “divide-and-conquer” strategy used in multi-stream CNN architecture has the advantage of learning the correlation between individual muscles and specific gestures with less effort [47]. Apart from model architecture, they both investigated the advantages of using a high-density electrode array to capture EMG data. Thus, they demonstrated that using more advanced CNN models with instantaneous EMG images can improve upon previous C-ML results, achieving a noticeably higher accuracy.

Later works focused on generalizing the learning method to cover inter-session and inter-subject scenarios. For the method to be able to perform as well in these scenarios, Zhai et al. proposed a CNN-based model that takes the down-sampled spectrogram of the sEMG’s signal segments as the input. They further improved their model’s performance by updating the network weights using the predictions of previous sessions corrected by the majority voting method [48]. Du et al. presumed that the information related to the specific characteristic of each subject or session corresponds to the statistical information in the BN layer, whereas the gesture-related information (labels) are preserved in the network model weights. Therefore, they used the Adaptive Batch Normalization (AdaBN) technique during each test to update only the normalization statistics for each subject using a few unlabeled data. They substantiated that this adaptation method improved the model’s performance [49]. Ding et al. proposed a parallel multiple-scale CNN architecture with different sizes of the receptive field and larger kernel size. Ding’s method was validated on the same dataset as Zhai’s, and they illustrated that their method could slightly improve Zhai’s method, although with more computational cost [50]. Chen et al. used the CNN feature extraction method to improve the accuracy of human motion classification in inter-subject tests. Their method used CNN’s available learned features and fed them to the C-ML algorithms as the last layer instead of feeding them to a fully connected layer. They demonstrated that this method could improve the accuracy of the generalized model [51].

Ulysse et al. performed a comprehensive study on hand gesture recognition using CNN and the transfer learning method, which employed AdaBN in a multi-stream CNN network [52]. The same hypothesis used by Du et al. [49] was used in their algorithm to be able to distinguish between weights corresponding to gestures and the BN weights corresponding to subjects. Furthermore, They investigated three

different DL networks that utilized three distinct modalities as input (raw EMG, Spectrograms, and Continuous Wavelet Transform (CWT)), either with or without transfer learning. These studies proved that their proposed transfer learning scheme could considerably enhance the performance of the designed CNN by learning general features across different subjects. They also denoted that the DL models can almost always outperform the ML models. Various modules of input lead to slightly different results based on the specific characteristics data set it was applied to [52].

The above-given literature review has illustrated that DL methods provide promising solutions for signal pattern recognition tasks. In particular, Ulysse's method demonstrated promising results by generalizing the model to encompass various sessions and different subjects and have the ability to adapt to the new user [52]. As a result, the study at hand incorporates the methodology developed by Ulysse et al. as a framework to design and train a model to detect the direction of motion through monitoring lower limb muscle motion. A detailed explanation of the developed CNN architecture and experimental results are described in Chapter 5.

2.4 Human-Robot Interaction Control

Rehabilitation robots need interaction control as they are cooperating with humans in a shared mode. The controller is of great significance in designing an HRI system, as it affects the interaction behavior and stability. In rehabilitation robots, the interactive controller should be optimized to guarantee engagement through reinstating neural plasticity and enhancing motor learning. The more responsive the controller is, the more engaged the user would feel during the interaction. The three most common

control strategies that are widely used in rehabilitation and assistive devices are listed below:

- **Impedance and admittance control** introduced by Hogan, have been the foundation of most interaction controllers [53]. The main objective of these controllers is to model the dynamic interaction between the robot and the environment instead of blindly regulating the relative position or force. The real interaction impedance/admittance is simulated via a mass-spring-damper system model acting at the interaction point. As a result, the contact parameters can be manually programmed to assist motion (by changing the spring constant) or resist it (via changing the damper constant) while performing particular tasks.
- **EMG-based control** uses electromyography (EMG) to measure patient's muscle activity and indicate the user's intention or muscle force. The robot can be programmed to respond to these signals by assisting the patient when the intention is detected (e.g., the patient flexes their muscle), or some amount of force is perceived based on some predefined or adaptive measured threshold. Implementing this type of controller ensures that the patient is actively engaged in the therapy. For instance, EMG activation in the ankle-foot orthosis is utilized to drive a pneumatic actuation system in [54].
- **Adaptive control** changes its behavior over time based on a predefined law or additional information collected during operation. For example, the admittance controller's degree of assistance can be tuned based on the patients' performance or based on the human subjects' disability level and voluntary participation during the activity [15].

2.5 Rehabilitation and Assistive Devices Application

The combination of collaborative human-robot interaction with the intention detection algorithms provides new sensory capabilities for designing sophisticated and intelligent HRI. The improvements in acquiring additional sensory feedback have paved the way for research toward a smart robot that can understand, assist, and monitor humans. This is particularly advantageous for rehabilitation robots and assistive devices.

Many researchers have investigated the use of sEMG-based intention detection for HRI based on the upper limb motion. However, as pointed out in the previous section, task recognition for the lower limbs is yet to be studied in depth for various applications. This study aims to propose an experiment that aims to fill this gap by investigating the feasibility of applying direction intention detection system to drive a gait rehabilitation robot. Gait rehabilitation robots can be categorized into three main branches based on their function and design: (i) End-effector-type robots that attach to the user's feet and guide the motion of the user's feet, (ii) exoskeleton-type robots that attach parallel to the patient's lower extremities, and (iii) actuated mobile gait trainers that enable overground gait training. The same concept applies to assistive devices, especially active prosthetic devices. Intelligent prosthetics, controlled through monitoring the patient's brain or distant muscle activities, have enhanced the quality of life of people with disabilities.

Hereunder a review of the sEMG-based HRI applications and samples of similar HRI methodologies is presented. In any of the applications mentioned above, there

are multiple ways to integrate and adjust the input-output relationships to develop a control strategy. For instance, sEMG signals can detect the gait phase, the task, the environment or it can provide torque estimations.

- **End-effector-type Gait Rehabilitation Robots** can be categorized into two types: (i) end-effector-based overground Walking robots and (ii) end-effector-based robots for sitting or bed-bound therapy. The former roots back to the invention of the therapeutic Body-Weight Supported Treadmill Training (BWSTT) as an intervention to recover autonomy and neural plasticity for stroke and spinal cord patients [55]. The passive BWSTT platforms were both physically-demanding, repetitive, task-specific, and, therefore, appropriate for automation. Colombo et al. used this idea and designed a robot that could lift the patient upright while actively and automatically moving their lower extremities through a trajectory that represents a gait pattern [56]. More advanced models of End-Effector type robots like Haptic Walker and G-EO are also designed to add more Degree of Freedom (DoF), providing a more versatile environment for the safe practice of different activities like stair climbing and walking on even surfaces [57, 58]. The latter category of robots is administered for patients who are not yet mobile enough to enable early therapy while the patients are still in bed. The MotionMaker, Lambda, Rutgers' Ankle, and the Virtual Gait Rehabilitation Robot are some examples. [59–63].

Integrating biological signals to enhance the general collaborative control strategy between humans and robots dominates this research area in recent years. For example, Recently, I. L. Petersen et al. proposed a new method for early stages of active rehabilitation by combining the end-effector-type robots with sEMG triggered functional electrical stimulation (FES), proving its impact on

high success rate during the experiments [64].

- **Exoskeleton-type Rehabilitation Robots** are attached parallel to the patient's leg and try to mimic the human leg structure by supplying interactive joint torque to guide the attached leg through an appropriate trajectory. Designing suitable control algorithms to enhance the patient's involvement during gait rehabilitation therapy is the main objective of these robots, along with lightness and compliance. Lokomat, AutoAmbulator, PAM-POGO, LOPES are some examples of these types of robots [56, 65–68].

Many researchers in this field have also tried to integrate sEMG signals to Exoskeletons' driving logic to help control the lower body based on neuromuscular signals. For instance, the Hybrid Assistive Limb (HAL) used sEMG signals to calculate the virtual torque to assist the lower limb movement by adjusting the impedance around the knee joint [69]. In another study, wearable EMG and kinematic sensors are used to design a fuzzy-based control algorithm for lower limb exoskeletons during sit-to-stand, and stand-to-sit movements [70].

- **Actuated Mobile Gait Trainers (AMGT)** are mainly designed to allow users to practice overground walking. They mainly consist of an actuated mobile base, a BWS mechanism that is fixed to the mobile base, and, in some cases, an exoskeleton or manipulator to assist the swing leg. One of the most sophisticated examples of AMGT, KineAssist, uses a constructed mechanical arm to support the patient during over-ground walks [71]. Having an exoskeleton or manipulator to assist the swing leg can enhance AMGT robots' capability to have patients with minimal muscle control utilize it for overground walking, like the one used in WalkTrainer [72].

Recently, A new autonomous robot-based platform (a smart walker compounded

with an exoskeleton) is designed to provide a combination of Peripheral Nervous System (PNS) and Central Nervous System (CNS) training for individuals with CP. This robotic rehabilitation system has a multi-model human-robot interface (MHRI) controller, which gives it high versatility and provides greater adaptability to the patient's needs during the therapy. The proposed MHRI system consists of: an EEG acquisition unit which takes into account the users' intentions, inertial measurement units (IMUs) to improve the patient's postural control, and a Laser Range Finder (LRF) to measure the human locomotor patterns. The controller unit uses the acquired information from all the MHRI sensors, integrates it with the information collected from the smart walker, and controls the whole robotic platform according to the users' parameters during the therapy [73, 74].

- **prostheses** are type of assistive devices that replace a missing body part, which may be lost through trauma, disease, or a condition present at birth. EMG-based powered intelligent prosthesis are widely investigated in the past decades to improve the symmetric walking pattern and metabolic energy efficiency. There are mainly two different approaches to have a smart device assisting a person with a transfemoral amputee. One of the control strategies for the powered intelligent prostheses is to estimate individual muscle torque or muscle stiffness from sEMG signals [75, 76]. Another approach is to use a hierarchical control strategy that mainly relies on pattern recognition from sEMG or sEMG fused with other mechanical sensors. Gait phase detection and/or environment prediction are the main objectives of these motion-intention recognition algorithms [77–79]. With the intention recognition system's aim, the controller perceives users' intentions and can adjust its parameters accordingly.

Fig. 2.1 shows samples of utilized pattern recognition capabilities in various biomedical and rehabilitation applications.

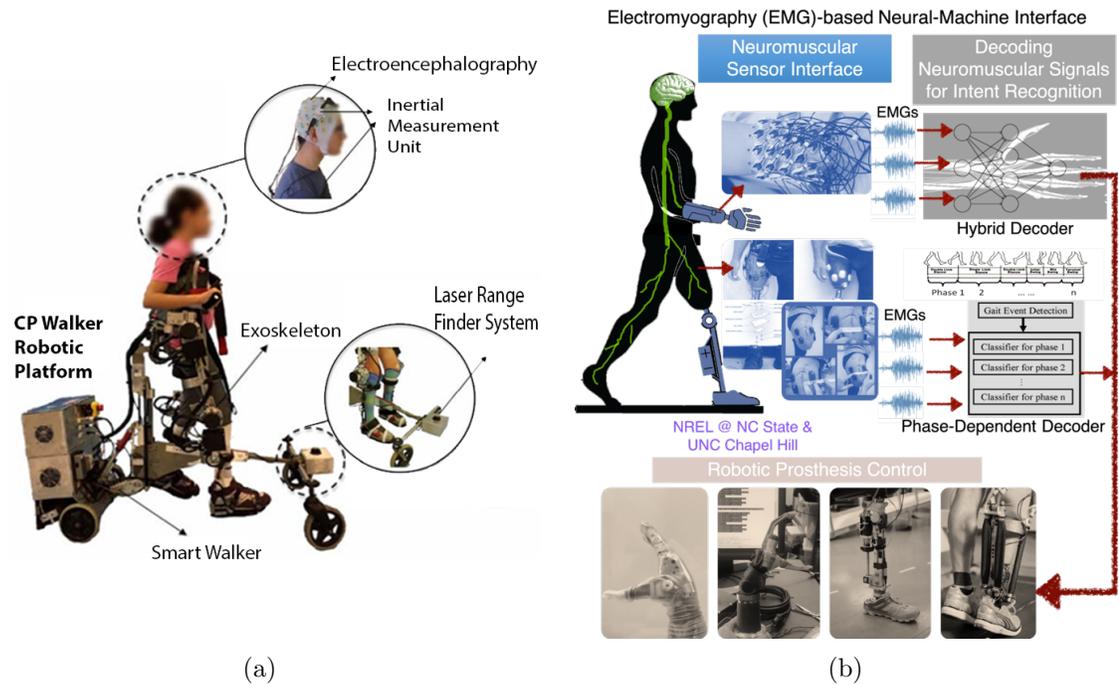


Figure 2.1: Samples of intention detection in biomedical control applications (a) Over-ground type robot with an Exoskeleton and EEG-based movement onset detection system, image from [73] - Permission to include the image was obtained from Elsevier. (b) Prosthetic robots, image from [80].

Chapter 3

Experiments

Throughout this chapter, the notion behind designing the experiment is detailed along with a description of different scenarios developed to cover many forms of activity commencement and continuation in Section 3.1. The initial phase of the experiment and the developed Graphical User Interface (GUI) system that is programmed for collecting users' data in a synchronized manner with the data acquisition setup are explained in Section 3.2. Further, the process to select the most informative group of muscles and the procedure used to configure the electrodes on the skin properly are described. Eventually, in Section 3.3, the pre-processing steps (segmentation and filtering) that is used similarly for preparing the input data to be utilized with both C-ML and DL techniques are detailed.

3.1 Introduction

Obtaining the user’s intention from sEMG can be categorized as a supervised classification problem, where the input and output of a learning model are known. Such known inputs are commonly collected from an EMG signal sensor through pre-designed experiments. The form of the input to the learning method can differ based on the machine learning approach that is chosen for classification. Specific pre-processing steps are required to prepare the experimental data to be further utilized in each of the employed learning methods. In this research, two phases of experiment are designed in total as: (i) the initial set of experiments in which biological data from sEMG signals are gathered from multiple participants while performing various gestures/movements, and an intention detection model is developed based upon that collected experimental data, and (ii) the second phase of experiment in which the participant are asked to repeat the first phase of the experiment but in a specific order and through a predetermined trajectory. The collected data is processed in a similar fashion to phase one and is used with the developed intention detection model in a simulation environment that models a gait training robot. The second phase of the experiment is detailed in Section 6.5. Different scenarios are designed to fully cover the moving action’s distinct phases. This enables the investigation of the feasibility of the proposed approach during different time slots of actions and through various applications.

3.1.1 Scenarios

The main objective of the intention detection part of this research is to detect the initiation of the motion and its direction before the activity is commenced or during the

motion. Mainly, three different walking-related scenarios are investigated in which the direction of the user’s motion can be detected. This detection is performed based on the patterns in the muscle activities throughout the duration of the motion and before its onset. However, it has been observed that there is a distinction between these patterns in (i) the initial intention phase, (ii) the initiation of motion phase, and (iii) the phase where the user is in motion. The more explicit the distinction between these input phases in training is, the more informative the model will become. Therefore, three different test scenarios (walking-related scenarios) and an extra validity (knee/foot-gesture-related scenario) scenario are planned to investigate each phase by differentiating between muscle patterns during gait cycle [81]. The validity scenario involves isometric knee/foot gestures in static position that closely resemble the hand/arm gestures commonly evaluated in the existing literature [37, 38]. The noteworthy performance of the model in predicting these gestures can be a promising sign of the model’s applicability to detection tasks of more complex gestures/activities. A description of each scenario is detailed below, and a sample of gestures performed in one of the scenarios is illustrated in Fig. 3.1. A complete list of gestures performed in different scenarios is presented in Table A.1 and Figs. A.1 to A.4 of Appendix A.

- **Standing Position scenario** comprises of the participant standing and taking a step towards the desired direction (based on the command of the GUI system). The participant will hold the desired pose for five seconds, and the recording begins when they are located in their pose. Appropriately, this scenario is categorized as a static scenario.
- **Attempted Motion scenario** the participant imagines and generates the force necessary to initiate the motion without actually moving. This scenario is called ”Attempted Motion”. In the absence of the motion, this test is still considered a static scenario. Attempted motion test is required because muscle groups of

many people with different types of disabilities either can't generate the required force to overcome their inertia (due to spasticity) or cannot properly balance the force between their agonist, and antagonist pairs.

- **Dynamic Motion scenario** where the participant goes through the entire motion that involves the participant intending, initiating, and walking with average speed during the time required to perform the task. Therefore, this scenario should be categorized as a dynamic test, called "Dynamic Motion" throughout this thesis.
- **Sitting Position Scenario** which has been foreseen to facilitate the evaluation of the static scenario by having the participant perform some test motion in a "Sitting Position". The outcome of this test demonstrates the applicability of the proposed intention detection algorithms to leg motion.

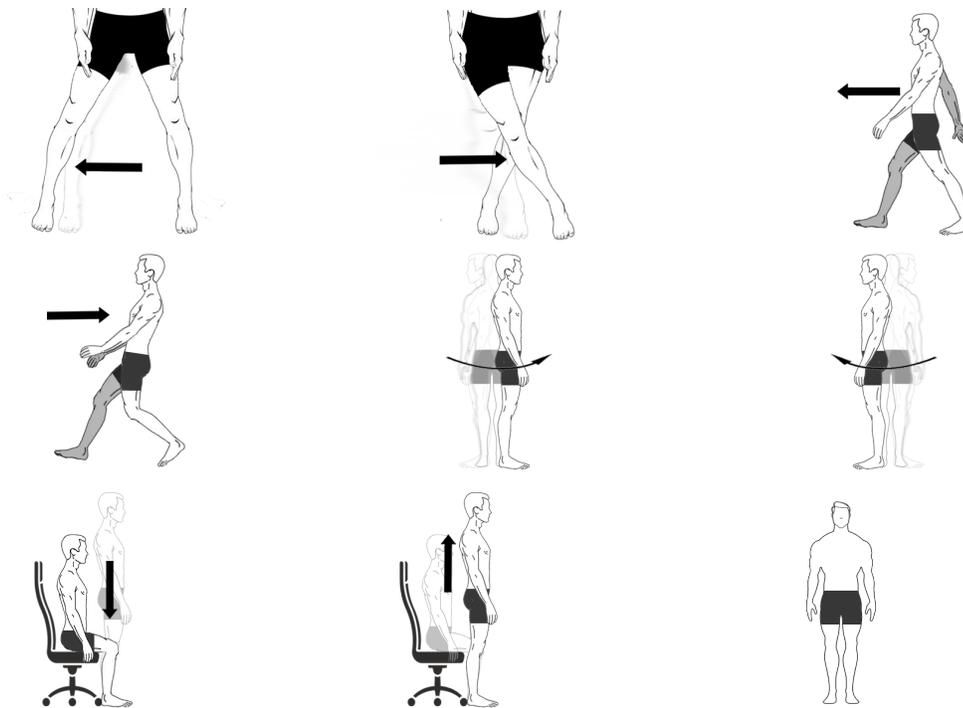


Figure 3.1: Sample of dynamic activities performed in Dynamic Motion scenario

3.2 Experiment

A set of experiments has been designed to collect data for each motion scenario. Data has been collected from nine subjects in five repeating cycles. Each cycle contains at least five seconds of data from the same subject performing each gesture in each scenario. The order of gestures assigned to each subject throughout an experiment is randomly selected in all the cycles. One random cycle, out of the data collected from each participant is put aside as the testing set that none of the developed models have seen before the final evaluation. Informed consent was obtained from all participants, as approved by Carleton University's Research Ethics Committee. After giving this consent, subjects conducted the experiment. The certificate of institutional clearance is attached to Section A.2 of Appendix A.

3.2.1 Processing Steps

Fig. 3.2 depicts the main components of the typical sEMG-based pattern recognition process. The participants are directed to perform a specific task by a GUI. Signals are collected by electrodes placed on the skin over users' muscles, amplified, discretized, and transferred to the computer. Then, in the pre-processing step, segments of data are separated to increase the number of the samples for the learning system, and a final filter is applied on top of that to eliminate noise. The filtered data samples are used to extract features based on the used learning approach (for C-ML approach, gathered features should be manually selected and fed to the system as input, for DL approach, the model learns the features on its own). Finally, the learning model is transferred to be utilized in a controller.

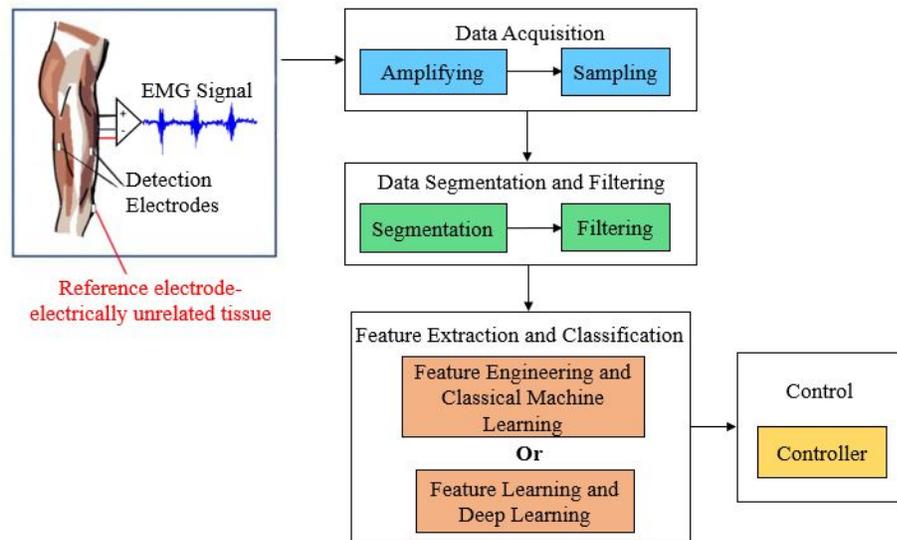


Figure 3.2: Machine Learning Process-General Diagram

3.2.2 Graphical User Interface

A custom GUI program is designed in the Biomechatronic Laboratory of Carleton University to prompt subjects to perform specific activities and collect the raw sEMG and IMU data at the same time. At the start of each recording trial, an image of a specified activity was shown to the participant. They were asked to either hold the gesture they see, try to perform the gesture in the absence of complete motion, or perform the activity they see on the screen for a specific duration. A time progression bar prompted the participant for the activity onset and offset. A sample of such a window is depicted in Fig. 3.3.

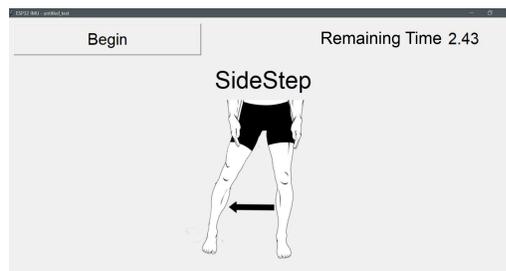


Figure 3.3: Sample of a designed GUI Window

3.2.3 Data Acquisition

Fig. 3.4 shows different hardware setups that were used in this study. The following sub-sections provide a thorough description of each module that has been used.

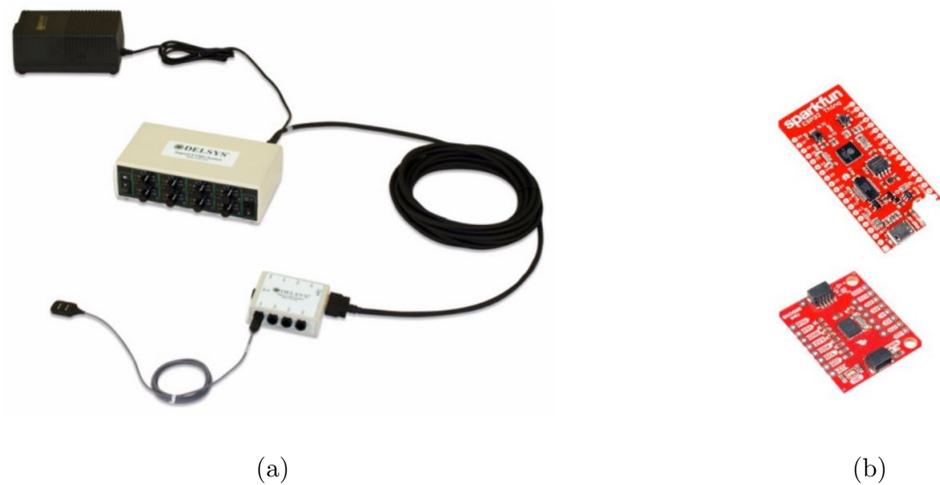


Figure 3.4: (a) EMG data acquisition (b) Custom-made IMU data acquisition setup

EMG Data acquisition

EMG data were recorded through eight EMG channels (each using Delsys DE 2.x series EMG sensors) mounted across the lower limb (Thigh and Shank) of the dominant leg and then processed by the Bagnoli desktop EMG system from Delsys Inc. The measurements from EMG channels were amplified using a Delsys Bagnoli- 8 amplifier to a total gain of 1000. A 16-bit analog-to-digital converter (National Instruments, NI-DAQ-6216) was used to sample the signal at 1000 Hz. These readings were stored on computer hardware by the custom-coded interface program tailored to the applications of this project. Subsequent post-processing and analysis were performed offline after the experiments were performed. A 2-slot adhesive skin interface was applied on each sensor to attach them to the skin firmly.

IMU Data Acquisition

A custom-made IMU reading and transferring setup have been used in this study to track pelvis motion. An **ESP32 – Thing** development board by Sparkfun was used as the microcontroller for the sensor; the IMU data is recorded via a BNO055 IMU; the system is powered by a 3.7V lithium polymer cell, and all components are contained in a 3-D printed box case. The system transmits IMU data to a laptop via WiFi. The IMUs are affixed to the participants' bodies by a belt.

The primary purpose of integrating the IMU sensor in this experiment is because it allows bringing humans' data into the simulation loop via calculating an approximate position of the user's torso for a short duration of time. The identification, calibration, and nullification of the internal and external error parameters of the IMU are critical to ensure an accurate position estimation. The detailed description of the calibration procedure and the fusion method used to remove the accumulating drift of acceleration when integrating to calculate the position for a short duration of time are presented in Appendix C.

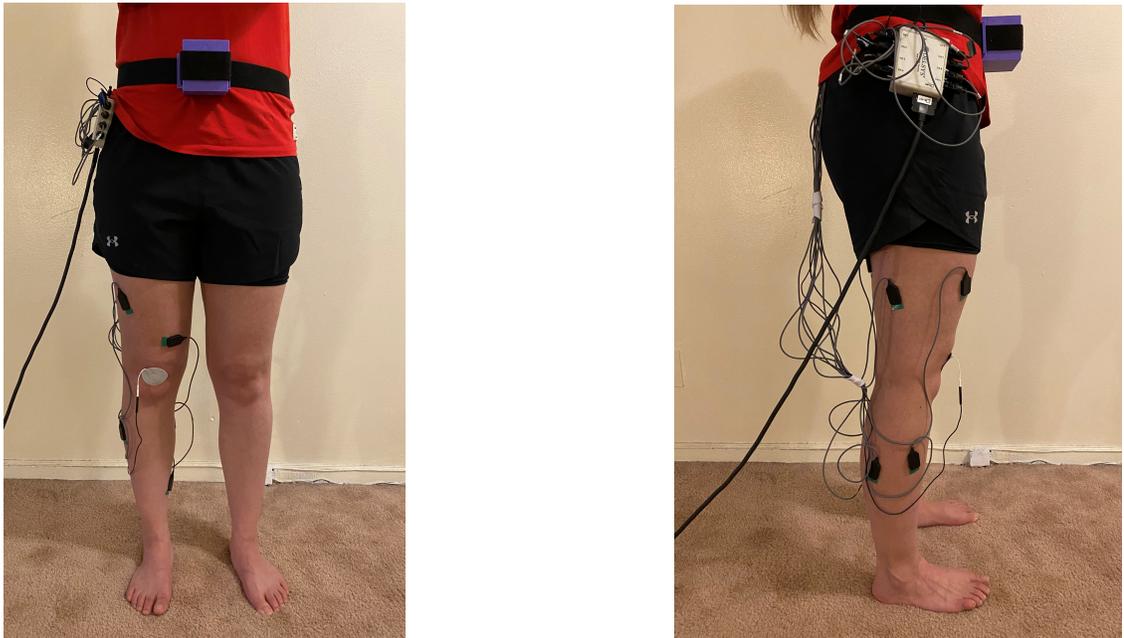
Muscle Selection and Electrode Placement

Different research has shown that positioning sEMG electrodes over specific muscles, over the region surrounding the neuromuscular junctions, or over the tendons' terminal area can alter sEMG estimates. Rainoldi et al. investigated the degree of inter-subject uniformity of the location of the innervation zone in 13 superficial muscles of the lower limb [82]. In their work, signals recorded from each muscle were classified based on their quality. This enables them to obtain accurate and repeatable estimates of the sEMG signal's amplitude, spectral properties, and muscle fiber

conduction velocity. Kyeong et al. sought and investigated specific muscles that can provide the most accurate results to recognize the walking environment from lower limb muscles [83]. Finding the best group of muscles that convey the most beneficial information regarding the specific activities performed by the lower limb is highly crucial in our study. The aforementioned empirical results are used to choose the muscles that have the highest quality of the signal, the highest inter-subject uniformity on the location of the innervation zone, and the best capability to provide the most significant features in combination with other muscles. These considerations lead us to choose eight particular muscles from the lower limb of the dominant leg to study, including: Tibialis Anterior (TA), Gastrocnemius Medialis (GM), Gastrocnemius Lateralis (GL), Soleus from the shank and, Biceps Femoris (BF), Vastus Lateralis (VL), Vastus Medialis (VM), and Semitendinosus from the thigh.

An advantage of selecting the various number of muscle groups is to investigate and analyze the effect of different combinations of lower limb muscle groups on detecting the users' intention. Taking these measurements from the optimal set of muscles can help the classifier detect the subject's intention better and find more correlated patterns across the feature set. Therefore, among these eight channels different muscle combinations have been investigated in Section 5.7 to find the smallest (in number) optimally-performing muscle group that provides the best accuracy in detecting the direction intentions. The sensors are positioned on muscles as recommended by Hermens et al. [84] as its available in <http://www.seniam.org>.

Fig. 3.5 shows a sample of the whole setup used with each subject to record the sEMG and IMU data from the human.



(a) Front View

(b) Side View

Figure 3.5: Sample of wearable setup used by a participant

3.3 Signal Pre-processing

Since sEMG signals originate from the muscles, they are inherently affected by various noise sources. These noise components are endemic and avoidable. They can disrupt sEMG signals and cause a misinterpretation of them. This is especially the case when the signal is obtained during dynamic contractions. Beyond using effective methods of locating and securing the sEMG sensor to the skin, one of the most direct means of increasing the integrity of the sEMG signal, is to filter the maximum amount of noise while maintaining as much of the desired EMG signal frequency spectrum as possible [85].

Based on the assumption that the muscles' corresponding voluntary contraction

has a correlation with the force that describes the user's intention, we can find a map between users' intentions and MUAPs that can be captured by sEMG. Each data sample captured by the sEMG sensor on its own is not capable of demonstrating the signal features. Having a window of data samples can ensure a continuous distribution of the information and provide the number of input samples required for any classification task.

3.3.1 Data Segmentation

Data segmentation is one of the most critical parts of the preprocessing procedure for acquiring features of a signal. To ensure a continuous distribution of the information, a segmentation method is used to allow simultaneous feature extraction and, eventually, classification. Two main techniques for segmentation exist that include Overlapping Segmentation (also called "overlapping sliding window") and Disjoint Segmentation (also called "Non-overlapping sliding window") [86].

For limb EMG analysis, most researchers prefer the Overlapping Segmentation method as it enables a more consistent and smoother controller output with minimal delay [87]. It is known that the length of the window represents a compromise between time latency and classification accuracy. The longer the window, the more accurate the prediction would be at the cost of a more delayed controller response. Hudgins et al. described that to reduce the users' perceived lag from the controller in real-time control, the time latency should be kept less than 300 ms [32]. It is worth mentioning that much of this time will be required to obtain enough signal samples to allow feature extraction. Thus, less time is going to be left to perform the actual feature extraction and pattern classification. On the other hand, Smith et al. point out that if classification accuracy drops, it may be beneficial to increase the window length [88].

Menon et al. showed that with window overlaps ranging from 0% to 90%, the mean classification error remains the same across all tests. Whereas the classification error falls as the analysis window becomes longer than that [89]. Based on literature review and empirical tests, a window length of 260 ms, and an overlapping length of 235 ms has been chosen for this study.

3.3.2 Filtering

Most studies incorporate a bandpass Butterworth filter with different orders and cutoff frequencies to filter the EMG signals. The bandpass filtering generally involves a high pass cutoff between 10–20Hz [90]. The filtering process also includes a lowpass filter with cutoffs between 450-500 Hz when the sampling frequency is 1000Hz [91]. These cutoff values have been suggested to remove noise, reduce motion artifacts, and preserve the actual EMG signal emanating from the muscle. This approach is mainly based on the assumption that all of the frequency content remaining, after differential amplification, actually enhances muscle force estimates' validity [92]. Considering this assumption, a bandpass Butterworth filter of order 4 with cutoff frequencies of 20-450Hz is employed.

Chapter 4

Lower Limb Intention Detection Using Classical Machine Learning

Throughout this chapter, C-ML methods, built upon Feature Engineering structure, are applied to detect human intention to move in a particular direction from lower limb muscles. In Section 4.2, Multiple features commonly used in the literature are discussed, and two different combinations of manually selected features are considered for this study, one with and the other without applying a dimensionality reduction technique (four feature sets in total). Ten C-ML models are fundamentally explained in Section 4.3 and evaluated in Section 4.7 using the obtained four feature sets to find the best model and best feature set that can enable the classifiers to differentiate between different patterns and features calculated from each window of sEMG data. Two different implementation algorithms detailed in Section 4.5, (a) Subject-Exclusive learning and (b) Generalized learning over all the participants, are investigated to estimate the robustness of the model to inter-subject scenarios. The results are thoroughly observed and discussed in Section 4.8.

4.1 Introduction

When a human limb repeats a movement, the common assumption is that sEMG signals from the corresponding muscles of that limb in the body (measured in distinct channels) either remain the same, keep similar characteristics or features, or keep specific correlation to the signals or features of other channels in each time window. Thus, the corresponding features of these similar windows are distinguishable from other windows consisting of features corresponding to other gestures/movements. Therefore, sEMG-based human intention detection is a classification problem for the task of detecting user's intentions to move in a particular direction prior to the activity initialization or during the act of moving. Historically, the research on sEMG-based intention detection/gesture recognition has been primarily focused on feature engineering structures that use predefined functions to find discriminators to characterize the sEMG signals. The features calculated through such a structure and the user's initially known intent can be incorporated as inputs and outputs in various classifier's of C-ML approach to find the best model with the highest level of accuracy.

4.2 Feature Engineering

Different types of feature extraction methods have been suggested in the literature for classification, each with evolving mapping and learning methods. The traditional and well-developed methods take advantage of the statistical characteristics of the measurements to identify certain information that relates to the features of interest. Extracted statistical properties that have been used to analyze the sEMG signals in this work are drawn from Time-Domain (TD) and Frequency-Domain (FD).

Time-Domain Features

TD features are computed directly from raw EMG time series, without a need of transformation. Their main advantage is that, as these features can be extracted quickly, the techniques to calculate these features are easily implemented. Therefore, they introduce lower computational complexity and costs to the algorithm [24]. The main concern about using TD features is in detecting the non-stationary properties of EMG signals, as they assume that the signal remains stationary. In stationary signals, the signal's statistical properties (i.e., mean and variance) do not change over time; however, this is not always the case in biological signals. Hence, this category's variation of features might be utilized with caution when the surface EMG signal is recorded through dynamic movements. Moreover, since the computation of time-domain features is based upon signal amplitude, major interference is introduced into the data through recording, particularly for features extracted from energy properties. TD features are widely used for real-time medical and engineering applications due to their classification performances in low noise environments and their lower computational complexity compared to methods using features in the FD and Time-Frequency Domain (TFD) [31].

Frequency-Domain Features

Frequency or spectral features are usually utilized to estimate muscular fatigue, perform MU recruitment analysis, and predict muscle force. The most important analysis tool for FD feature extraction is Power Spectral Density (PSD). The most intuitive description of PSD is that it is defined as the rate at which the MUs are fired within the muscles. Various kinds of statistical properties are attributed to the PSD, which is defined as a Fourier transform of the EMG signal's autocorrelation function. These

properties can be estimated using either Periodogram or parametric methods (i.e. the Auto-Regression (AR) model). Some widely used FD features calculated based on PSD include: Mean Frequency (MNF), Median Frequency (MDF), the 1st, 2nd, 3rd Spectral Moments (SM[1-3]), power spectrum (total power), Peak Frequency (PKF), Mean Power (MNP), Frequency Ratio (FR), Power Spectrum Ratio (PSR), and Variance of Central Frequency (VCF) [31, 86].

Recent studies have shown that FD features result in the less-separable classes for pattern recognition tasks, a disadvantage of these features compared to the TD features. Moreover, FD features result in spectral leakage and high variance due to their computational complexity [86].

4.2.1 Feature Selection and Dimensionality Reduction

Features extracted from TD and FD can be arranged into a multi-dimensional vector space. Among all features, some are redundant, and analyzing them only adds to the computational complexity and cost. Therefore, two common solutions are (i) manually picking the features proven to give the best results for specific classification tasks or (ii) using supervised or unsupervised learning methods to find the most important features. This process is called “Feature Selection” which eventually leaves us with different feature groups or combinations of miscellaneous features from different domains. Implementing this step may help achieve optimal classification results in less time, and it is particularly important for real-time applications [86]. The main algorithm used in this research to automatically reduce the number of features is Principle Component Analysis (PCA). In a nutshell, PCA aims to find the directions of maximum variance in high-dimensional data and projects it onto a new subspace with equal or fewer dimensions than the original one. In this study, it was observed

that on average about 80% of input features have a high variance (meaning they carry distinguishing information) and the rest of the input features have a considerably lower variance (meaning that they might not provide as much information for classification in comparison). Thus, the 80% of input features that contain most of the data information are kept and the remaining 20% are neglected.

4.2.2 Feature Sets

TD features have shown better performance in comparison to FD features for classification tasks with less required computational power causing less delay on real-time applications [31]. Two different combinations of manually selected features that mostly consist of TD features are considered for this study, each with or without applying the dimensionality reduction technique. The most commonly used TD features in the literature are considered to avoid redundancy and ensure that the best performance can be achieved. Further, AR coefficients and Histogram features that can convey FD information of the signals are also added to investigate their effectiveness in each group. TD and FD features that have been used in this study are: Mean Absolute Value (MAV), Zero Crossing (ZC), Waveform Length (WL), Slope Sign Change (SSC), Root Mean Square (RMS), Auto-Regressive coefficients (AR), integrated Mean Absolute Value (iEMG), Histogram (HIST), and Skewness (Skew). In the following section, the main reformulation of these features is presented. Table 4.1 summarizes different combinations of features used as input variables of C-ML models and their abbreviation.

Table 4.1: List of features extracted to be fed to C-ML

List of Manually Selected Features		
Abbreviation	Features	PCA
TD	MAV,SSC,ZC,WL,RMS,AR	No
TD-PCA	MAV,SSC,ZC,WL,RMS,AR	Yes
Enhanced TD	MAV,SSC,ZC,WL,RMS,AR,iEMG,HIST,SKEW	No
Enhanced TD-PCA	MAV,SSC,ZC,WL,RMS,AR,iEMG,HIST,SKEW	Yes

A brief description of the listed features are:

- **Mean Absolute Value (MAV)** This feature depicts the mean of a fully-rectified signal in each window.

$$MAV(x_i) = \frac{1}{L} \sum_{K=1}^L |x_{i,k}| \quad (4.1)$$

where $x_{i,k}$ represents k^{th} sample data in i^{th} segment and L denotes length of the sEMG signal.

- **Slope Sign Changes (SSC)** This feature measures the frequency at which the sign of the signal slope has changed. Given three consecutive samples of $x_{i,K-1}, x_{i,K}, x_{i,K+1}$, the value of SSC is incremented if:

$$(x_{i,K} - x_{i,K-1}) \times (x_{i,K} - x_{i,K+1}) \geq \epsilon \quad (4.2)$$

Where $\epsilon \geq 0$ is employed as a threshold to reduce the impact of noise in this feature.

- **Zero Crossing (ZC)** This feature counts the frequency at which the signal passes through zero. The value of ZC is increased when the following condition is

satisfied:

$$(|x_{i,K} - x_{i,K+1}| \geq \epsilon) \wedge (\text{sgn}(x_{i,K}, x_{i,K+1}) \Leftrightarrow \text{False}) \quad (4.3)$$

Where a threshold $\epsilon \geq 0$ is again added to lessen the effect of noise. $\text{sgn}(a, b)$ is true if either a and b (two real numbers) have the same sign and is false otherwise. It is important to note that if the slope of the signal is small and ϵ is not selected properly, the ZC point might not be detected.

- **Waveform Length (WL)** A feature that shows a simple characterization of the signal's waveform. It can be calculated using following equation:

$$WL(x_i) = \sum_{K=1}^L |x_{i,k} - x_{i,k-1}| \quad (4.4)$$

- **Root Mean Square (RMS)** This feature, also known as the quadratic mean, is closely related to the standard deviation. They are equal when the mean of the signal is zero. RMS is calculated using:

$$RMS(x_i) = \sqrt{\frac{1}{L} \sum_{K=1}^L x_{i,k}^2} \quad (4.5)$$

- **Autoregression Coefficient (AR)** An autoregressive model tries to predict future data, based on a weighted average of the previous data. Such a model characterizes each sample of the signal as a linear combination of the previous sample with an added white noise. The number of coefficients calculated is a trade-off between computational complexity and predictive power. The measure of this trade-off, called AR, is defined as follow:

$$x_{i,k} = \sum_{j=1}^P \rho_j x_{i,k-j} + \epsilon_t \quad (4.6)$$

Where P is the model order, ρ_j is the j th coefficient of the model and ϵ_t is the residual white noise.

- **Integrated EMG (iEMG)** A feature describing the sum of the fully-rectified signal.

$$IEMG(x_i) = \sum_{K=1}^L |x_{i,k}| \quad (4.7)$$

- **EMG Histogram (HIST)** When a muscle is in contraction, the EMG signal deviates from its baseline. The idea behind HIST is to quantify the frequency at which this deviation occurs for various amplitudes. HIST is calculated by determining a symmetric amplitude range centered around the baseline. This range is then separated into n bins of equal length (n is a hyperparameter). The HIST is obtained by counting how often the amplitude of the signal falls within each bin's boundaries.

- **Skewness (SKEW)** is a measure of the asymmetry of the probability distribution of a variable about its mean. The skewness of a distribution is defined as:

$$S = \frac{E(x - \mu)^3}{\sigma^3} \quad (4.8)$$

where μ is the mean of x , σ is the standard deviation of x , and $E(t)$ represents the expected value of the quantity t .

4.3 Classic Machine Learning (C-ML) Fundamentals

A classification procedure identifies various categories of feature information to attribute to a corresponding action in any active system. After a group of features is extracted, discrete classes are derived from them, resulting in the actuation. In the process of pattern recognition, classification is performed by either linear or nonlinear classification algorithm schemes. Linear classifiers are more often suited for real-time control systems due to their lower computation cost. However, linear classifiers underperform when dealing with complex systems. Such schemes often fail to integrate the interactive and nonlinear behavior of muscle systems. The controller logics designed using nonlinear classifiers are readily capable of extracting information about nonlinear behavior from EMG data and follow the desired output. They are also prevalently applied in the multi-modal analysis required for upper limb control systems. On the downside, the nonlinear analysis leads to higher processing time and requires repeated training to avoid over-fitting.

4.3.1 Primary Models

K Nearest Neighborhood

K nearest neighbors (KNN) is an instance-based and nonparametric learning method which classifies cases based on the majority of the votes of its neighbors. A case being the state that the trained logic is attributing to the system at that instance. In this method, for an instance in time, the case is assigned to the class that is most commonly identified among the K nearest neighbor instances of identification

generated about it, measured by a distance function [93].

Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is based upon the concept of searching for a linear combination of features (predictors) that best separates classes (targets). The classes are separated at separation (or decision) boundaries, the shape of which depend on the distribution of the covariance matrices of the features. This classifier uses a score function to capture separability. Given the score function, the problem is to estimate the linear coefficients that maximize the score. LDA assumes that feature covariance matrices of all classes are the same, which results in a linear decision boundary. LDA's superiority over other C-ML classification algorithms partially stems from its fast computation [93]. Many studies have been made on the comparison of LDA to nonlinear classification tools, although it might have a lower recognition rate than other techniques.

Quadratic Discriminant Analysis

Additional changes have been built on the base concept of LDA in order to improve upon its pattern recognition capability. One example is Quadratic Discriminant Analysis (QDA). It is a general discriminant function with quadratic decision boundaries (nonlinear separation boundary) which can be used to classify datasets with two or more classes. QDA needs to estimate the covariance matrix for each classes to operate [93].

SVM

A Support Vector Machine (SVM) performs classification by finding a hyperplane that maximizes the margin between all classes. The vectors that define the hyperplane are called the support vectors. The simplest way to separate two groups of data is with a straight line (1D), flat plane (2D), or an N-dimensional hyperplane. However, there are situations where a nonlinear region can separate the groups more efficiently. SVM handles this by treating the nonlinear region as the image of a mapping from a kernel function (nonlinear) on an affine subspace of a higher-dimensional space onto a different space (containing the nonlinear region) where a hyperplane (linear) cannot be used to do the separation. It means a linear learning machine learns a nonlinear function in a high-dimensional feature space. In literature, this is called the kernel trick, which means the kernel function transforms the data into a higher dimensional feature space to make it possible to perform the linear separation [94].

Most often, the results of SVM outperform other linear classifiers. Although, this is not the case in comparison to nonlinear classifiers. Comparative studies of SVM with nonlinear classifiers do not provide decisive conclusions. However, SVM has generally failed to achieve better classification accuracy in comparison to complex neural networks.

Decision Tree

Decision trees (DTs) are built using recursive partitioning to classify the data in the form of a tree structure. It is built top-down from a single root node. Then, at each step, the data features are partitioned into subsets that contain instances with similar classes (homogeneous). The algorithm chooses the attribute most capable

of predicting in order to split the data. This is done by calculating the amount of randomness (entropy) within each node and then calculating the decrease of entropy (information gain) after a dataset is split based on that attribute. Entropy is a measure of the homogeneity of the samples in each node. Information gain is a measure of the ability of each attribute to decrease the entropy after splitting. At each level, the attribute that results in the highest information gain will be chosen to advance to further branches in the tree. Constructing a DT is all about finding the feature that returns the highest information gain (i.e., the most homogeneous branches). This algorithm employs a greedy search approach through the space of possible branches with no backtracking.

The main advantages of the DTs are: (i) ease of interpretation, (ii) high accuracy even with little hard data, and (iii) the capability to help determine the worst and best attributes along with an expected value estimate for different attributes. The main disadvantages of DTs are: They are prone to being over-fit. Therefore, they are unstable when facing large variance, meaning that a small fluctuation in the data can lead to a large change in the structure of the optimal DT. Thus, they can be relatively inaccurate when used in real-time applications in comparison to other classifiers that use a similar data pool. This can be remedied by replacing a single DT with a random forest of DTs, which is described in the following section.

Gaussian Naive Bayes

Naive Bayes is a probabilistic machine learning model that uses a classification technique based on Bayes' Theorem with an added assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. Gaussian

Naive Bayes (GNB) is a specific type of Naive Bayes which is built on the assumption of a normal distribution of probabilities.

The classification rate of Naive Bayes is usually lower in comparison to other linear and nonlinear classifiers for EMG classification. Thus, its applications are also limited. However, considering its lower processing time, its performance has been investigated in real-time applications. As an example, Al-Timemy et al. used Naive Bayes as one of the classifiers to investigate control of an EMG-based system against different levels of forces [95].

4.3.2 Ensemble Models

Ensemble technique utilizes multiple independent models/weak learners which can be similar or different in order to derive an output or make predictions. These models aim to improve generalizability/robustness over a single classifier.

Two families of ensemble models are: (i) Bootstrap aggregating method “Bagging”, and (ii) Boosting method. The general idea behind a Bagging procedure is to reduce the variance for those algorithms that are less capable of handling high variance (e.g., DT). Ensemble Bag and Random Forest are samples of Bagging-based ensemble models. Adaptive Boosting (AdaBoost) and Gradient Boosted Decision Trees (GBDT) are samples of the boosting family.

Bagging-Based Ensembles: Ensemble Bag and Random Forest

In these types of ensemble models, first, subsets of the original training set are generated. Then bagging methods form a class of algorithms that build several instances

of a classifier on each generated subset and aggregate their individual predictions to form a final prediction. In an “Ensemble Bag” classifier, the base classifiers are allowed to look through all input variables (features) of a training set to detect the most optimal feature to split the algorithm [96]. On the other hand, when the classifier is limited to choosing a random sample of features, the algorithm is called Random Forest (RF) [97]. DT is most commonly used as the base classifier for ensemble models. This is due to the high-variance nature of this model. Ensemble Bagging (Bag) method work best with strong and complex base models (e.g., fully developed DTs).

RF classifier has generally shown promising results for pattern recognition tasks using EMG signals. It is capable of distinguishing between significantly similar features and detect at a higher classification rate in comparison to SVM and NNs. This algorithm’s efficient processing time is considered as another advantage over other classifiers. Liarakapis et al. applied random forest to design EMG-based task-specific control models in their reach-to-grasp studies [98]. Song et al. applied random forest and grid search random forest to predict human motion intention. They showed that the algorithm could achieve a high level of accuracy for pattern recognition for five different motions [99].

Boosting-Based Ensembles: Adaptive Boosting and Gradient Boosted Decision Trees

In contrast to Bagging-based Ensembles where the base classifiers are built and combined in a parallel configuration, base classifiers of Boosting methods are built sequentially. The algorithm first starts out with a base classifier that is prepared on the training data. A second classifier is then created following the first to focus on the instances in the training data that the first classifier detected wrong. Each subsequent

classifier attempts to correct the wrong predictions made by the classifier before it in the sequence. The process continues to add classifiers until either a desired accuracy is met or a limit on the number of models is reached. In general, boosting methods are designed to work best with weak base models (e.g., shallow DTs). The difference between AdaBoost and GBDT is in the method they use for correcting the error in the predictions. In AdaBoost, this end is achieved by finding those examples in the training dataset that were misclassified in each tree, and adding more weight to those examples in further training. This way the newly generated tree is trained on the data which has more focus on the previously misclassified samples [100]. GBDT is a generalization of boosting to arbitrary differentiable loss functions. Models are fit using any arbitrary differentiable loss function and a gradient descent optimization algorithm [101].

4.4 Hyperparameter Tuning

Hyperparameter tuning is a crucial part of the model preparation step as it can control the overall behaviour of each of the machine learning model. Each C-ML model has different hyperparameters that are need to be set during the training and validation process [102]. In order to empower all the models equally to explore their capabilities to the best, instead of enforcing hyperparameters for each model, a grid-search toolbox [103] is utilized that applies a global search to find the optimal hyperparameters that provide highest accuracy on average over all validation sets. Then, a custom-written fine gridsearch code further optimizes the hyperparameters in the vicinity of the initial results of the global grid search. The optimal hyperparameters, different for each scenario, implementation algorithm, chose of validation set, etc., were eventually utilized in the training to prepare their respective final model. The parameters

that are tuned for each model are listed in Table 4.2.

Table 4.2: Selected range of hyperparameters for each of the C-ML Models

Model	Hyper-Parameters	Investigated range/types
KNN	n-neighbors	in range of (2, 20) with step of 1
LDA	solver	['svd', 'lsqr']
QDA	reg-param	in range of (0.0, 0.5) with step of 0.1
SVM	kernel	['linear', 'rbf']
	C	[0.01, 0.1, 1, 10]
DT	criterion	['gini', 'entropy']
	max-depth	in range of (10, 100) with step of 1
GNB	-	-
RF	criterion	['gini', 'entropy']
	n-estimators	in range of (10, 150) with step of 5
	max-depth	in range of (10, 150) with step of 5
Bag	n-estimators	in range of (5, 100) with step of 5
	max-features	[0.3, 0.5, 1]
	max-samples	[0.3, 0.5, 1]
Ada	n-estimators	in range of (5, 100) with step of 5
	learning-rate	[0.01, 0.1, 1]
GBDT	n-estimators	in range of (5, 100) with step of 5
	max-depth	in range of (1, 50) with step of 1
	learning-rate	[0.01, 0.1, 1]

4.5 Different Implementation Algorithms

One of the objectives of this thesis is to investigate the possibility of generalizing the classifier and making it robust in its prediction against the choice of the subject it is trained on. Therefore, two different implementation algorithms are designed. Both singling out the same one cycle for eventual testing (of five cycles collected, one set aside for eventual testing), but they train their models differently as described below.

- **Generalized Algorithm:** Out of the four remaining cycles, three are concatenated to be used as a training set. Each classifier is trained on this training set. The fourth cycle is set aside as a validation set. The classifier is optimized through further training on the validation set. In order to avoid having the choice of validation set affect the outcome too much, an iterative loop is used. Each loop uses a different cycle as the validation set. The reported accuracy is averaged over the performance of these iterations.
- **Subject-Exclusive Algorithm:** Each classifier is trained and validated per each participant separately in the same manner described in the previous section. The accuracy of the trained model is then evaluated on the test set of that same participant. The overall performance of the model is then calculated by averaging these individual accuracies. This implementation is “Subject-Exclusive” since the learning is not generalized over all the participants. It can represent how the model works for each participant.

4.6 Intention Detection Evaluation Metrics

There are many different metrics that are being used as indicators to assess and evaluate the model's performance. The number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) are used in evaluating the performance of a classifier. Several metrics can be used to interpret these numbers to evaluate the performance of the classifier. These include: Precision, Recall, F1-Score, Jaccard Score, and Accuracy Score. The Accuracy Score is the most used metric for classification problems. It is particularly convenient when there is an equal number of observations in each class (which is the case in this research) and when all predictions and prediction errors are equally important (which also is the case in our research). In multi-label classification, Accuracy Score computes the subset accuracy: where the set of labels predicted for a sample (\hat{y}_i) must exactly match the corresponding set of labels in y_true (y_i). The formula used in Scikit-learn library [103,104], to calculate a subset accuracy can be described as:

$$Accuracy(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} 1(\hat{y}_i = y_i) \quad (4.9)$$

where $1(x)$ is the indicator function that works in form of:

$$1(x) = \begin{cases} 1 & \hat{y}_i = y_i \\ 0 & \hat{y}_i \neq y_i \end{cases} \quad (4.10)$$

Another way of assessing the performance of a classifier is to pay attention to its ability to distinguish between various labels by using a confusion matrix. The confusion matrix is a handy presentation of the accuracy of a model with two or

more classes. The table presents correct expected predictions on one of the axes and the classifier’s prediction outcomes on the other axis. The values of the diagonal elements represent the degree of correctly predicted classes (normalized among zero to one, where one indicates 100% true prediction of the corresponding label).

4.7 Results

Data was collected, pre-processed, and organized as detailed in Chapter 3. Based on that, various features are computed in the form of four different feature sets described in Section 4.2.2. Then, the ten explained C-ML models with two different implementation strategies are applied on each computed feature set. Figs. 4.1, 4.2, 4.3, and 4.4. depict an analysis of the application of different feature sets in each scenario found by Generalized algorithm, compared in terms of: (i) Classifier, (ii) whether the classifier is utilizing PCA or not, and (iii) experiment scenarios.

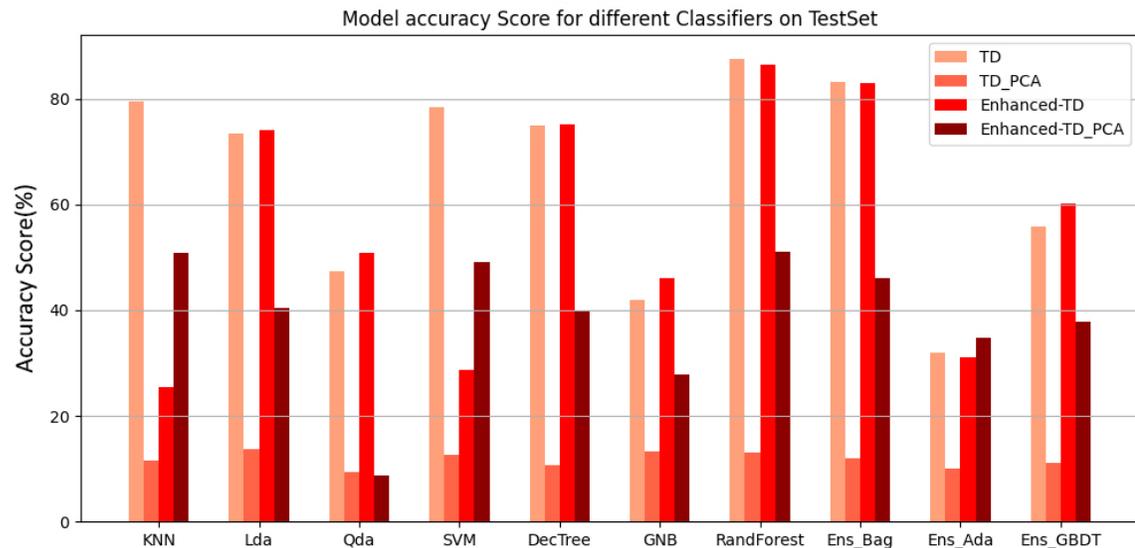


Figure 4.1: Each Generalized classifiers’ performance when different set of features are selected in Sitting Position Scenario.

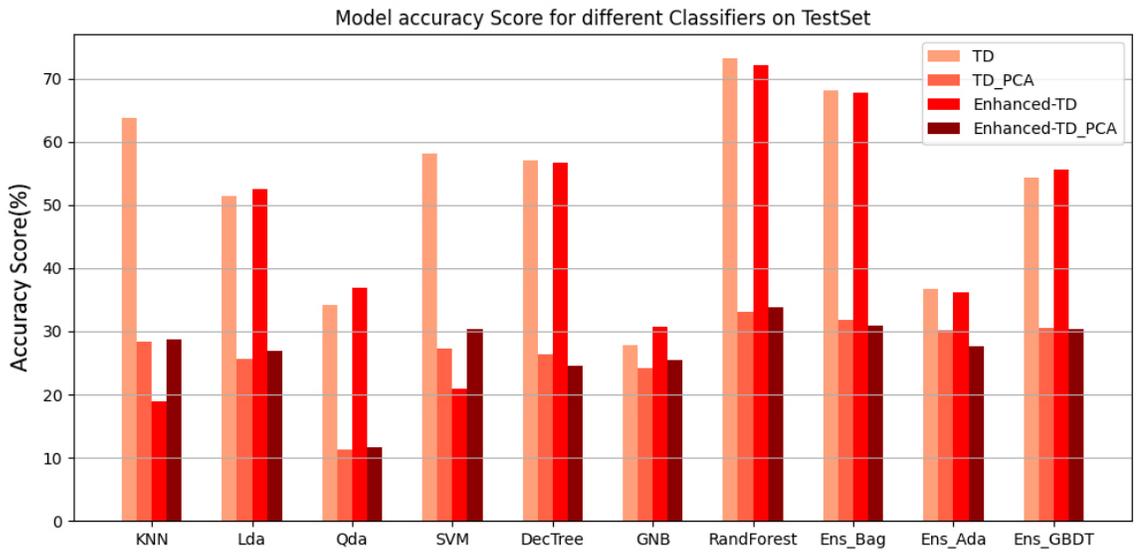


Figure 4.2: Each Generalized classifiers' performance when different set of features are selected in Standing Position Scenario.

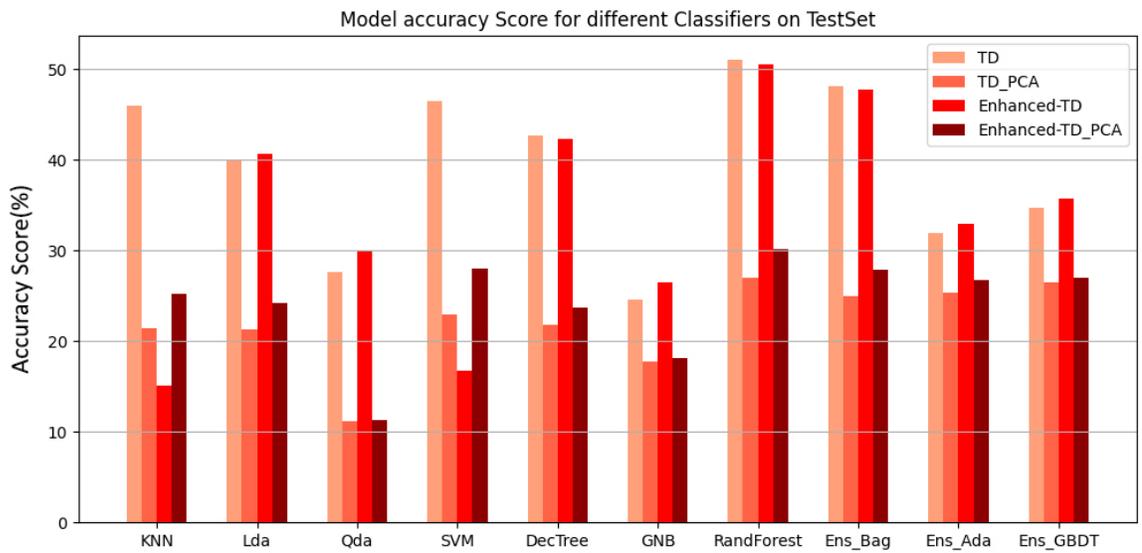


Figure 4.3: Each Generalized classifiers' performance when different set of features are selected in Attempted Motion Scenario.

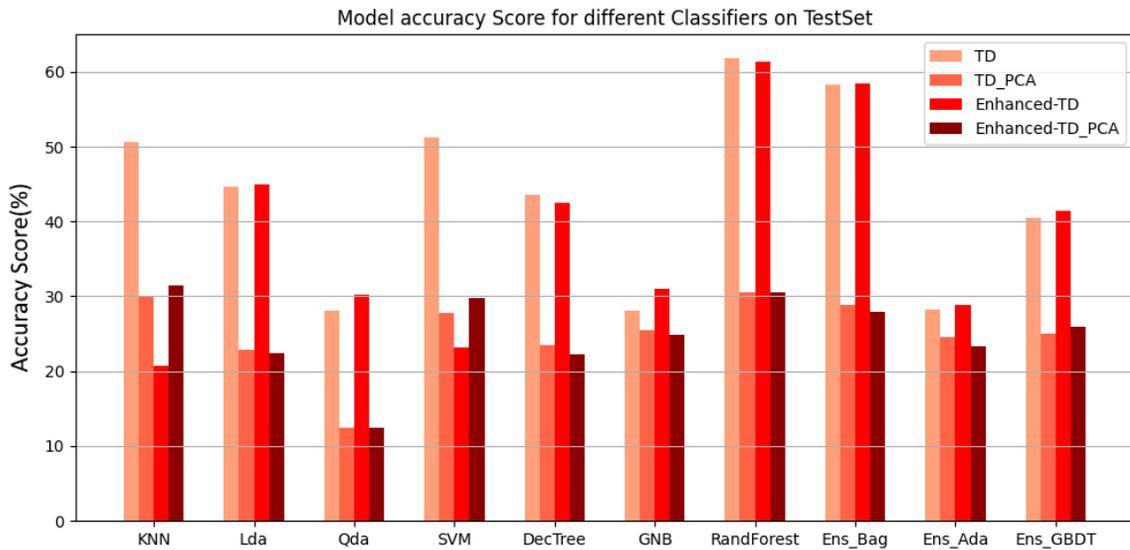


Figure 4.4: Each Generalized classifiers' performance when different set of features are selected in Dynamic Motion Scenario.

Tables. 4.3, 4.4, 4.5, and 4.6 depict a comparison of two different algorithms that are used to investigate the generalization capabilities of each model. The TD feature set is used to carry out this analysis because of its superior performance as can be seen in Figs. 4.1, 4.2, 4.3, and 4.4.

4.8 Discussion

Based on the results of Figs. 4.1, 4.2, 4.3, and 4.4 best-case scenario in terms of detection accuracy happens when TD feature set (10 features per channel) is chosen as the source of intention detection information. PCA worsens the classifiers' performance in all cases. This is not unexpected because PCA is an unsupervised feature reduction method. PCA reduces the feature dimensionality by selecting a new feature space based on the eigen vectors corresponding to the largest eigenvalues. Essentially, this

Table 4.3: C-ML Models' Accuracy Score evaluated on test set in Sitting Position scenario with TD features

	Generalized(%)	Subject-Exclusive(%)
	mean \pm <i>std</i>	mean \pm <i>std</i>
KNN	79.43 \pm 0.71	83.63 \pm 0.96
LDA	73.34 \pm 0.59	91.67 \pm 0.84
QDA	47.4 \pm 1.5	82.98 \pm 0.81
SVM	78.43 \pm 1.02	85.51 \pm 0.55
DT	74.92 \pm 1.78	79.94 \pm 0.88
GNB	42.03 \pm 1.01	82.1 \pm 1.19
Ens-RF	87.63 \pm 0.65	87.2 \pm 0.85
Ens-Bag	83.12 \pm 1.12	83.81 \pm 1.89
Ens-Ada	31.98 \pm 12.57	27.46 \pm 4.07
Ens-GBDT	55.76 \pm 6.08	68.64 \pm 5.11

Table 4.4: C-ML Models' Accuracy Score evaluated on test set in Standing Position scenario with TD features

	Generalized(%)	Subject-Exclusive(%)
	mean \pm <i>std</i>	mean \pm <i>std</i>
KNN	63.72 \pm 1.09	67.06 \pm 1.04
LDA	51.48 \pm 0.85	75 \pm 1.12
QDA	34.17 \pm 3.06	64.91 \pm 1.28
SVM	58.08 \pm 0.84	68.29 \pm 1.54
DT	57.14 \pm 1.24	63.08 \pm 2.53
GNB	27.74 \pm 0.7	62.84 \pm 1.49
Ens-RF	73.32 \pm 1.27	73.38 \pm 1.47
Ens-Bag	68.1 \pm 2.18	68.35 \pm 1.64
Ens-Ada	36.69 \pm 1.87	29.1 \pm 3.17
Ens-GBDT	54.28 \pm 1.36	60.26 \pm 2.48

Table 4.5: C-ML Models' Accuracy Score evaluated on test set in Attempted Motion scenario with TD features

	Generalized(%)	Subject-Exclusive(%)
	mean \pm <i>std</i>	mean \pm <i>std</i>
KNN	46.04 \pm 0.8	48.90 \pm 0.94
LDA	39.87 \pm 0.9	53.1 \pm 1.32
QDA	27.67 \pm 0.75	49.56 \pm 0.97
SVM	46.49 \pm 0.4	53.43 \pm 1.6
DT	42.73 \pm 1.22	47.13 \pm 0.68
GNB	24.65 \pm 1.08	50.61 \pm 1.31
Ens-RF	51.13 \pm 1.18	53.81 \pm 1.32
Ens-Bag	48.15 \pm 0.46	51.52 \pm 0.93
Ens-Ada	31.88 \pm 1.30	31.1 \pm 1.29
Ens-GBDT	34.79 \pm 1.53	45.52 \pm 2.79

Table 4.6: C-ML Models' Accuracy Score evaluated on test set in Dynamic Motion scenario with TD features

	Generalized(%)	Subject-Exclusive(%)
	mean \pm <i>std</i>	mean \pm <i>std</i>
KNN	50.59 \pm 0.83	55.06 \pm 0.7
LDA	44.73 \pm 0.08	61.82 \pm 0.84
QDA	28.06 \pm 1.02	45.54 \pm 1.16
SVM	51.17 \pm 0.23	58.9 \pm 0.54
DT	43.62 \pm 0.63	51.12 \pm 0.77
GNB	28.02 \pm 0.57	39.62 \pm 0.93
Ens-RF	61.89 \pm 0.43	64.8 \pm 0.37
Ens-Bag	58.34 \pm 0.73	59.76 \pm 0.97
Ens-Ada	28.19 \pm 1.34	25.38 \pm 2.13
Ens-GBDT	40.58 \pm 2.72	49.86 \pm 0.52

keeps the features with the highest variance. These parts often contain the “most important” information for discrimination; however, this is not necessarily helpful for all classifiers because: (i) by removing some less important features, a portion of the information that the classifier can utilize is thrown away, and (ii) probable correlations that exist between the remaining features and the discarded ones might be lost.

In all of the scenarios, most of the classifiers trained using the Subject-Exclusive method surpassed the trained classifiers using the Generalized method. This is particularly true for the classifiers with high prediction accuracy, representing how well they can distinguish between different gestures. This was expected since, in the Subject-Exclusive methodology, the training set and the test set are collected from the same participant. This method’s more desirable prediction performance can be attributed to its focus on a specific subject. On the contrary, this requires that each subject is tested several times before the algorithm is able to make predictions about them. The algorithm lacks vision in distinguishing between factors related to different participants.

In almost all cases, the Boosting-based classifiers and QDA classifier have exhibited the least reliable performance. The former classifiers are trying to focus on samples that are incorrectly classified. Due to these classifiers’ shallow depth, the incorrectly classified features that appear with a high variance cause them to mispredict. The QDA classifier shows less accuracy because, with quadratic boundaries, it can easily overfit the linearly distributed features, resulting in lower accuracy in the test set when it has to analyse new behavior.

In the **Sitting Position Scenario** with 11 distinct gestures; The LDA (91.67%)

and the RandomForest (87.63%) approaches achieve the highest accuracy of prediction among Subject-Exclusive and Generalized methods, respectively. Due to the highest accuracy achieved in this scenario, lower limb intention detection via sEMG is proven to be reliably capable of facilitating joystick-type controllers. Furthermore, in everyday activities, not all specific foot and toe gestures are made frequently. Therefore, by additional training on the gestures that the algorithm looks for and providing online feedback to the subject, their ability to control a device using their leg motion can be enhanced.

In the **Standing Position Scenario** with 9 distinct gestures; LDA (75%) and RandomForest (73.32%) approaches achieve the highest accuracy of prediction among Subject-Exclusive and Generalized methods, respectively. The number of gestures to be identified in this scenario is lower in comparison. Still, relative to the Sitting Position scenario, the classifiers were not as highly capable of distinguishing between different types of gestures. The cause of this may be rooted back to the muscles used to record sEMG data. The most involved muscles in making gestures in the Sitting Position Scenario are mostly the ones that are being monitored in this experiment. On the other hand, in the Standing Position scenario, many gestures originate from the hip muscles. Due to the extra burden imposed on participants to attach hip muscles, these muscles are not monitored during the data collection phase of these experiments. Lack of hip muscle motion information leaves the classifier weaker in distinguishing the tasks that are highly dependant on it. One can increase the prediction accuracy by including hip muscles' data, adding data from other sensors such as IMUs on the leg or Force plate underneath the feet, and fusing the data collected from these sensors with the sEMG data.

In the **Attempted Motion Scenario** with 9 distinct gestures; Random Forest achieved the highest accuracy of prediction among Subject-Exclusive methods with

53.81% and Generalized methods with 51.13% respectively. Similar to the Standing Position scenario, the lack of hip data can result in lower prediction accuracy compared to the Sitting Position scenario. Analyzing each classifier's results when trained in each mode shows us that there is a noticeable difference among their ability in differentiating between various gestures. The cause can be sought out by taking a close look at the experiment itself. During the experiment, the participants are asked to imagine and provide the muscle tension required as if they were supposed to move in a specific direction without actually performing the task. Each person has a distinct manner of providing such an intention while abstaining from motion. Thus, an explicit difference in the sEMG signal between different subjects is expected. Although, this is the case that most resembles the behavior of a participant who experiences some level of disability. Thus, in such a case, Subject-Exclusive methodology is most advantageous to enhance the accuracy level of the detector.

In the **Dynamic Motion** with 9 distinct activities; RandomForest surpassed all other classifiers either trained with a Subject-Exclusive method or with Generalized method with prediction accuracy of 64.8% and 61.89%, respectively. Low prediction accuracy was expected in this scenario due to the activities' more complex dynamic nature. The lack of hip data, as discussed previously, is also a factor. The inherent dynamic complexities make it difficult for classifiers to relate different windows of data. Therefore, they ignore the correlation between windows over time. This problem cannot be easily solved using C-ML algorithms. DL methods described in Chapter 5, are investigated to overcome this shortcoming of C-ML methodology in detecting dynamic-based scenarios.

In General, Ensemble_RF, LDA, Ensemble_Bag, SVM, and DT, respectively, showed better performance across all the scenarios. Nevertheless, Random Forest has been the best classifier for almost all of our scenarios. Standardizing data played

a significant role in increasing the SVM and LDA classifiers' performance, although its effect on other classifiers was negligible. Standard Deviation (std), calculated by averaging each models' prediction accuracy on all subjects, gives a measure of the range of variation or dispersion of prediction accuracy. Relatively low values of std that are observable in most of the machine learning models indicate that the prediction accuracy values tend to be close to the mean of prediction accuracy. Especially, since the std values are relatively smaller than the differences between accuracies of the different models, particularly for the models with higher accuracy (RF and LDA), the superiority of RF and LDA over other models is granted.

4.8.1 Confusion Matrix

Confusion Matrices contain information that can help to interpret the classifiers' performance and the reason behind their shortcoming in identifying the incorrect classes with more intuitive insight. The diagonal elements of this matrix represent the number of times for which the predicted class is equal to the true class, while off-diagonal elements are those that are mislabeled by the classifier. The elements of the matrix can be normalized with respect to the total number of samples of each class. This results in having elements between 0 and 1. The closer the diagonal values of the confusion matrix to 1, the better it indicates many correct predictions in that class. For example, as shown in Fig. 4.5, in the Sitting Position Scenario, the least accurate prediction happens in the case of Knee Extension where it is mostly mistaken for Knee Abduction gesture. This can be the result of the close correlation between these gestures. Both gestures are capturing the motion of extending the leg either in lateral direction or perpendicular to the participants' body. The muscles that are involved in performing these gestures are quite adjacent to each other. They are all working to overcome gravity to some degree. Moreover, lack of data from hip muscle

can lead the model to confuse these two gestures during classification. In addition, performing Knee Abduction and holding the ankle in a near 90° position with respect to the user's thigh is not a comfortable situation. Therefore, many of the users held their shank within a lower range of angles and instead pushed their shank forward. This might be another reason behind the classifier seeing a similar pattern between these two gestures.

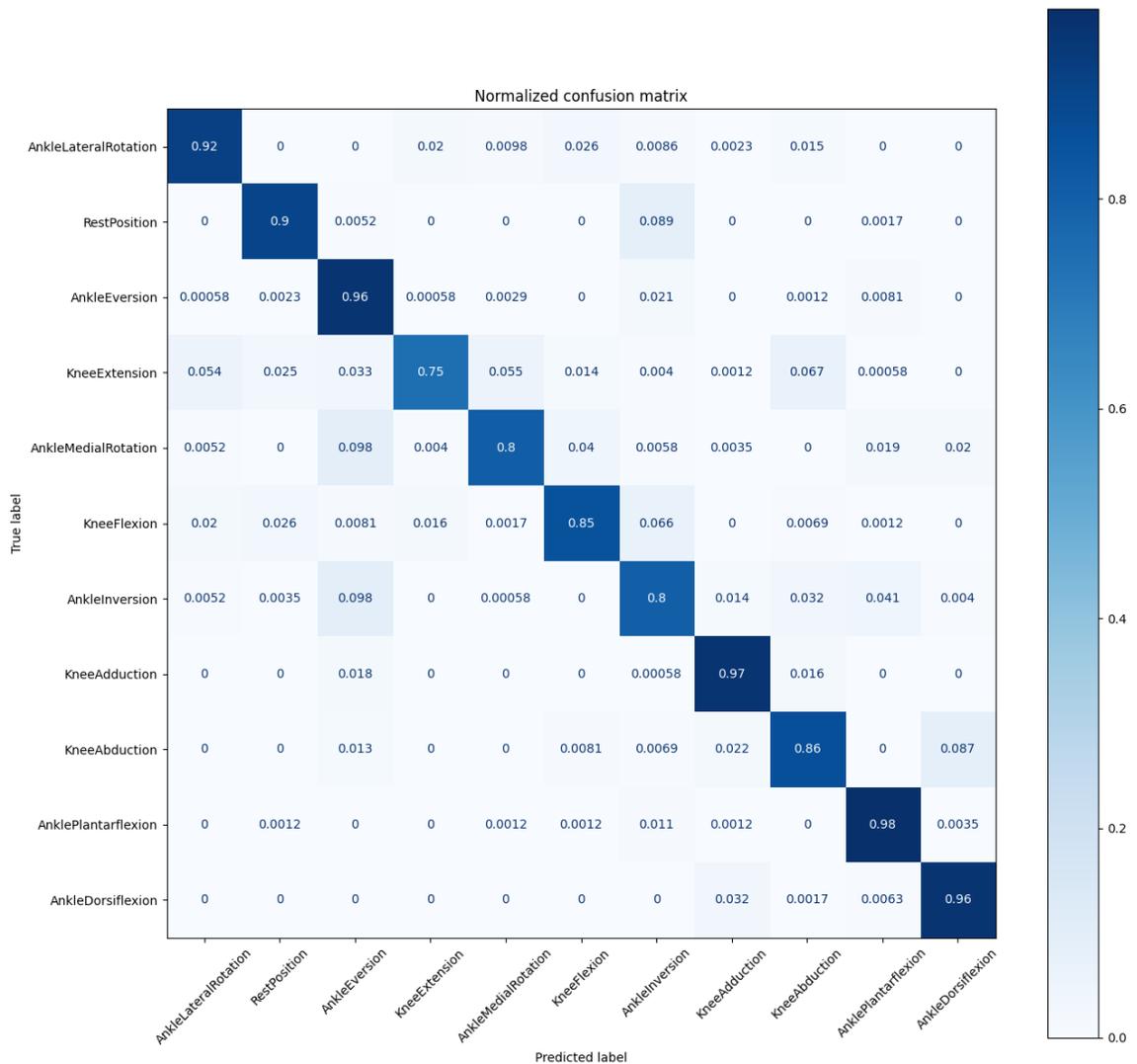


Figure 4.5: Confusion Matrix of Ens-RF Classifier based on TD feature set in Sitting Position scenario implemented based on Generalized algorithm.

As it is shown in Fig. 4.6, in the Standing Position Scenario, when the participant is performing the first stage of Walk Forward, the classifiers are least capable of identifying their act correctly. The detection algorithm confuses this gesture frequently with the Hip Internal Rotation and Cross Step gestures. By taking a closer look at these gestures one can conclude that external muscles are more engaged and extended in these gestures and thus the model is more susceptible to confuse these gestures with each other and make wrong predictions.

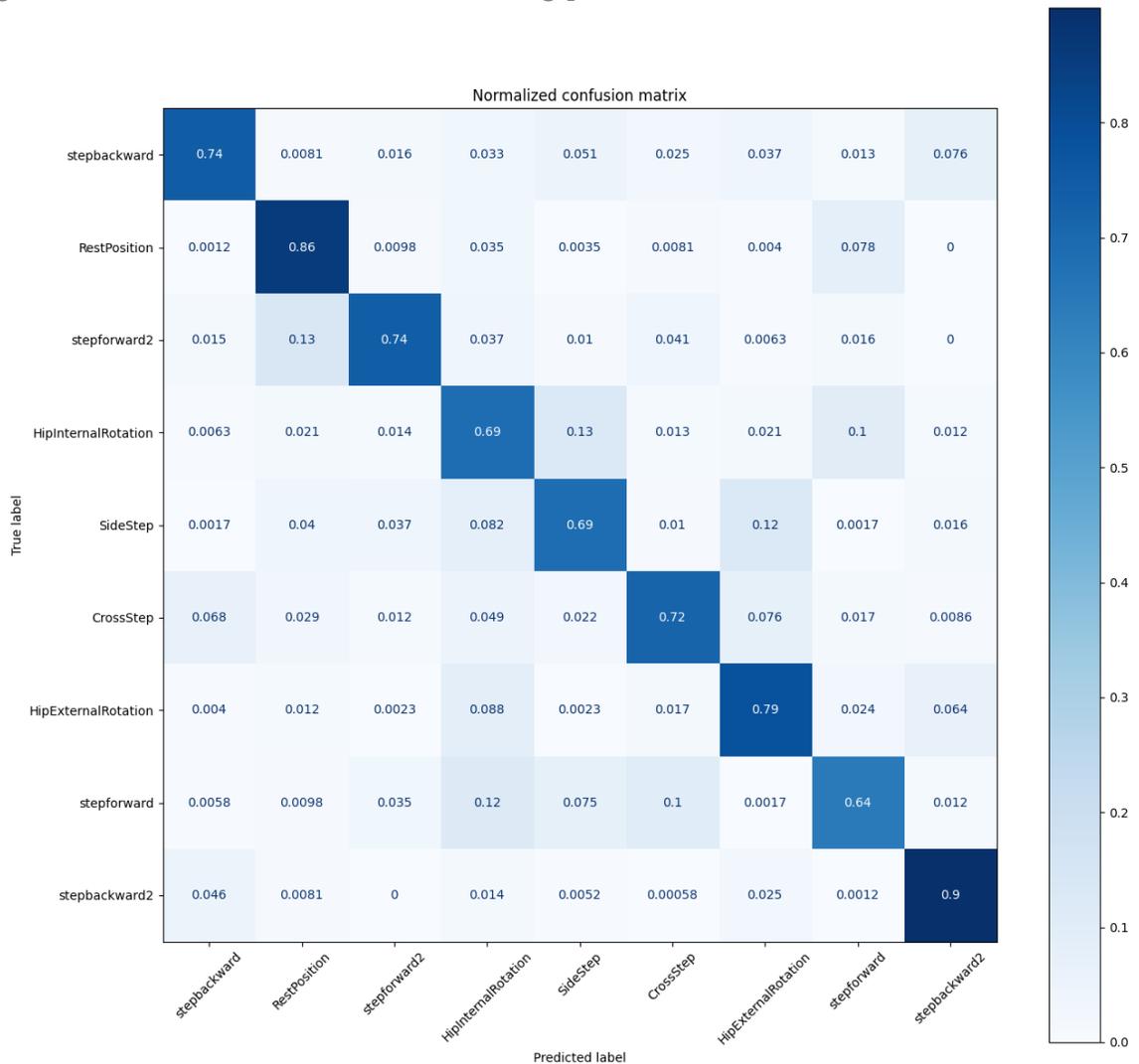


Figure 4.6: Confusion Matrix of Ens-RF Classifier based on TD feature set in Standing Position scenario implemented based on Generalized algorithm.

As it is shown in Fig. 4.7, in the Attempted Motion Scenario, the prediction algorithm faces the most difficulty when the participant tries to Walk Backward. This gesture gets confused with Walk Left Side, Walk Right Side, and Sit gestures. The common feature between these gestures is that the distinguishing characteristics of the gesture completely rely on hip muscles. This is because all these gestures originate directly from the hip, while other muscles have relatively less effect in the prediction. The lack of hip muscle data influences this scenario the most because of the complexity of the gestures and their reliance on hip muscles. Moreover, the manner each subject use to abstain themselves from the motion might interfere with the true muscle activities and interfere with prediction, particularly when trained with Generalized method.

As it is shown in Fig. 4.8, in Dynamic Motion Scenario, the predictor model performs least accurate among the movement related activities. The confusion in the results can be rooted back to the dynamic nature of this scenario. In each sample window, the model might not observe the complete activity and therefore, some of windows might become controversial to other windows (especially for the activities that their cycle takes more time to complete (e.g. Walk Backward that subjects are not used to perform regularly)). Further, due to the inherent structure of C-ML models, they cannot relate the patterns of each window to other adjacent windows.

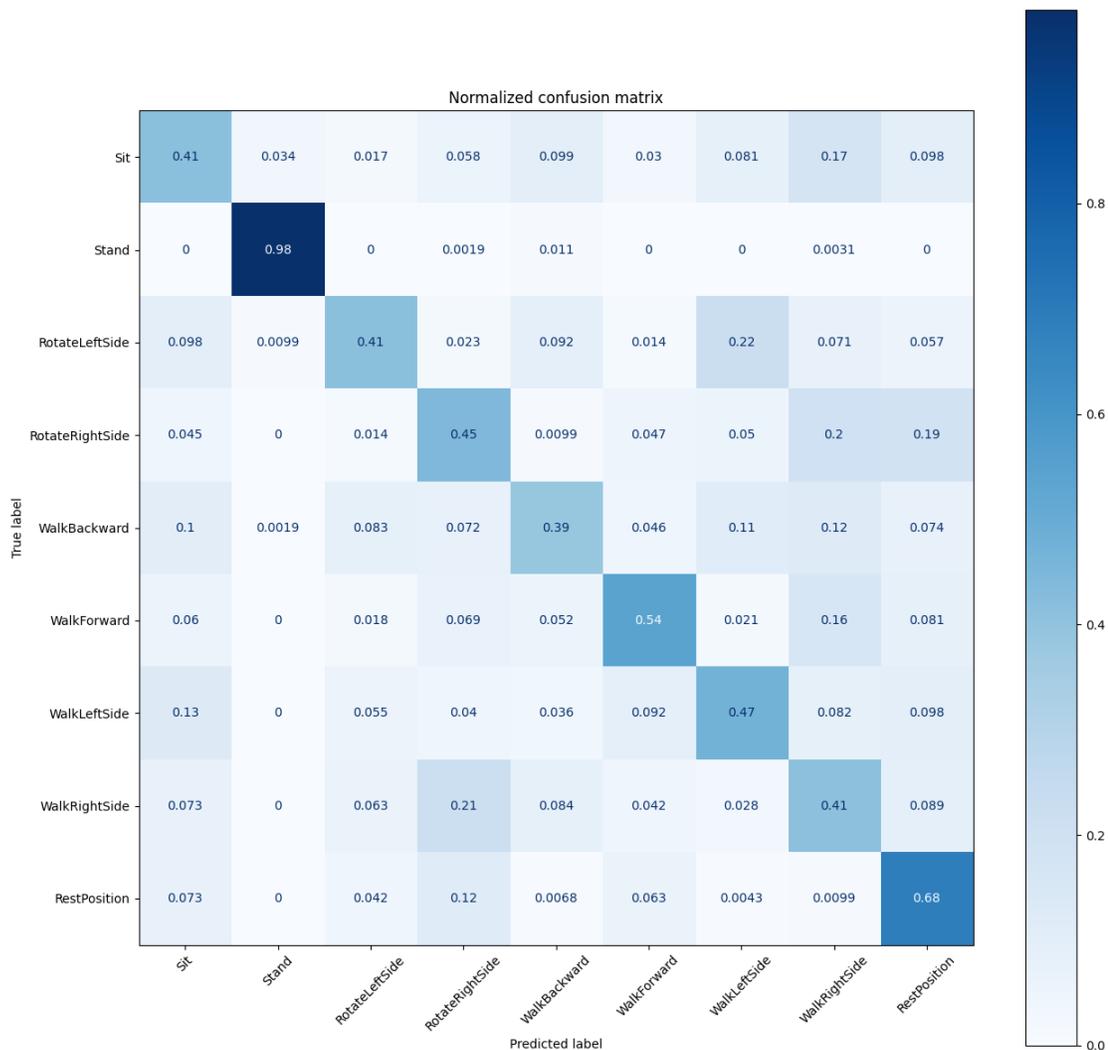


Figure 4.7: Confusion Matrix of Ens-RF Classifier based on TD feature set in Attempted Motion scenario implemented based on Generalized algorithm.

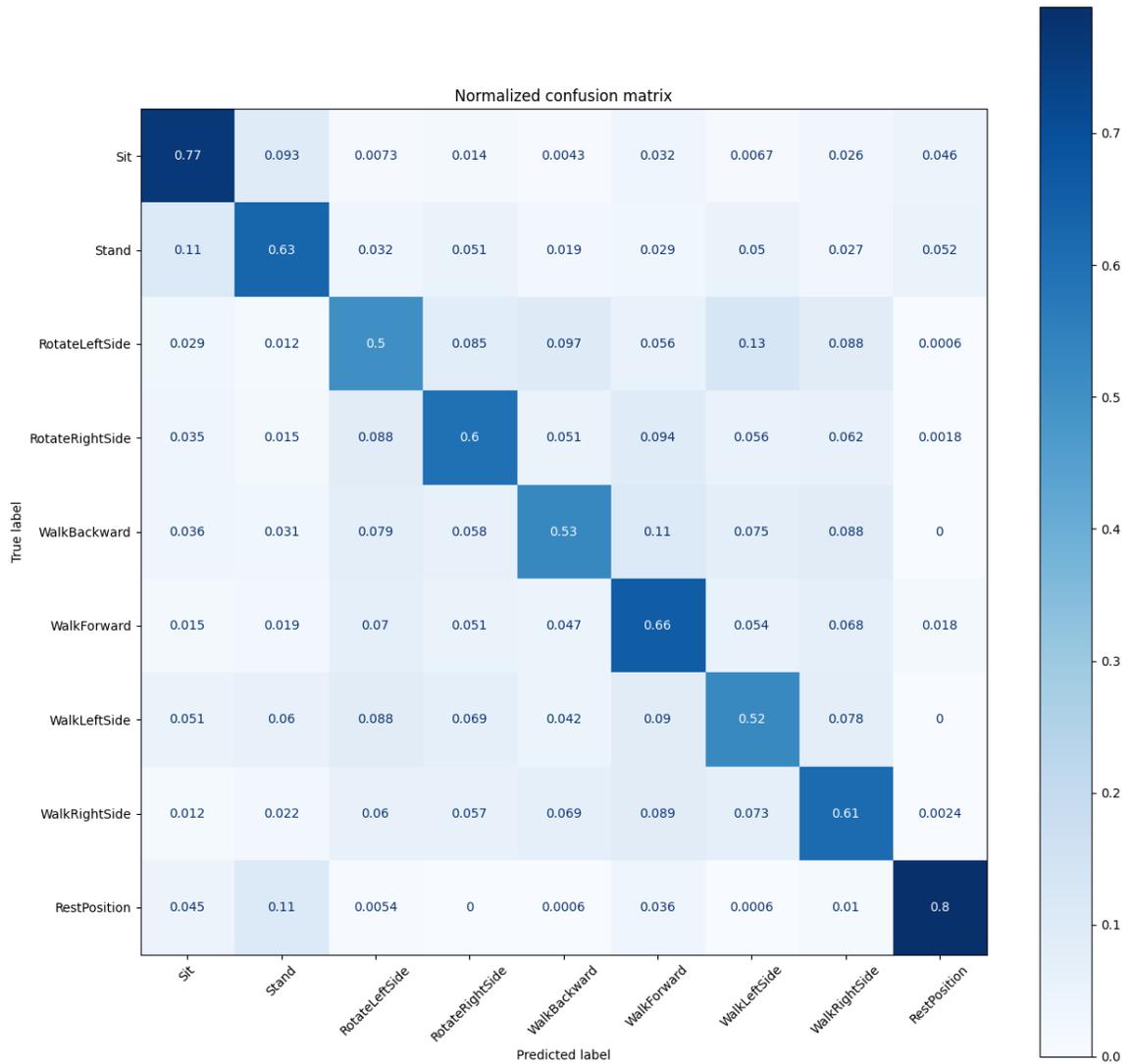


Figure 4.8: Confusion Matrix of Ens-RF Classifier based on TD feature set in Dynamic Motion scenario implemented based on Generalized algorithm.

4.9 Chapter Summary

Overall, the TD feature set showed the best performance in differentiating between tasks for all scenarios. The PCA dimensionality reduction technique though reducing the training time, also causes a noticeable drop of accuracy in our analysis. The Subject-Exclusive method was almost always the best implementation method, facing occasional competition from the Generalized method. Subject-Exclusive method was identified as the most beneficial method to be utilized when the gestures/activities are complex and subjects have distinct manner for performing them. Moreover, due to the bigger data set used in the Generalized algorithm, the training time required was much more than the Subject-Exclusive method. Therefore, applying this methodology for each new subject would be time consuming and not very practical. The LDA classifier of the Sitting Position scenario, with more than 90% accuracy, is demonstrated to be able to reliably facilitate foot-related intelligent assistive devices and joystick-type controllers, especially if the subjects are trained to use their foot gestures as controller more frequently. The overall prediction accuracy among walking-related scenarios, though not as remarkably high as in the Sitting Position scenario, can reach up to 75%.

The lack of hip muscles, dynamic nature of the movement, difference between the manner of performing the task, and the inability of the C-ML models to adapt to the new participant are recognized as the downsides of the C-ML model. DL methods can be investigated to overcome the shortcomings of C-ML in detecting dynamic scenarios, enhance the prediction accuracy and generalization and adaptation capabilities of the learning model by providing the transfer learning framework.

Chapter 5

Lower Limb Intention Detection Using Deep Learning

This chapter initially goes through the notion behind employing DL in Section 5.1 and the structure of neural networks in Section 5.2. In Section 5.3, a specifically designed CNN is detailed and several implementation methods are explored to find the best way of adapting to a new user while enhancing the algorithm's generalization capability and robustness. The results are presented, discussed and compared to the results of Chapter 4 in Sections 5.5, 5.6 and, 5.7, respectively. At the end, an analysis of the optimal configuration of sensors in each scenario is presented.

5.1 Introduction

Before 2015, the literature on sEMG-based intention detection/gesture recognition primarily focused on Feature Engineering using predefined functions to find discriminators to characterize the sEMG signals. Since its introduction, DL has gained significant attention from gesture-recognition researchers. DL is a sub-field of machine

learning that deals with Deep Neural Networks (DNNs). DNN are deeper versions of Neural Networks (NNs) that have more generalization and learning capabilities. The recent enhancements in software structures have provided the groundwork enabling the training of DNNs in a shorter time. This has made DNNs a feasible option to utilize for detection tasks. DL's prominent characteristic is its ability to automatically extract high-level features from raw data using statistical methods. This is a different approach compared to C-ML, where experts must manually define specific functions to extract the most significant features of the inputs that can be separable using various C-ML methods. In addition, DNNs are able to depict uncertainties like inherent noise, electrode re-positioning, sweat at the electrode-skin site, fatigue, and physiological changes. Altering the paradigm from C-ML (Feature Engineering) to DL (Feature Learning), regardless of the method employed, comes with the end-goal of improving the classifier's robustness. DL has shown promising performance in EMG-based intention detection as detailed in Section 2.3.2.

Another deciding factor whether to employ the DL method or not is the amount of accessible training data. When performing some particular tasks, a single user cannot realistically provide tens of thousands of examples in a single session. Tests involving gesture recognition are one example of them. One way to overcome this issue is by aggregating multiple participants' recorded data in one or multiple sessions. This lays the ground for DL methods to have access to many samples of data that they require and learn a general mapping of users' sEMG signal. Moreover, DL enables Transfer Learning (TL), a method to leverage inter-user data by initializing the network's weights through pre-training a model on multiple subjects before a final training and adaptation that runs on a new participant's data. Combining an improved TL concept with an adaptive method (i.e. AdaBN [49]), can enhance the models' understanding to differentiate between the task and the user. This way, when adapting to a new user,

the network already has an understanding of the features of tasks and by requiring less amount of data from the new user, it can further learn the user's specific features for performing those tasks. Moreover, since brain injuries and illnesses have a wide variety, various causes and different effects, the intention interpretation methodology has to be customizable and adaptable due to the high variability of the persons with CP or other illnesses. As a result, the adaptability to specific subjects' needs is a very strong and important feature of this thesis's approach when applied to the target population.

5.2 Deep Learning Fundamentals

5.2.1 Deep Neural Network Basic Structure

At their core, NNs' are inspired by the signal processing performed in a neuron. They mimic the brain neurons' behavior via using a weighted sum of the input values plus a bias. The resultant of these combinations passes through a nonlinear activation function which acts like a threshold that generates the neuron's output. Neural Networks are structured as layers of neurons. Each neuron receives input from previous layer's neurons (calculated output of activation function), and generates an output based on its own layer's activation function. The weighted sums, computed and updated by one or more hidden layers, propagates, and finally reaches the output layer, the neurons of which compute the final output of the network [105]. This process is depicted in Fig. 5.1 in a nutshell. The output of each neuron through an activation function in each layer is calculated by:

$$a_j^l = f(\theta) = f\left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l\right) \quad (5.1)$$

where w_{jk}^l denotes the weight for the connection from the k^{th} neuron in the $(l-1)^{\text{th}}$ layer to the j^{th} neuron in the l^{th} layer, b_j^l is the bias of the j^{th} neuron in the l^{th} layer, and a_j^l is the activation of the j^{th} neuron in the l^{th} layer. The sum is taken over all k neurons in the $(l-1)^{\text{th}}$ layer. The activation of the input layer (a_j^0) is the input values (x_i). The function $f(\theta)$ or $g(\theta)$ is the activation function and can take various forms based on the layer it is being used on.

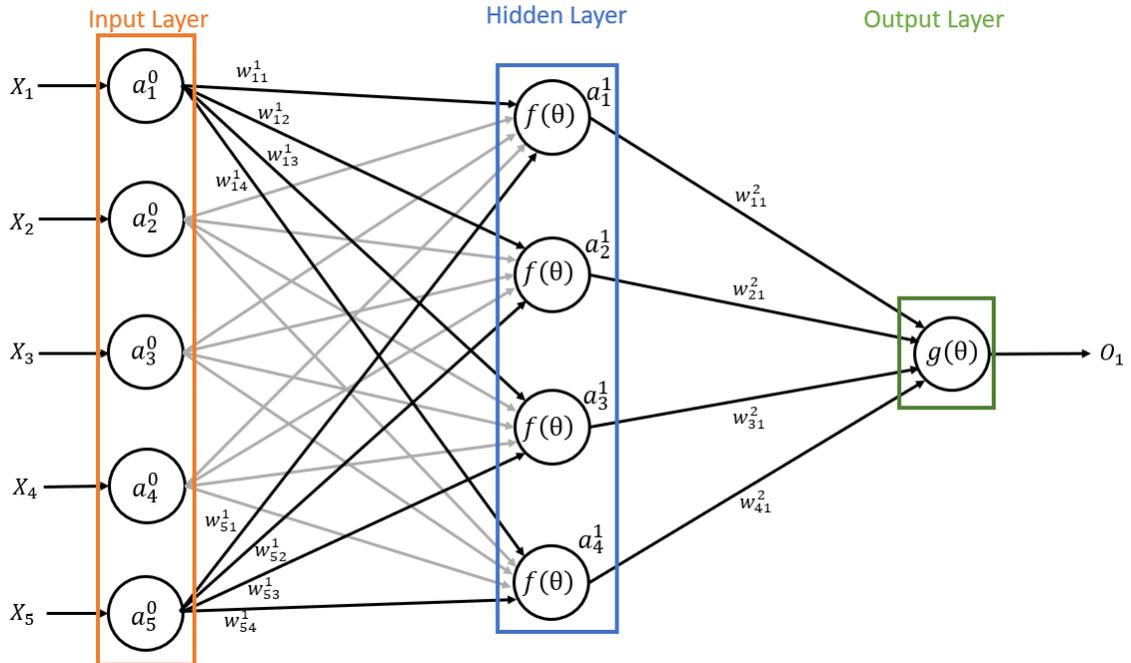


Figure 5.1: Model of the computations performed by brain neurons: x_i are the input activations, w_{jk}^l are the weights of the connection between neurons on each layer, $f(\theta)$ is the non-linear activation function and the resultant of this function which would be the input of next layer is shown as a_j^k , $g(\theta)$ is the non-linear activation function of each neuron in the output layer.

DL, which focuses on DNNs, deals with networks that have more than one hidden layer or, equivalently, more than three layers in total. DNNs can learn high-level features with more complexity and abstraction than shallow networks.

In all types of NNs, the basic program is implemented in the form of functions structured in layers. Learning consists of determining the value of the network's

weights (w_{jk}^l) and biases (b_j^l) through an output error optimization during the training phase. Once trained, the network can execute its task by computing the output from the input using the optimized weights and biases. For a classification task, the training scenario relevant to this work is supervised training (where the true classes are known and used during optimization). In this process, the trained DNNs receive input data and return a vector of scores, one score for each class. The highest-score class is the one that the network identifies as the most probable estimate of the true class for a given input.

5.2.2 Convolutional Neural Network (CNN) Basic Structure

Various architectures and sizes of DNNs have been introduced aiming to improve their performance relative to their application characteristics. CNNs are a successful class of architectures, which are introduced only after surveying the strategies to progressively reduce the storage and computation required by DNN's layers. CNNs fall into the category of feed-forward networks. In a feed-forward network, all computations are executed as a sequence of operations taking place from one layer to the next. Therefore, a feed-forward network has no memory, and the output for a given input is always identical irrespective of the history of the inputs fed previously. Fundamentally, CNN is composed of multiple Convolutional layers followed by an activation function, Pooling Layers, and Fully Connected (FC) layers, each defined as follows:

- **Convolutional Layer (CONV):** generates a progressively higher-level representation of the essential information in the input data. Each convolutional layer mainly consists of high-dimensional convolutions called the input feature maps (input fmaps), as shown in Fig. 5.2. Generally, in this computation, the input activations of a layer have the structure of a set of 2D input fmap, each referred

to as a channel. Each channel is convolved using a distinct 2D filter from the stack of filters, one for each channel. The convolutions' results are summed across channels, and a 1-dimensional bias is optionally added to them. The outcome of this computation is the output activations constituting one channel of the output fmap. By applying additional filters on the same input, additional output channels can be generated.

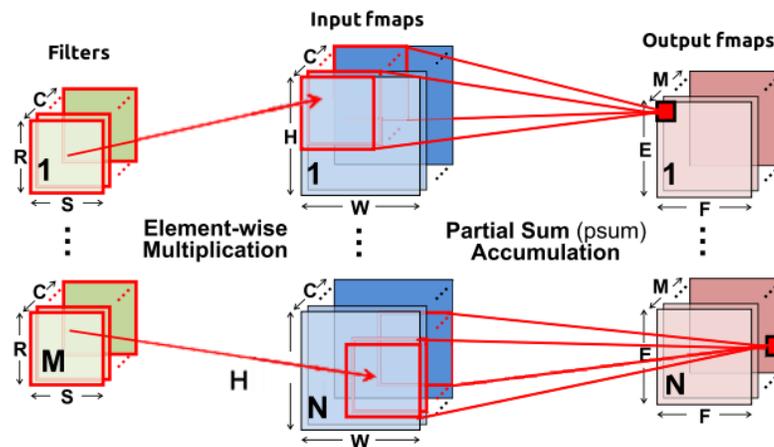


Figure 5.2: convolutions in CNNs, image from [105] - Permission to include the image was obtained from IEEE.

- **Pooling Layer:** is typically used as a block in CNNs to reduce the number of parameters and computation in the network. They perform a specific function such as max pooling, which takes the maximum value in a certain filter region, or average pooling, which takes the average value in a filter region to progressively reduce the representation's spatial size. The pooling layer operates on each feature map independently.
- **Fully Connected (FC) Layer:** is used to flatten the results just before the classification. Neurons in an FC layer have full connections to all activations in the previous layers, as it is in regular NNs, and they function similarly.

- **Output Layer:** as the final layer, produces an estimate of the probability that the highest-level features represent a particular task.
- ★ **Activation Function:** Non-linear activation functions are typically applied after each convolutional or FC layer, or after Batch Normalization (BN) (if present). Choosing the right activation function is an essential step to enhance the performance of any NN. Historically, the Sigmoid and the Hyperbolic Tangent (htan) are the most conventional. Simultaneously, the Rectified Linear Unit (ReLU) has become common in the recent years due to its simplicity and ability to make training faster. The Leaky ReLU and Parametric ReLU (PReLU) are variations of the ReLU explored to increase accuracy. PReLU is a type of leaky ReLU that, instead of having a predetermined slope, makes the slope a parameter for the network to figure out on itself: $f(x) = \alpha x$ when $x < 0$.

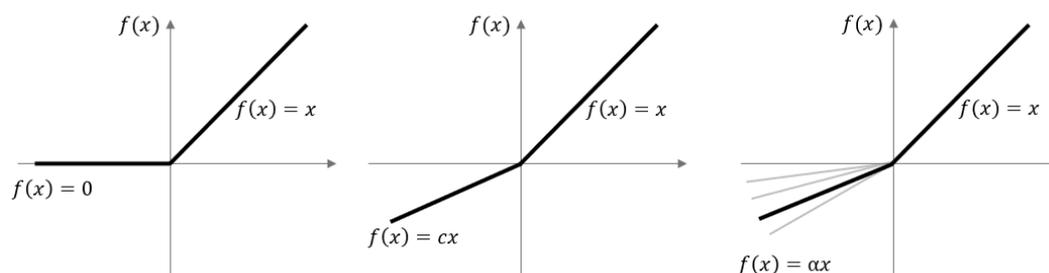


Figure 5.3: Samples of modern activation functions: (Left) ReLU, (Middle) LeakyReLU and (Last) PReLU

5.2.3 Network Optimization

DNNs are optimization algorithms that learn values hidden in each filter's weights and biases in order to determine the best value that can be used to differentiate between different classes (each class representing an identified task). Trained DNNs

receive input data and return a vector of scores, each element referring to one class. The class with the highest score is considered the most probable one. Since this is a supervised learning method, the class's true label is known, and the dissimilarity between the known ideal scores and the scores computed by the DNN (based on its weights and biases in each step) can be calculated. This calculated dissimilarity is called the loss (L), which is the objective function to be minimized by the optimizer. Gradient descent is the fundamental method used to perform optimization in DL networks. In order to tackle complex training scenarios where gradient descent methods behave poorly, multiple variants of gradient descent have been introduced in recent years. Adaptive Moment Estimation (Adam) is the most robust and efficient optimization algorithm for DL. It can be looked at as a combination of RMSprop and Stochastic Gradient Descent (SGD) with momentum [106]. The most prominent features of Adam are: (i) it can handle sparse gradients on noisy datasets, (ii) it is computationally efficient, (iii) it requires little memory, thus memory efficient, and (iv) it works well on large datasets and large parameters.

5.2.4 Regularization

One of the well-known problems occurring during the training process of a small data volume or a large data volume with very deep network is over-fitting, especially in DNNs with a large number of parameters to learn. Regularization methods have been used to prevent the networks from overfitting. Examples of regularization methods include Dropout, BN, and L1 and L2 regularization. Dropout and BN are used to prevent over-fitting in this research which are described as follow:

- **Dropout:** is a regularization method, which avoids the over-fitting problem by reducing the co-adaptation of hidden neurons [107]. In an FC layer where most

of the weight parameters in between layers -corresponding to connections - are occupied, neurons develop co-dependency amongst each other during training. In this case, some of the connections will have more predictive capability than the others. Through further iterations, these powerful connections are learned more while the weaker ones are ignored. Over many iterations, only a fraction of the node connections are trained and as a result, the rest stop participating. This will lead to the over-fitting of training data. To avoid this, during training individual neurons are randomly either dropped out of the net with a probability of $(1 - p)$ or kept with a probability of (p) , so that a reduced network is left for each sample of the mini-batch. This way, Dropout forces a neural network to learn more robust features that are useful in conjunction with many different random subsets of the other neurons.

- **Batch Normalization (BN):** is a technique that accelerates training convergence and provides some form of regularization by controlling the input distribution across layers throughout training [108]. BN accomplishes this by normalizing the mean (μ) and variance (σ) of each dimension of a batch of examples. The linear transformation that is applied to each channel on each layer of the network to bring mean to zero and standard deviation to one can be described as:

$$y = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}}\gamma + \beta \quad (5.2)$$

where the parameters γ and β are learned from training, and ϵ is a small constant used to avoid numerical problems. BN is mostly applied between the Convolutional or FC layer and the non-linear activation function, and is usually turned off after training. At the time of the test, these parameters are used to normalize the layer activations. BN has been shown to yield faster training times while allowing better generalization of the network.

5.3 Proposed CNN Architecture

The proposed CNN architecture for the application of this thesis takes the pre-processed window of eight channels sEMG data with a length of 260ms (which is collected, segmented and filtered as described in Chapter 3). During the trial and error procedure (tuning models hyper parameters), the best CNN model was identified as depicted in Fig. 5.4. The final architecture consists of two convolutional layers and two pooling layers. PRelu has been used as a nonlinear activation function in the proposed CNN. PRelu has shown that it can increase the performance of classifier types used in this work significantly. BN and dropout are used to prevent overfitting. Finally, an FC layer is used just before the output layer. The parameters used to train CNN are presented in Table. 5.1. An NVIDIA Tesla V100 GPU card, version 10.0.10 cuda, was used to implement the CNN. The architecture is implemented using PyTorch library version 1.6.0. and torch vision version 0.7.0.

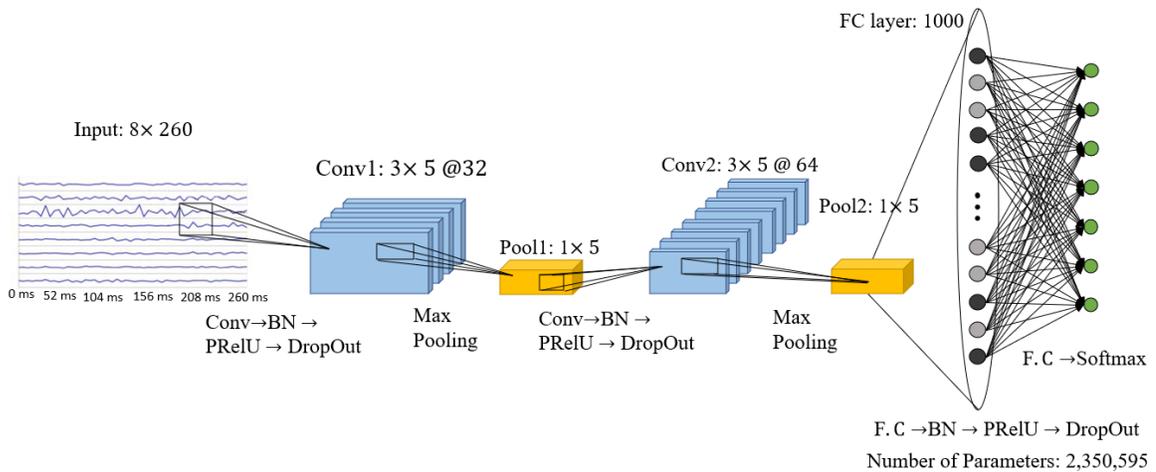


Figure 5.4: Designed CNN architecture for eight channel sEMG input.

Table 5.1: Parameters of the proposed CNN architecture

Parameter	Value
Optimizer	Adam
Loss Function	Cross Entropy
Initial Learning Rate	0.003
Mini-Batch Size	256
Number of Epochs	500

5.3.1 Transfer Learning

Among the main factors contributing to accuracy of predictions, especially when working with DL algorithms, is the amount of training data available. For gesture recognition, collecting data from a single user to train a supervised learning model can be prohibitively time-consuming. By meticulously pre-training a classifier on the accumulated data of preceding subjects before adapting to a new participant, one can potentially reduce the amount of data required from a new user while enhancing the system’s accuracy and generalizing its applicability to the inter-user and inter-session cases. The most prevalent TL method to generalize a model and adapt it to different subjects and sessions is fine-tuning. This method uses data collected from participants to train a model and then, uses the trained weights as a starting point when presented with a new participant. This method is proven to be prone to catastrophic forgetting [109]. This means that relevant and significant information learned during pre-training is lost when applied to a new participant. This method performs even worse when considerable differences between the initial and the new participants or manner of performing the tasks exist. This is because the method can bias the network into poorly adapted features. The gesture recognition task explored in this work falls into the same category due to the fact that each individual has a distinct

manner of performing each task, which was also observable during the experiment conducting sessions. Thus, this method is inherently prone to misinterpretation.

Progressive Neural Network (PNN) is another TL method, which is immune to catastrophic forgetting. PNN pre-trains the model on the Source Network and freezes the learned weights. Contrary to fine-tuning which applies the stored weights as a starting weight on the new subject, PNN creates a new network for new subject with random initialization of the weights. The new network then gets connected to the Source Network with freezed weights via non-linear lateral connection [109].

Due to the extreme dependency of DNNs on the training data set, their learned features cannot be easily transferred to a different but relevant task without fine-tuning. Adaptive Batch Normalization (AdaBN) was initially proposed to increase DNNs' adaptation capability by modulating the statistics in all BN layers across the network. This method can increase the subject adaptation effect [110]. Later, AdaBN was utilized in the context of gesture recognition under the hypothesis that the differentiating information specific to each gesture is stored in the weights of each layer, whereas the distinguishing information related to the difference between various participants is represented by the statistics of the BN layer [49].

Combining the PNN intuition and the AdaBN hypothesis in the context of gesture recognition, an advanced Transfer Learning-based architecture is established for gesture recognition with the capability of adaptation to a new user [52]. In this method, instead of one Source Network per subject during pre-training, a single Source Network is shared among all participants. However, BN statistics from each subject are calculated independently from one another. Thus, during pre-training, only the Source Network is trained to represent the information of all the participants in the pretraining dataset. At the stage of adaptation, the Source Network gets connected

to the Second Network by element-wise summation multiplied by a learnable coefficient to generate the Target Network that will be used to adapt to the new subjects way of performing the tasks. Thus, the Second Network only needs to learn weights that express the difference between the new target and source task. TL, in this case, occurs by freezing all the network weights (learned during pre-training) when training on the target user, except for the parameters of the BN. AdaBN employs almost identical network for both previously-learned and new tasks. The only difference is in the percentage of Dropout used in Source Network (35%) and the Target Network (50%). A slightly different architecture for the Source Network and Target Network were required based on the trial and error analysis to augment the performance of the system as a whole, as suggested by [52]. This seems to indicate that the two tasks (i.e. learning a general mapping of hand gestures and learning a subject-specific mapping), might be different enough that even greater differentiation through specialization of the two networks might increase the performance further. Fig. 5.5 demonstrates the architecture of this advanced TL method.

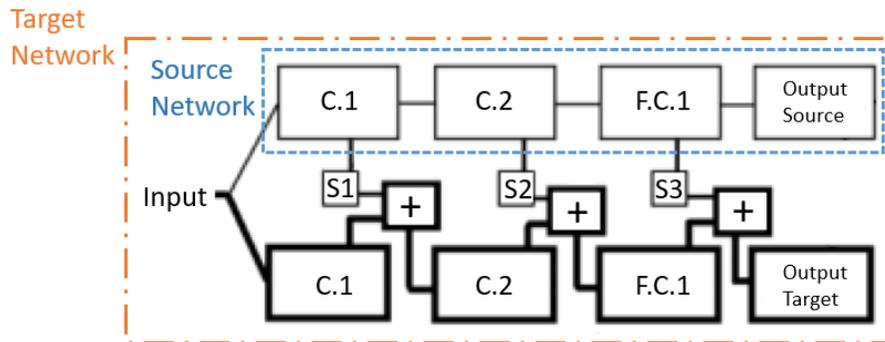


Figure 5.5: Advanced-TL architecture. C.1,2 and F.C.1 correspond to the two stages of convolutions and one stage of FC layers respectively. The $S_i (i = 1, 2, 3)$ boxes represent a layer that scales its inputs by learned coefficients. The number of learned coefficients in one layer is the number of channels or the number of neurons for the convolutional and FC layers respectively. The + boxes represent the merging through an elementwise summation of the output of the corresponding layers.

5.4 Different Implementation Algorithms

Three different implementation algorithms were designed and their performances on each participant were investigated. Same as before, all three algorithms are singling out the same one cycle for eventual testing (of total five cycles collected), but they train their models differently, as described below. The main goal behind this was to lower the burden of collecting a lot of data from a single user while preserving the algorithm’s performance. Meanwhile, reducing the training time required to adapt to the new participant is also considered as the deciding factor to develop the implementation methods. Each Algorithm is detailed below:

- **Subject-Exclusive Algorithm:** where a Source Network is trained and validated per each participant. The training and validation sets are randomly chosen, having a 90% and 10% portion of main train set, respectively. The accuracy of the trained model is then evaluated on the test set of that same participant. The overall performance of the model is then evaluated by averaging these individual accuracies. This algorithm can represent how the model works for each participant, since the learning is not generalized over all the participants.
- **Generalized Algorithm:** in which a pool of samples of all four training cycles of all the participants are concatenated to form a training set. The samples of the test cycle of all participants merge to form the test set. After randomizing, the designed CNN architecture is trained using 90% of the training set, and the remaining 10% is used for validation. Ultimately, the accuracy of the model is evaluated on the test set.
- **Subject-Adapted Algorithm:** where the algorithm uses the advanced-TL architecture to reduce the burden of acquiring immense data from each participant,

while enhancing its prediction accuracy once adapted to a new participant. In this algorithm the Source Network of advanced-TL architecture is trained across a certain number of participants (5 participants out of 9 in this study). In the same manner, 90% of the main train data set of all participants’ is used for training, and the remaining 10% is used for validation. Therefore, when pre-training on the Source Network, the data from all subjects are aggregated and fed to the network. During the adaptation phase, the data from each new participant is used to train and learn the Target Network’s weights and connection coefficients to adapt to the new user’s specific features. The final accuracy is reported as an average of the model’s performance on the 4 remaining participants. It is important to note that each training batch is comprised solely of examples that belong to a single participant. This allows the update of the participant’s corresponding BN statistic.

5.5 Results

Due to the stochastic nature of the algorithms presented in this chapter, unless stated otherwise, all experiments are reported as the average of 20 runs. Each run goes over each implementation scenario with a new random selection for training and validation, and eventually reports it by averaging them all. Results of the trained different algorithms in each scenario are presented in Tables. 5.2 to 5.5.

Table 5.2: DL models’ accuracy in Sitting Position scenario

	Generalized(%)	Subject-Exclusive(%)	Subject-Adapted(%)
4 Cycles	90.33	91.79	91.26
3 Cycles	88.57	88.96	86.05
2 Cycles	82.72	81.95	82.84
1 Cycles	78.76	72.45	82.83

Table 5.3: DL models' accuracy in Standing Position scenario

	Generalized(%)	Subject-Exclusive(%)	Subject-Adapted(%)
4 Cycles	74.98	75.02	71.41
3 Cycles	73.55	72.68	70.88
2 Cycles	66.17	63.61	62.25
1 Cycles	54.01	48.24	47.63

Table 5.4: DL models' accuracy in Attempted Motion scenario

	Generalized(%)	Subject-Exclusive(%)	Subject-Adapted(%)
4 Cycles	51.86	54.6	46.16
3 Cycles	48.79	50.30	41.49
2 Cycles	45.7	49.93	40.86
1 Cycles	44.9	40.28	39.27

Table 5.5: DL models' accuracy in Dynamic Motion scenario

	Generalized(%)	Subject-Exclusive(%)	Subject-Adapted(%)
4 Cycles	66.29	67.65	76.85
3 Cycles	66.22	62.47	73.2
2 Cycles	62.08	55.94	70.21
1 Cycles	53.46	42.14	57.27

5.6 Discussion

As predicted, Tables. 5.2 to 5.5 clearly illustrate that increasing the number of cycles and correspondingly the number of examples during training can improve the CNN model's performance. This was expected due to DL models' main requirement to access as many samples as possible. Further, it is worth noticing that the difference between the final accuracy of different methodologies gets smaller as the number of

cycles increases. Therefore, one can observe that the model’s final accuracy becomes less dependent on the number of cycles once at least four cycles of data are collected.

With 4 cycles of training, the Subject-Exclusive implementation method’s performance always surpasses the Generalized Method. However, this is not always true in the lower number of cycles. The reason behind this is the fact that the network has access to less amount of data for training and validation in the Subject-Exclusive case (which only contains data from n -cycles of each participant). On the other hand, in the generalized method, the network has access to a larger data set (n -cycles of all participants). This deduction emphasises on the dependency of the CNN’s performance on the size of the input dataset.

The Subject-Adapted algorithm combines the features of Subject-Exclusive and Generalized methods to build a robust but adaptable learner. It achieves this goal by generalizing over different participants to learn gesture patterns and adapting to individual new test subjects by manipulating the BN layers’ weights to discern and adapt to the coming user that performs that gesture in their own distinct manner. Results illustrate that the Subject-Adapted algorithms can enhance the generalized models’ prediction in some scenarios. Due to the time limitations of this thesis, only 9 subjects were examined in total. The Subject-Adapted scenario uses a portion of subjects (5 Subjects in this reserach) during pre-training (generalization) phase and the remaining portion for evaluation during evaluation (adaptation) phase. Since these numbers of pre-training and evaluation are relatively low (half of the Generalized method), we believe that the network did not have access to as many samples as required to fully demonstrate Subject-Adapted mode’s capabilities. The final hypothesis here is that by increasing the number of participants’ during generalization phase this model can achieve an accuracy higher Generalized method or at least similar to the Subject-Exclusive scenario, guaranteeing that the model is robust enough

to detect the gesture and adapt to the new user.

In the **Sitting Position Scenario** with 11 distinct gestures; Subject-Adapted method was able to surpass the Generalized method and achieve an accuracy close to Subject-Adaptive method. Thus, one can deduce that the Subject-Exclusive method was able to generalize and adapt to the new user quite promisingly. High accuracy that is achieved in this scenario can prove that the lower limb intention detection via sEMG which is adapted to a new subject through Subject-Adapted method, is reliable for facilitating foot-related intelligent assistive devices and joystick-type controllers.

In the **Standing Position Scenario** with 9 distinct gestures; the Generalized method was able to get very close to the Subject-Exclusive method's prediction level. This can indicate that the network was able to find the gesture distinguishable features between all participants even though the dissimilarity they had when performing each gesture. However, Subject-Adapted implementation method was not able to reach the Generalized and Subj-Exclusive methods' performance. Since it could not reach the Generalized method's performance, the only reasonable justification here is that the network did not have enough information (due to smaller training set) to get generalized enough. Therefore, increasing the number of participants during pre-training can improve network generalization capability and eventually raise its prediction accuracy when used in the adaptation phase.

In the **Attempted Motion Scenario** with 9 distinct gestures; the Generalized method was not able to get close to the Subject-Exclusive method's prediction. Moreover, the Subject-Adapted method worsens the prediction of the Generalized method even further. This can demonstrate that in the case of complex gestures, like Attempted Motion scenario where different subjects have very different manners of

performing, relying on any of the generalization methods would result in poor prediction. Moreover, the overall prediction accuracy of this scenario is lower in comparison to other scenarios. This can be justified by taking a close look at the nature of this scenario; which is to mimic the behavior of muscles' of the target group who are experiencing some level of disability. One can conclude that artificially asking the participants to restrain themselves from doing a task while generating the required force might inversely effect some of the highly important and informative muscles for distinguishing between gestures, or it might disturb the true correlation between muscles (i.e. agnostic and antagonistic muscles). Therefore, experimenting with the target population is suggested instead of mimicking the disability. Hence, the only possible option is to collect more data from the same participant and train using the Subject-Exclusive method.

In the **Dynamic Motion Scenario** with 9 distinct gestures; the Generalized method was able to get close to the subject exclusive method's accuracy. Moreover, the Subject-Adapted algorithm outperformed the other two methods remarkably. This can highly demonstrate that the algorithm was fully capable of distinguishing between activity patterns and features related to each participant. Thus, when it got adapted to the new participant through the Subject-Adaptation method, it was able to capture participant-related features and consequently improve the performance of this method.

Same as the results of the C-ML method, a drop in detection accuracy in Standing Position, Attempted Motion, and Dynamic motion scenario as opposed to Sitting Position scenario can be observed. The main Justification for this can be explained by the lack of data from hip; where the main related motion is generated.

5.7 Comparison of C-ML and Deep Learning

Comparing results obtained using the C-ML approach with the DL approach demonstrate that DL can achieve an accuracy at least as good as the C-ML approach in all scenarios when trained in either Subject-Exclusive or a Generalized manner. Since the dataset available for this research is comprised of a limited number of subjects, it can be reasonably assumed that by increasing the number of subjects, DL will exhibit even better results. The Subject-Adapted algorithm of the DL approach showed that although with the data set at hand, it cannot reach the level of accuracy of a Subject-Exclusive approach, it has the capacity to perform better as the pre-training data pool enlarges, specifically in scenarios where there is not a big difference between the manner of performing each class among participants. This algorithm can alleviate the burden of collecting immense data from each subject while exhibiting a reliable behavior in comparison to the Subject-Exclusive method.

Moreover, the DL-based learning method remarkably improved the prediction level in the Dynamic Motion scenario, specifically when trained in a Subject-Adapted method. This should indicate that the distinct difference between the manner of performing the activities was recognizable by the Subject-Exclusive method utilized in this scenario (as apposed to Attempted Motion scenario).

It is well-known that DL is capable of exploring and classifying the data set without any assistance in terms of manual feature selection. On the other hand, C-ML models require features to be selected for them by the designer. They also become very susceptible to these features that might affect their prediction performance. But, since known features are chosen and inputted to C-ML models, these models are more interpretable in comparison to DL. Although in terms of prediction accuracy in different implementation scenarios, DL surpassed C-ML as expected, top-performing

C-ML models' strength should not be underestimated since their prediction accuracies were relatively close to the DL approach in our small dataset while having a lower computational cost. In general, we can say that, for smaller datasets, C-ML models can be the go-to option due to their low computational cost and relatively similar performance to DL. In a larger dataset, DL can overshadow C-ML due to its ability to learn complex features and functions independently. TL is another capability that can be added on top of DL, which the solo C-ML lacks. It allows for better adaptability and generalization, which is particularly beneficial to the current study by lowering the size of required data collected from individual subjects. Further, by transferring part of the previously trained models to new sessions/subjects, it reduces the computational cost when applied to a new subject.

5.8 Identifying the Optimal Configuration of Sensors

Decreasing the number of sensors will reduce the time required for preparation and, more importantly decrease the load and burden imposed on the participants. The main trade-off here is that decreasing the number of sensors means decreasing the input channels, thus not utilizing some information. However, identifying the most informative channels can be beneficial if it is combined with other sources of data (EMG of the hip, IMU or force under the feet). In this section, different configurations of four channels out of the original eight channels are selected. Their performance are examined when the Subject-Exclusive DL algorithm is applied to them. Since the Subject-Exclusive is the best performer that we have at hand, it is plausible to say that the Subject-Exclusive algorithm has the highest chance of making a prediction

using only four input channels. If Subj-Exclusive detection results are not reliably accurate, we can confidently say that the other two implementation approaches would also not generate desirable results. In terms of CNN architecture design, the input layer, in this case, comprises of a window of four channels of sEMG data for the length of 260ms (segmentation and filtering as detailed in Chapter 3). Thus, an alternative CNN architecture is designed as shown in Fig. 5.6.

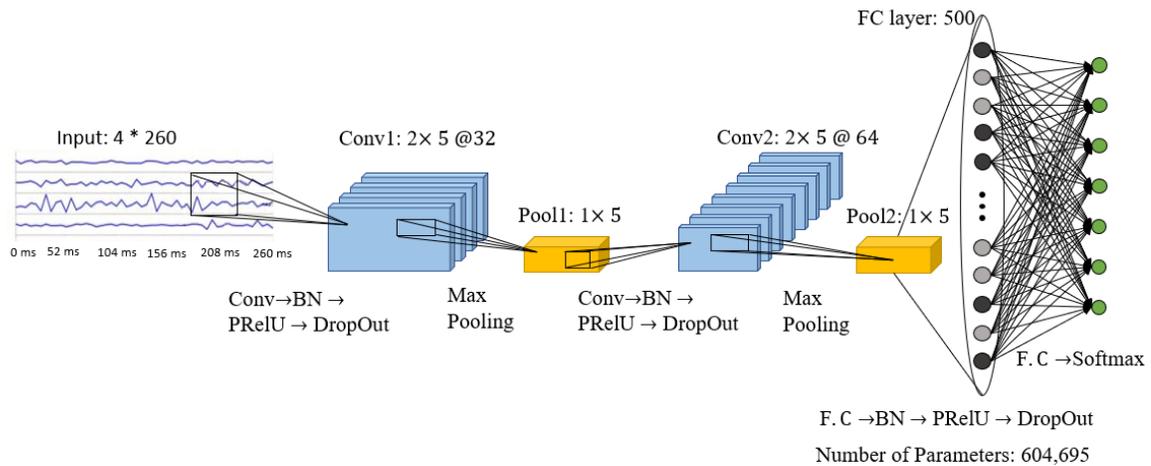
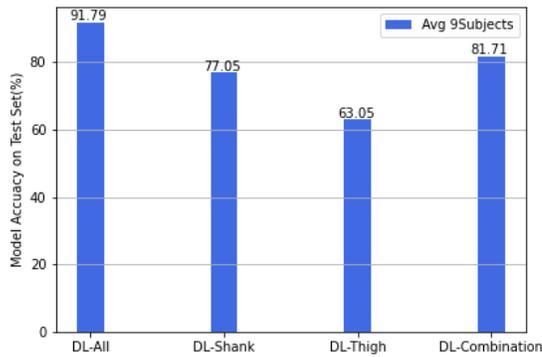
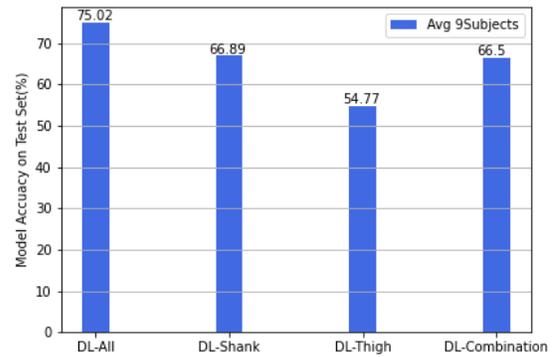


Figure 5.6: Designed CNN architecture for four channel sEMG input.

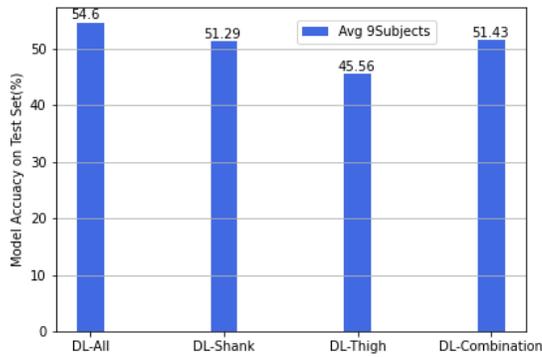
Three different configurations of muscles are selected to investigate as:(i) all four shank muscles, (ii) all four thigh muscles, (iii) combination of two of shank (TA and GL), and two of thigh (BF and VM) muscles. A complete comparison of results obtained with each configuration of muscles in each scenario is illustrated in Fig. 5.7.



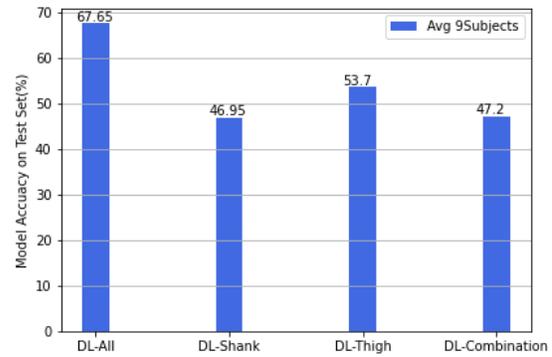
(a) SittingPosition



(b) StandingPosition



(c) AttemptedMotion



(d) DynamicMotion

Figure 5.7: Performance comparison of various sensor configurations in each scenario

Results can show that the best model with 8-channel input data always surpasses the 4-channel input. Using combined muscles group showed better results than opting for only a portion of the leg. Though, this is not true for Dynamic motion scenario. This can be attributed to the fact that during Dynamic Motion, muscles of the upper side of the leg are critical indicators of the activity and the direction of the motion.

5.9 Chapter Summary

The results illustrated that increasing the number of cycles and consequently enlarging the training data set enhances the accuracy of prediction immensely. Overall, the DL approach achieved better results than the C-ML approach when implemented either in the Subject-Exclusive or the Generalized strategy. An advanced TL-based Subject-Adaptive implementation method was introduced to leverage inter-user data from the pre-training datasets and adapt to new users according to their distinct manner of performing the tasks. This came with the final goal of enhancing the detection accuracy and alleviating the data generation burden imposed on a single individual. Although Subject-Adaptive method increased the prediction accuracy in Dynamic Motion Scenario and Sitting Position Scenario in comparison to Generalized method, it was not capable of improving the prediction in other two scenarios. This was justified by highlighting the fact that the Subject-Adaptive algorithm used half portion of data set that Generalized algorithm used and thus, due to smaller amount of data in pre-training phase it could not generalize enough. Therefore, it is proposed to increase the size of pre-training data set to fully exploit the potentials of this method as well, particularly in Standing Position. On a different note, due to the complex nature of the Attempted Motion scenario, it showed better prediction accuracy when trained in a Subject-Exclusive manner. This is attributed to the varying and disparate gestures performed by different subjects in the Attempted Motion scenario.

Chapter 6

Application to the Control of Assistive Robotic Systems

In Section 6.1, the motivation behind integrating the intention detection system in conjunction with a proper controller for improving rehabilitation robots assistance is detailed. In Section 6.2, the basics of interaction controllers (impedance/admittance) and their implementation are discussed. In Section 6.3, the simulation environment where the intention-detection-augmented-controller is implemented and the dynamic substitute model for SoloWalk robot are detailed. The methods to integrate the intention detection system into the simulated interaction control logic are explained in Section 6.4. In Section 6.5, the designed test scenarios are presented. Eventually, the results of augmenting the robot with different methodologies and a thorough discussion on the effectiveness of each method and their comparison is presented in Sections 6.6 and 6.7, respectively.

6.1 Introduction

In the field of rehabilitation robotics, employing an assist as needed strategy and providing the required assistance exactly when the human requires it, are two of the most important factors. Assist-as-needed strategy based on volitional motion is about tuning the level of assistance correlated to the users' need and it ensures that the users are thoroughly engaged and compelled to take action [13, 111, 112]. In addition to that, as highlighted by various researchers, the immediate and corresponding response positively influences humans' neuroplasticity and promotes motor learning for better recovery and rehabilitation [3, 113]. Intention information of the user can be utilized by the robot to provide an immediate feedback response to the user based upon their detected intents and needs. The robot's reaction to the detected intent should be perceived by the user in the most stable, intuitive, and comfortable manner to enable the highest benefits in terms of rehabilitation and recovery. (e.g., some patients need assistance to initiate the movement in straight motion or may need the robot to stabilize their lateral motion while they practice stepping in their body's frontal plane). The intention detection system, as developed in Chapters 4 and 5 can provide information regarding the user's intentions immediately. It can convey the request for assistance to the robot's main controller so that it can alter its motion to better assist the user in correspondence to the detected intent. The most critical factor in ensuring the mentioned requirements is the choice of the main robot control methodology and its characteristics. The second important design consideration is how to augment the main controller with the intention detection algorithm in order to provide assistance when the need for assistance is identified.

Choosing the robot's main controller requires considering application requirements, determining the final goal, and predicting possible complexity in the system.

Primarily, any control system that enables cooperation between humans and robots needs to guarantee stable interaction (e.g., user inputs generate smooth, consistent, and predictable device motions). The most common controllers, position and force control, may behave unpredictably when interacting with an unknown environment (a human in this case) as they have no means of adapting their structure to changes in their environment. Furthermore, neither of them have the ability to satisfy the rehabilitation requirements (i.e. position control overlooks the assist-as-needed requirement by forcing the user's leg to go through a predefined motion while the user remains passive, and force control does not have any designed trajectory upfront that directs the user through a known beneficial practice). An alternative approach that has been broadly incorporated in the control of rehabilitation robots is interaction control with two possible causalities known as impedance and admittance control. Both these control methods dynamically model the relationship between force and position of the robot instead of solely controlling position or force during the human and robot's interaction. If designed meticulously, they can guarantee a stable user-device interaction. These controllers are capable of recreating the desired encoded virtual interaction dynamics in the form of tactile sensation felt by the user [53]. Due to their compliant nature, they are best suited for gait training rehabilitation robots where the assist-as-needed strategy should be granted and the robot's reaction should be perceived by the user immediately.

As it has been described in the project background Section 1.1, the SoloWalk gait training robot incorporates an admittance controller. Despite all of the advantages of using admittance controller, it also faces some inherent challenges. First, both of the impedance/admittance frameworks implicitly assume that the user is capable of generating the force or position inputs necessary to command the robot's motion, which is not always possible (e.g. when dealing with a patient with severe disability or CP).

Therefore, an intelligent intention detection system can augment these controllers by regulating its parameters for better assistance in these circumstances. In addition, there is significant theoretical and experimental evidence to suggest that typical admittance control implementations are intrinsically more unstable when required to render very small or soft impedances (e.g., when a robotic gait trainer is commanded to display a small mass) [114]. Thus, the user might experience an impulsive response when the apparent mass is low (which is desirable to ensure that the robot feels light from the user's perspective). Particularly, the effect of lowering the mass is expected to be felt more in the initiation of the motion phase because the robot is transitioning from static to dynamic motion. This might not be adequate from the comfortability perspective, or optimal from a therapeutic perspective [8]. Regulating the admittance controller parameters during this phase or assisting the user to smoothly initiate the motion can be addressed if an intention detection system is employed to detect the states properly and in advance. Moreover, people with disability might involuntarily input a misleading force command to the robot with their hands due to spasm, weakness, stiffness, unbalanced contractions, and etc. Thus, relying on just a single source of information (force sensor) might not be sufficient to predict the user's intention/state, especially if that source is located in a distant muscle relative to the source of motion (arm and feet in the case of SoloWalk). To demonstrate the practicality of the proposed intention-detection-augmented-controller throughout this chapter, a simulation of admittance control-based gait rehabilitation robot interacting with a human model is developed. Further, various methods of augmentation of the main admittance controller are examined and evaluated in terms of intuitiveness and compliant performance.

6.2 Admittance Control vs Impedance Control

In robotics and haptics applications that tactile feedback and mechanical stimuli are required, both impedance/admittance control are immensely used to impose a specific interaction at a contact interface (like rendering a virtual environment). This desired dynamic interaction and virtual environment is encoded by a commonly-used mass-spring-damper system [115]. Impedance/Admittance controllers utilize two opposite cause and effect relations for regulating the interaction dynamics between the robot and the environment (in this case human). It is generally known that admittance control is the inverse of impedance control. Therefore, their performance requirement, benefits, and implications are also complementary [116]. At the most basic level, the two control approaches can be differentiated as follows: impedance control transforms measured position inputs on the contact interface to force outputs imposed to the user based on a given virtual impedance, while admittance control transforms force inputs measured on the contact interface into position outputs according to its virtual impedance. Impedance controller have an inner force control law and admittance controllers have an inner position control law that follows the reference force or position outcome rendered by a virtual environment, respectively.

During the interaction between the robot and the environment, one must behave as an impedance while the other presents admittance. The impedance control law rests on a significant assumption that the robot itself has no inherent impedance (i.e. between the actuator and the end-effector). This is an idealized assumption since there will always be some level of friction, mass, or other impeding factors in the motor, gears, links, etc., that cannot be modeled and compensated thoroughly. Therefore, the model in impedance controller will most commonly include some level of error in reproducing real dynamics. These errors behave as uncertainties or external

disturbing forces and result in poor trajectory following in the free space tasks as required by rehabilitation devices. Therefore, devices controlled by an impedance controller should ideally have low inertia and friction (no hardware imperfections, if possible), which is not typically true for rehabilitation robots. In general, robotic systems with impedance control have upper limits on the stiffness that they are able to render, thus, work better when interacting with environments stiffer than themselves (they represent the admittance in the interaction). In contrast to impedance control, admittance control is intrinsically more capable of displaying larger impedances [117, 118]. Therefore, it can be incorporated on robots with a fairly high impedance (i.e. having large gear reductions and rigid links) and making sure that the inner position control law is accurate. Thus, admittance Control provides high level of accuracy in tasks that involve motion in free-space, while they have a lower limit on the stiffness they are able to emulate for a given robot.

Impedance Controller Implementation

Hogan proposed Impedance controller as a means of applying forces to the environment without the instability issues associated with direct force feedback [119]. Impedance control, as shown in Fig. 6.1, uses the measurement of the position and velocity of the robot, calculates the trajectory deviation of the robot, and translates that to a desired force based on the model's desired impedance. Then, with the hypothesis that the accurate multi-body dynamics of the robot is known, the desired force is followed using the inner force control loop.

Assuming that we have an accurate multi-body dynamics of the robot, by applying a control torque τ_c to the robot through the actuators, the robot moves under the

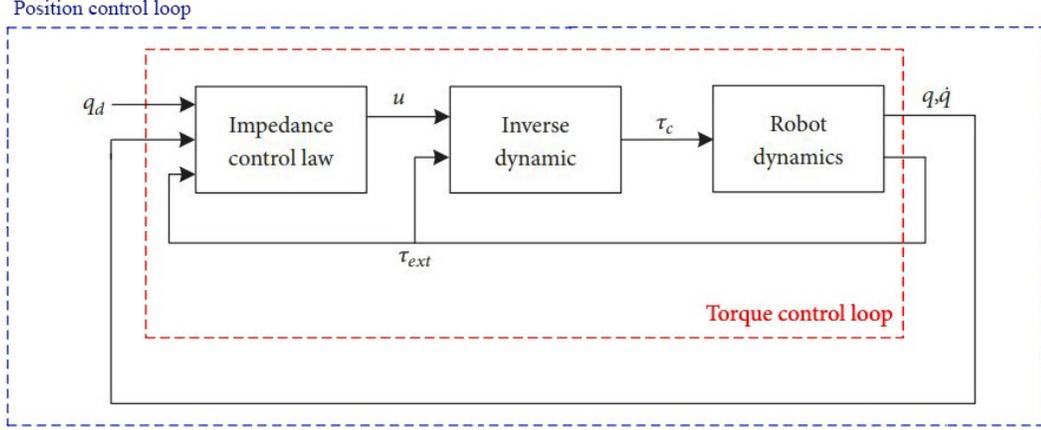


Figure 6.1: Block diagram of generic impedance control.

influence of the control torque and all other external torques, based on the following:

$$M_{rob}(q)\ddot{q} + B_{rob}(\dot{q}, q)\dot{q} + G(q) = \tau_c + \tau_{ext} \quad (6.1)$$

where $M_{rob}(q) \in \mathbb{R}^{N \times N}$ is the positive-definite matrix of the robot inertia in the joint space, $B_{rob}(\dot{q}, q) \in \mathbb{R}^{N \times N}$ is the coriolis force and centrifugal force, and $G(q) \in \mathbb{R}^N$ is the gravity moment on different joints. $\tau_c \in \mathbb{R}^N$ is the control moment and $\tau_{ext} \in \mathbb{R}^N$ is the sum of the external moments applied to the robot. $\ddot{q}, \dot{q}, q \in \mathbb{R}^N$ are vectors of position, speed, and acceleration of all joints of the robot.

The desired behaviour felt by the user can be described by the expected (predefined) dynamic parameters $M_d, B_d, K_d \in \mathbb{R}^{N \times N}$ (expected inertia matrix, expected damping matrix, and expected stiffness matrix, respectively). The deviation from this expected dynamics based on the measured trajectory following error Δq can be attributed to the external torques as:

$$\tau_{ext} = M_d \Delta \ddot{q} + B_d \Delta \dot{q} + K_d \Delta q \quad (6.2)$$

where the trajectory following error is calculated as

$$\Delta q = q - q_d, \quad (6.3)$$

where q_d is the expected joint position. The expected inertia is a fixed value, but the damping and stiffness matrix are variable.

Using the calculated τ_{ext} , the measured Δq , and the inputted desired trajectory $q_d, \dot{q}_d, \ddot{q}_d$, the reference force calculated from impedance control law is formed as follows:

$$u = \ddot{q}_d + M_d^{-1}(-B_d\Delta\dot{q} - K_d\Delta q + \tau_{ext}) \quad (6.4)$$

The inverse dynamic command τ_c is obtained based upon the reference force and the previously calculated external torques as:

$$\tau_c = M_{rob}(q)u + B_{rob}(\dot{q}, q)\dot{q} + G(q) - \tau_{ext} \quad (6.5)$$

Admittance Controller Implementation

Admittance control was first proposed by Whitney as a means to incorporate force feedback into a position controller [120]. In the same vein as the impedance control, admittance control aims to manipulate the robot's perceived dynamics from the user's perspective. Admittance control, as it can be seen from Fig. 6.2, generates a Cartesian reference trajectory (x_r) using the virtual desired impedance, based on the measured external force (F_{ext})(applied to the interface), a predefined additional direct force (F_d) acting as an adaptation mechanism, and a predefined desired trajectory (x_d). The inner position control loop follows the trajectory that comes from translating the reference trajectory into the joint space (q_r).

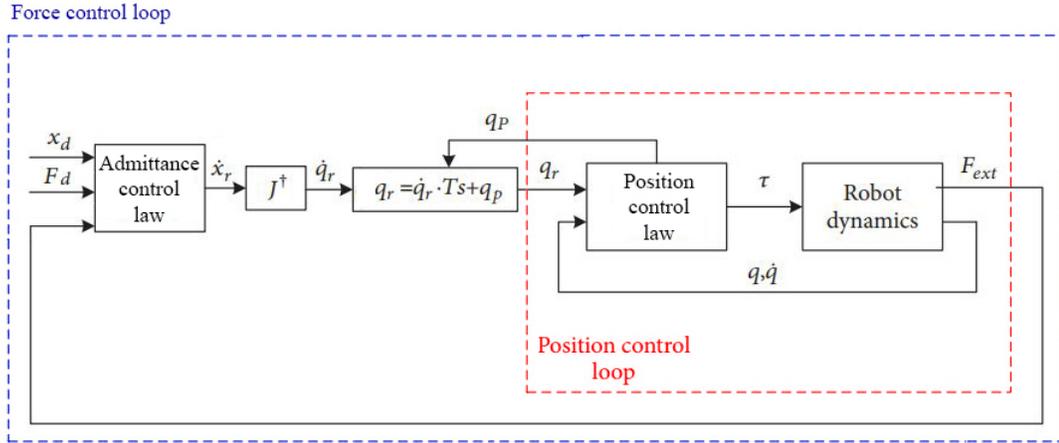


Figure 6.2: Block diagram of a generic admittance control.

The output trajectory of the admittance control law (\dot{x}_r) is calculated from the following expected dynamic equations of the end-effector:

$$M_d \ddot{e} + B_d \dot{e} + K_d e = F_{ext} + F_d \quad (6.6)$$

where $M_d, B_d, K_d \in R^{6 \times 6}$ are expected inertia matrix, expected damping matrix, and expected stiffness matrix, respectively. \ddot{e}, \dot{e} and e are positional deviation, speed deviation, and acceleration deviation from some desired equilibrium trajectory as:

$$e = (x_r - x_d) \quad (6.7)$$

$$\dot{e} = (\dot{x}_r - \dot{x}_d) \quad (6.8)$$

where x_r and \dot{x}_r are the output task-space reference trajectory. A modified version of dynamic equation of the end-effector (Eq.6.6) can be employed which facilitates the free-space cooperation of the human and the robot and ensures that the robot can be easily dragged by the human:

$$M_d\ddot{e} + B_d\dot{e} = F_{ext} + F_d \quad (6.9)$$

The inverse Jacobian then maps the calculated Cartesian task space reference trajectory x_r to the joint space trajectory q_r . Then an inner-loop (PD position controller) is used to ensure that the robot follows the reference trajectory in joint space. The PD controller law is:

$$\tau = K_p(q_r - q) + K_d(\dot{q}_r - \dot{q}) \quad (6.10)$$

where q_r and \dot{q}_r are the reference position and velocity in the joint space, q and \dot{q} are robot actuators position and velocity, and K_p and K_d are proportional gain and derivative gain.

6.3 Simulation

Because of limitations of real-life experiments during the COVID19 outbreak in the year 2020, it was decided to simulate the experiment scenario instead of empirical implementation. In this simulation, a robot with SoloWalk's characteristics assists an impaired Human model to follow a desired therapeutic trajectory. This desired trajectory is recorded in advance by an IMU sensor from a healthy human subject going through test scenarios. sEMG signals used in the simulation are also recorded from the same humans and during the same tests. The simulation aims to evaluate the degree to which augmenting the admittance controller by the sEMG-based lower limb intention detection system can improve the intended trajectory following accuracy of the impaired human model.

6.3.1 Model

Fig. 6.3 shows a block diagram of the developed simulation model, which is comprised of: a dynamic model of the user, a dynamic model of the robot, a contact model between the user and the robot, and the control model of the admittance controller that is augmented through various methodologies via the intelligent intention detector using sEMG signals.

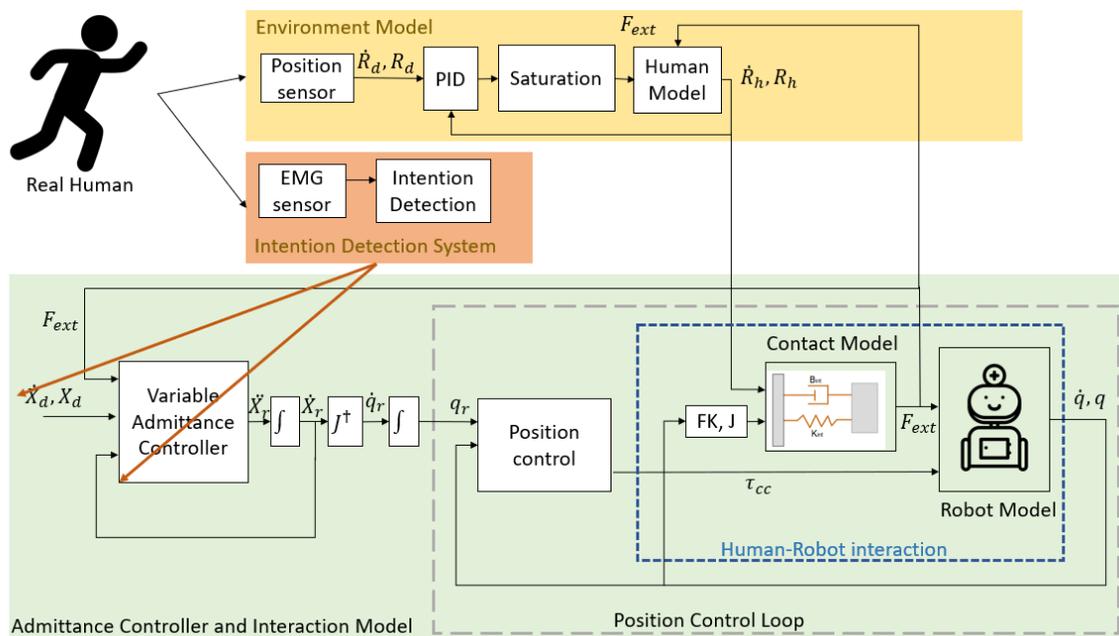


Figure 6.3: Block diagram of the developed simulation model

Elements of the developed simulation environment as shown in Fig. 6.3 can be described as follow:

- (i) A 3-DoF (x, y, θ) rigid body robot model with the inertia of 130 kg and the angular inertia about the z -axis of 5.41677 kg.m^2 is created via the available elements from Simscape Multi-body library.
- (ii) An admittance control block including the measured external force (F_{ext}) to

trajectory $(x_r, \dot{x}_r, \ddot{x}_r)$ translation admittance law and the inner position control PD loop is implemented. The detailed admittance control scheme in Section 6.2 was implemented on the robot model. Admittance control is adopted to regulate the dynamic behavior at the interaction point when the robot interacts with an unknown environment (human).

- (iii) The task-space to joint-space Jacobian generator is calculated in a separate block. It is worth noting that since a simplified rigid block model is used to represent the robot, the inner joint states of the robot overlap with the task-space properties, and therefore, the Jacobian in this specific case has become identity.
- (iv) An impedance contact model in the robot joint-space is employed to estimate the interaction force/torque between the human and robot model. This impedance contact model is coded as follow:

$$B_{cm}\dot{e} + K_{cm}e = F_{ext} \quad (6.11)$$

Where $B_{cm}, K_{cm} \in R^{3 \times 3}$ are the contact model's damping matrix and stiffness matrix, respectively. e and \dot{e} are positional deviation and speed deviation between human model and robot model as:

$$e = (x_h - x) \quad (6.12)$$

$$\dot{e} = (\dot{x}_h - \dot{x}) \quad (6.13)$$

where x_h and \dot{x}_h are the task-space reference trajectory of the human model and the x and \dot{x} are the task-space trajectory of the robot model. This contact model connects their movements, provides the opposing force to the human

model from the robot and provides to the admittance controller the expected force that the user would exert to the robot. The latter force (F_{ext}) is also the input to the admittance control law. It is worth mentioning that a threshold is also coded to ensure that the human model and the robot model don't intersect or get closer than a certain distance. The contact force is therefore driven to infinity as the distance between the two nears the defined limit.

- (v) An sEMG-based intention detection block is created, reading pre-recorded EMG measurements of a subject going through two different test scenarios and generating detected direction signals based on which the admittance control law is adapted to the subject's needs.
- (vi) For the intention detection to work with the sEMG signals of the real human, the simulation must either (i) include the subject in its loop, or (ii) include a model of the subject. For this work, it has been decided to develop a 3-DoF (x, y, θ) rigid body human model with an inertia of 67.5 kg and an angular inertia about the z-axis of 0.6328 kg.m^2 via the available elements from Simscape Multi-body library. This approach was taken because a human in the loop simulation of an admittance controller requires the human operator to receive physical feedback from the virtual simulated robot which, due to limitations in manufacturing a feedback method, has been deemed less accurate than modeling the patient in Simulink. Further, by having the human element be simulated, physical constraints can be added to it such as limitations on the control forces they can generate which allows the model to analyse a form of patient with motor impairment based on a healthy subject's data. While access to real patients with motor impairment or motor control difficulty was not possible due to the quarantine situation, this allowed the author to test the capabilities of the proposed control method for a simulated user with motor impairment.

In order to have the simulated human model follow a path equivalent to the physical trajectory corresponding to the collected sEMG data, the human model reads pre-recorded IMU data collected from the real human going through the tests. PID velocity control drives this model to follow the recorded trajectory from the IMU. This trajectory is considered as the true desired trajectory of the user, which the model of the subject with motor impairment aims to traverse. In order to simulate disabilities on a subject with motor impairment, limitations are introduced to this controller in the form of saturation limits on the control command that is allowed to generate.

The parameters used in each simulation block is detailed in Table. 6.1.

Table 6.1: Designed and tuned parameters of simulation

Parameters	Description	Value along (x, y, θ)	unit
$K_{p_{init}}$	Proportional gain on IMU trajectory	[2500 2500 50]	-
$K_{d_{init}}$	Derivative gain on IMU trajectory	[50 50 1]	-
$K_{i_{init}}$	Integral gain on IMU trajectory	[800 800 100]	-
M_{vir}	Apparent Mass	[15 15 0.15]	Kg
B_{vir}	Apparent Damping	[75 75 0.75]	Ns/m
K_p	Proportional gain of admittance control	[20000 20000 8000]	-
K_d	Derivative gain of admittance control	[20000 20000 8000]	-
Offset	Offset between robot and human model	[0 50 0]	cm
Safety threshold	Contact Model's allowable distance	[0 25 0]	cm

6.4 Methods for Augmenting Admittance Control

In augmented admittance control of robot manipulators, the parameters of the controller can be adapted based upon the intention detection system in one of following three manners:

- **Velocity Adapted (VA):** in which a desired velocity in the form of (\dot{x}_d) is fed to the admittance control law (formulated in Eq. 6.6) in the direction of intended motion [121]. Before the onset of the motion, the augmented controller tries to direct the robot toward a trajectory corresponding to the detected intention of initiation of motion. Thus, the robot leads the human through motion in this phase. During the motion, the detection system provides additional support by foreseeing the intended trajectory. In the case of the designed simulation for this research, the magnitude of the predefined desired velocities (\dot{x}_d) along the x, y, θ axes are set to be the average of the linear/angular velocity of each participant going through the tests.
- **Force Adapted (FA):** in which an extra force (F_d) along with user's force command is inputted to the admittance controller (formulated in Eq. 6.6) toward the detected intended direction. Proportional EMG-based controllers have been used to generate assistive torque/force for powered exoskeletons according to the human's intention prior to the movement [122]. For this study, once the intention of the initiation of the motion is detected a predefined force is going to be applied to the interface of admittance controller on behalf of the user. During the motion, the detection system provides additional support by adding the same predefined force in correspondence to the detected intended direction. In the case of the designed simulation for this research, the magnitude of the predefined desired force (F_d) applied along the x, y, θ axes are set

to be proportional to the velocity inputs used in VA methodology.

- **Impedance Adapted (IA):** in which the apparent impedance of the robot (formulated and modified in Eqs. 6.6 and 6.9) is tuned as felt by the user in the direction of the detected intended movement [16, 123]. This would ease the robot's operation for the user, enhance their comfort level in handling the robot and assist them towards following a therapeutic trajectory that can work towards their rehabilitation. In the case of the designed simulation for this research, the impedance parameters (M_d, B_d) of the admittance controller are decreased along the detected direction using the ratio of 0.33 and 0.66, respectively.

6.5 Tests

The participant goes through two collective tests for a total duration of 25 seconds, which is comprised of a combination of movements similar to what was taught to the intention detection algorithm. These tests are designed to evaluate the assisting effects of the admittance control in the motion. The designed path for each test include:

- **Test1 (Lefties)** Attempt to move forward - Move forward - Rest - Attempt to rotate left - Rotate Left - Rest - Attempt to move left - Move left
- **Test2 (Righties)** Attempt to move right - Move right - Rest - Attempt to rotate right - Rotate right - Rest - Attempt to move backward - Move backward

These tests shown in Fig. 6.4, cover all the trained Dynamic and Attempted Motion scenarios known to the intention detection algorithm (scenarios described in Section

3.1.1, and models trained in Section 5.4) that are applicable to SoloWalk. Therefore, provide an evaluation of its performance in all the studied movements.

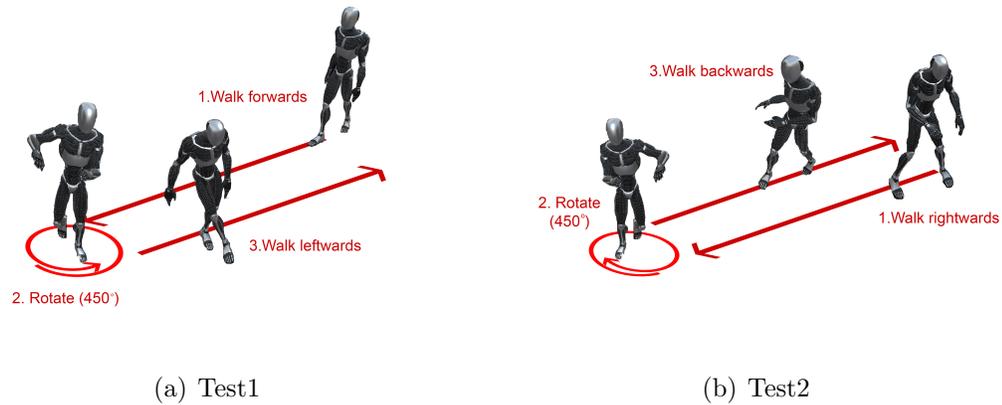


Figure 6.4: The predefined trajectory of designed tests

To record the position of the participant and their trajectory throughout the experiment, IMU sensors are considered since they are light and portable sensors, and they can be quickly set up. Although the IMU sensors are readily usable for our case, they are susceptible to various types of errors. Further, since they are also exposed to noise, both the initial error and the noise are amplified in the form of a growing drift in time because of the double-integration to calculate the position. Therefore, a rigorous calibration procedure is required to capture and to the best possible degree compensate for the initial modeled errors. A dead reckoning method is utilized to correct the integration error and lower the drift. The primary purpose of integrating the IMU sensor in this experiment is to bring humans data into the simulation. Therefore, the ultimate goal is to calculate an approximate position of the user for a short period of time (25 seconds). Thus, calibration and identifying the internal and external parameters of the IMU error model is critical. The utilized IMU calibration procedure, dead reckoning method for integration-error correctness and fusing method to calculate the corresponding acceleration and orientation are brought in Sections C.1, and C.2 of Appendix C, respectively.

In order to evaluate the accuracy of the model and the imposed impairment, and the extent of effectiveness of the admittance controller, and the improvement provided by the augmentation of the controller, the simulation runs in the following Simulation Modes (SMs):

- **SM1 (Human Model Trajectory Following):** The healthy human model is set to follow the real trajectory read through IMU (hereafter called the desired position/trajectory) without interacting with the robot. The contact model is turned off for this SM. By careful tuning of the PID parameters, the human model is able to follow the real trajectory with a very low tracking error, recreating the subject's motion reliably.
- **SM2 (Impaired-Human Model Trajectory Following):** Constraints are added to the human model's controller to simulate the impairment of the patient. By saturating the control command, the model is no more capable of perfectly following the corresponding real trajectory and lags behind it with a certain distance. (not tuned PID model would also result in imperfect path-following but would less accurately capture the behaviour of a subject with motor impairment.)
- **SM3 (Passive Robot Model):** The contact model is turned on, making the human model have to push/drag the robot with it through its trajectory. The admittance control and assistance is kept off. At this SM the test corresponds to a patient with motor impairment trying to go through a desired path while dragging an uncooperative passive robot with itself. As expected, this further makes it lag behind the desired trajectory and renders it less able to follow its intended path.

From this point onward, the goal of the added control elements is to reverse the

negative path-tracking effects of robots stiffness and its opposition to the motion and overcome the limitations introduced due to impairment. In more rigorous terms, the next elements aim to enable the human model to track the desired trajectory with a lower error and bring its trajectory-following capabilities as close to the first SM as possible.

- **SM4 (Active Robot Assistance):** The robot is made an active component by turning the admittance control on. Based on the felt force at the contact between the human and the robot models, the admittance controller will exert a control command to the robot that aims to assist the human in reaching the reference position as guessed by the admittance controller based on the contact forces. This SM makes the robot an active contributor to the walking procedure and, by making the robot move in agreement to the human, lessens the distance by which the user lags behind its intended trajectory. This allows the user to overcome the obstacles it faced from the robot stiffness and the impairment.
- **SM5 (Intention Augmented Control):** The admittance controller is now adapted based on the (now turned on) intention detection block's generated predictions from the recorded EMG data which correspond to the same desired trajectory that the human model is trying to follow. This adaptation, as detailed previously in Section 6.4, is done either in the form of VA, FA, IA, or a combination of these. Knowing the intended direction motion from both the contact force and EMG data allows the admittance controller to further lower the unfavorable effects of impairment and robot's stiffness on the human model's trajectory tracking. This makes the lagging drop and drives the trajectory of the model closer to the desired trajectory.

The degree to which the admittance controller and the adapted admittance controller are capable of lowering the path-following error (improve the trajectory-tracking accuracy) is indicative of how effective they are in assisting a patient with motor impairment in going through a certain rehabilitative path, how much they can lower the patients' burden throughout the said processes and how helpful they are in allowing the patients to traverse their desired paths.

6.6 Results

Two healthy participants undergo the two test scenarios (Lefties and Righties) as defined in Section 6.5. The IMU and sEMG data are collected from these participants and pre-processed as described in Chapter 3.2. The parameters of the trained Subject-Adapted DL model are adapted to the new participants' data. The three channels of recorded acceleration from IMU data (acc_x, acc_y, acc_z) are filtered by a high-pass and low-pass Butterworth filter with cutoff frequencies of 0.001Hz and 5Hz, respectively. The filtered data then can be utilized to detect stationary positions via comparing acceleration magnitude to a small threshold (where participants are assumed not moving):

$$\|acc\| = \sqrt{acc_x^2 + acc_y^2 + acc_z^2} \leq \epsilon \quad (6.14)$$

where $\epsilon = 0.05$ is employed as a threshold to reduce the impact of the noise or very small unrelated motion.

The DL-model trained in the Attempted Motion scenario was used when the motion was identified to be in the stationary stage, and the DL-model trained in the Dynamic Position scenario was used in non-stationary stages. The Subject-Adaptive

strategy that was described in Section 5.4 was used to adapt the network parameters to each new user’s data in both DL models. Moreover, to improve classification accuracy, Majority Vote post-processing is employed. The Majority Vote uses the current classification result, along with the previous nine classification results (with an analysis window spacing of 25 ms, this corresponds to the classification results within the 250 ms period) and makes a classification decision based on the class that appears most often prioritizing the last windows’ detection. The resulting effect is a smooth operation that removes spurious mis-classification [87].

Three methods of augmentation described in Section 6.4 are employed to enhance the admittance controller’s ability to help the simulated model of each person follow their desired trajectory. Root Mean Square Error (RMSE) is utilized to quantify the error between the desired position as recorded from IMU (x_{imu}) and the human model’s trajectory (x_h) during each step of the simulation (N). The equation used to calculate the RMSE is as follow:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_{imu} - x_h)^2} \quad (6.15)$$

By running the developed simulation environment in each SM of each subject in each test scenario, Figs. 6.5, 6.6, 6.7, and 6.8 are obtained. These figures visualise the trajectory following in each SM for each subject and each test scenario. Table. 6.2 represents the deviation of the human model’s trajectory from the human’s desired position in terms of RMSE in each SM for each subject, and in each test scenario.

Table 6.2: Simulation results with data collected from subjects in each test scenario and each Simulation Modes (SMs)

Subject 1	Test1 (Lefties)			Test2 (Righties)		
SMs	RMSE-X	RMSE-Y	RMSE- θ	RMSE-X	RMSE-Y	RMSE- θ
SM1	0.000874	0.006127	0.034664	0.006358	0.001141	0.012607
SM2	0.000912	0.074852	0.180694	0.090059	0.013337	0.102267
SM3	0.019421	0.138613	0.243335	0.207860	0.063263	0.187688
SM4	0.002789	0.117061	0.206427	0.145484	0.016369	0.135233
SM5 (VA)	0.003095	0.066291	0.185285	0.097847	0.014547	0.121630
SM5 (FA)	0.003167	0.054725	0.152182	0.094222	0.014549	0.112331
SM5 (IA)	0.002733	0.085773	0.192204	0.102947	0.016352	0.120202
SM5 (IA+VA)	0.002819	0.074007	0.184981	0.096961	0.015539	0.115046
SM5 (IA+FA)	0.003215	0.060358	0.142727	0.127940	0.014670	0.108256

Subject 2	Test1 (Lefties)			Test2 (Righties)		
SMs	RMSE-X	RMSE-Y	RMSE- θ	RMSE-X	RMSE-Y	RMSE- θ
SM1	0.000536	0.004508	0.014735	0.010907	0.001403	0.010433
SM2	0.000628	0.045925	0.115759	0.075867	0.005108	0.100801
SM3	0.030430	0.167369	0.210613	0.152440	0.185159	0.202135
SM4	0.003159	0.130931	0.150756	0.109306	0.006144	0.141874
SM5 (VA)	0.004247	0.093705	0.138951	0.092184	0.006783	0.088717
SM5 (FA)	0.004778	0.085357	0.138087	0.090142	0.007088	0.115503
SM5 (IA)	0.002941	0.084445	0.136847	0.089776	0.006141	0.116511
SM5 (IA+VA)	0.003148	0.070581	0.132827	0.085535	0.006282	0.097998
SM5 (IA+FA)	0.004623	0.063194	0.144755	0.086702	0.007200	0.152743

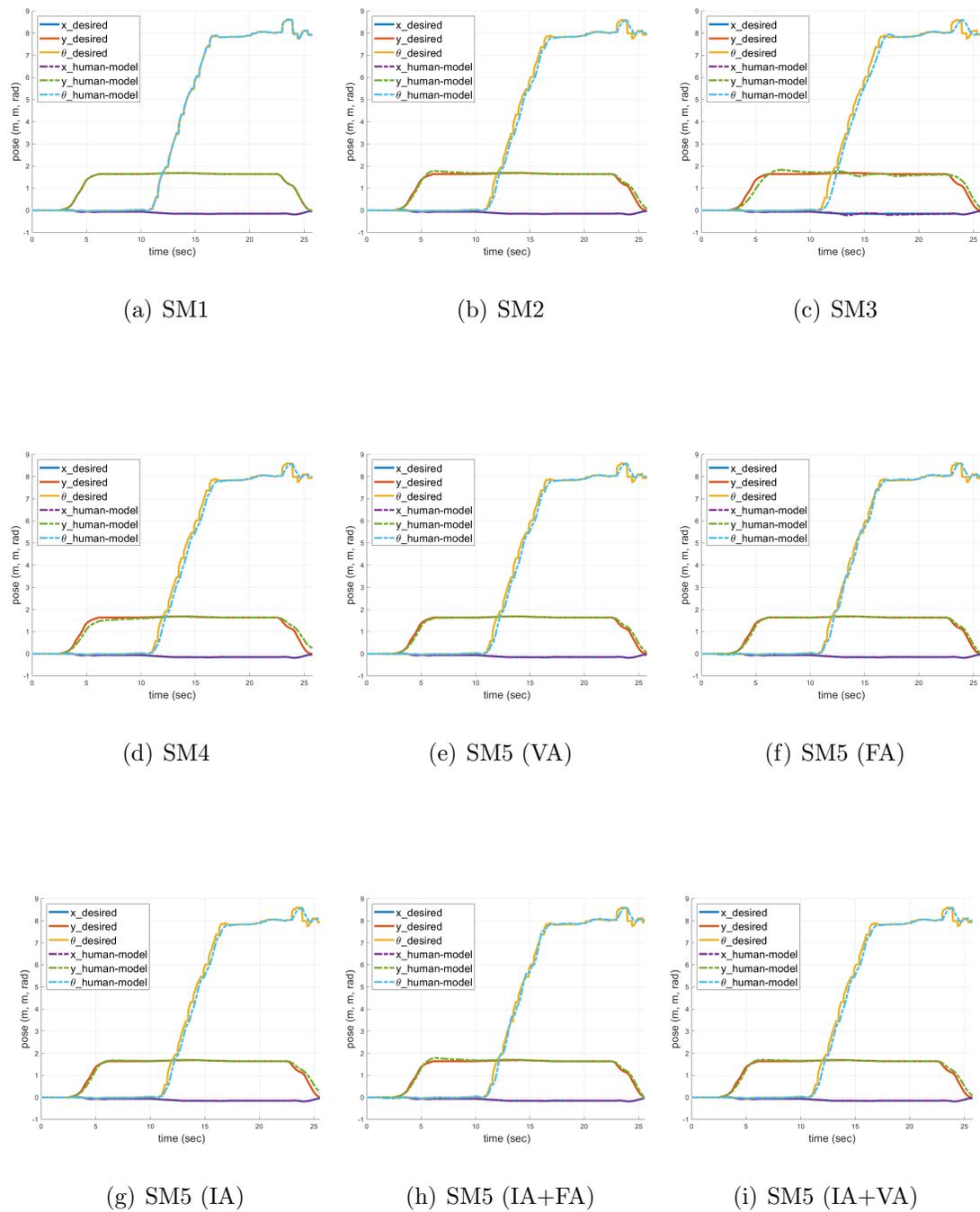


Figure 6.5: Comparison of the desired position and human model's position of Subject 1 while performing the Test1 (Lefties) in each SM.

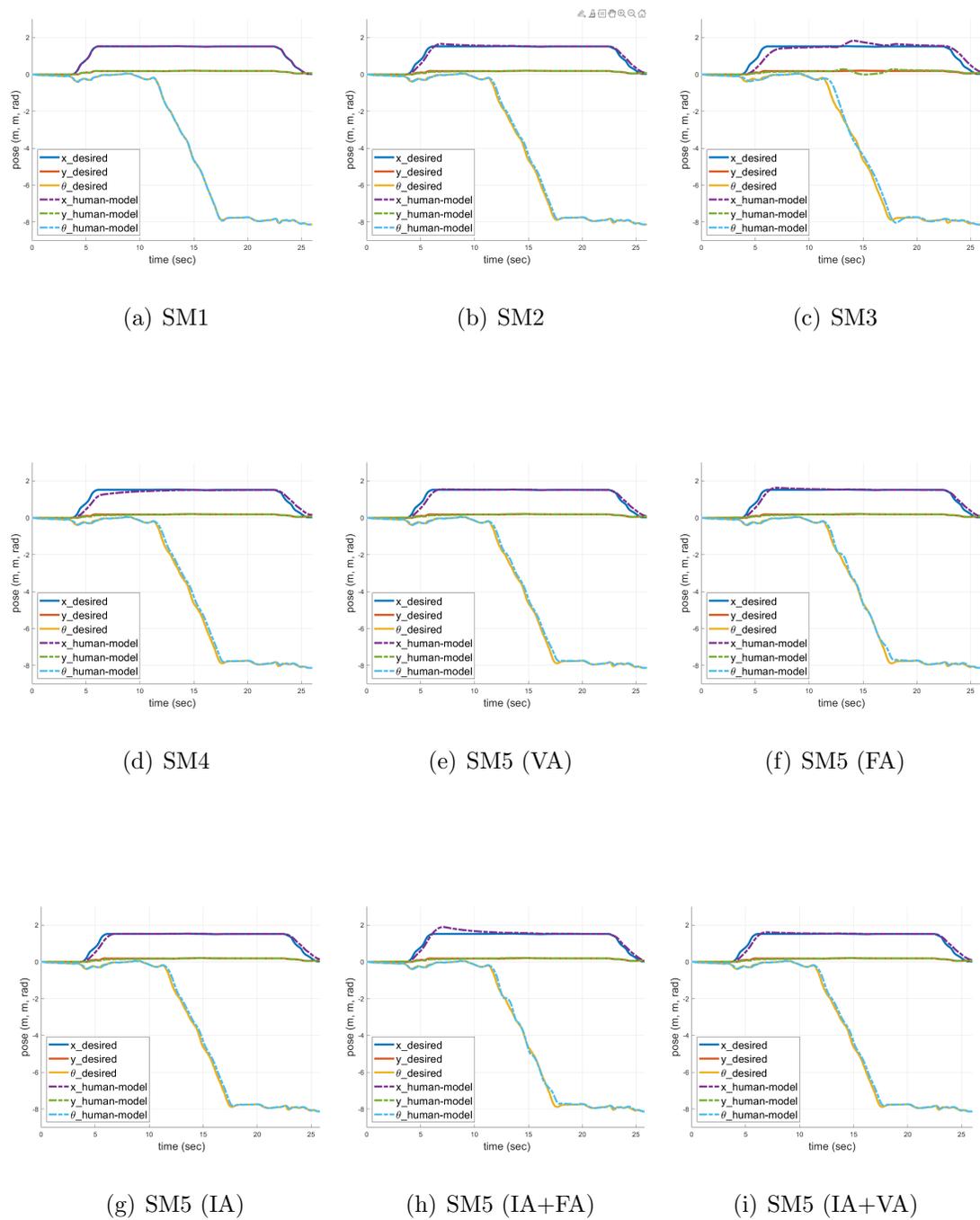


Figure 6.6: Comparison of the desired position and human model's position of Subject 1 while performing the Test2 (Righties) in each SM.

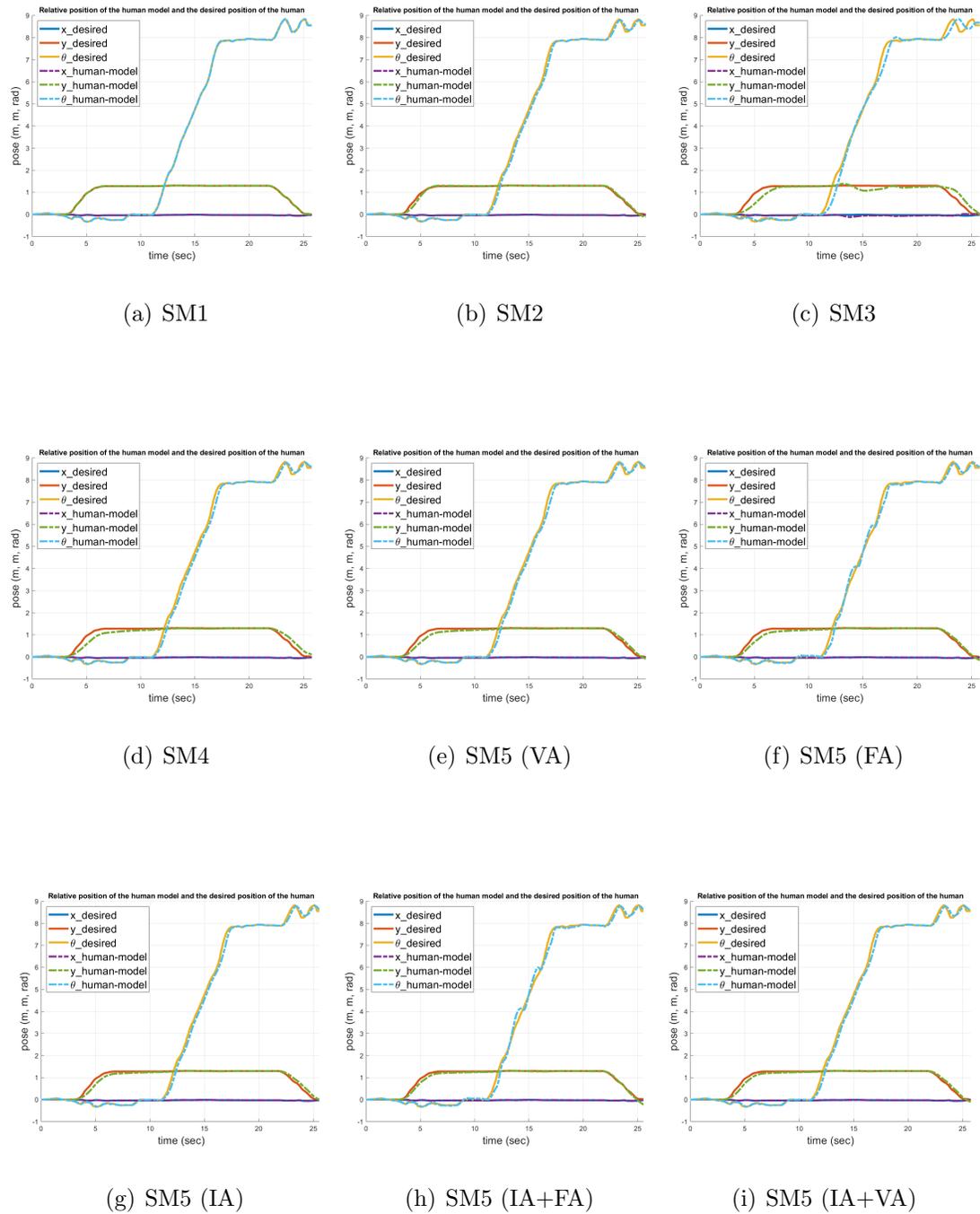


Figure 6.7: Comparison of the desired position and human model's position of Subject 2 while performing the Test1 (Lefties) in each SM.

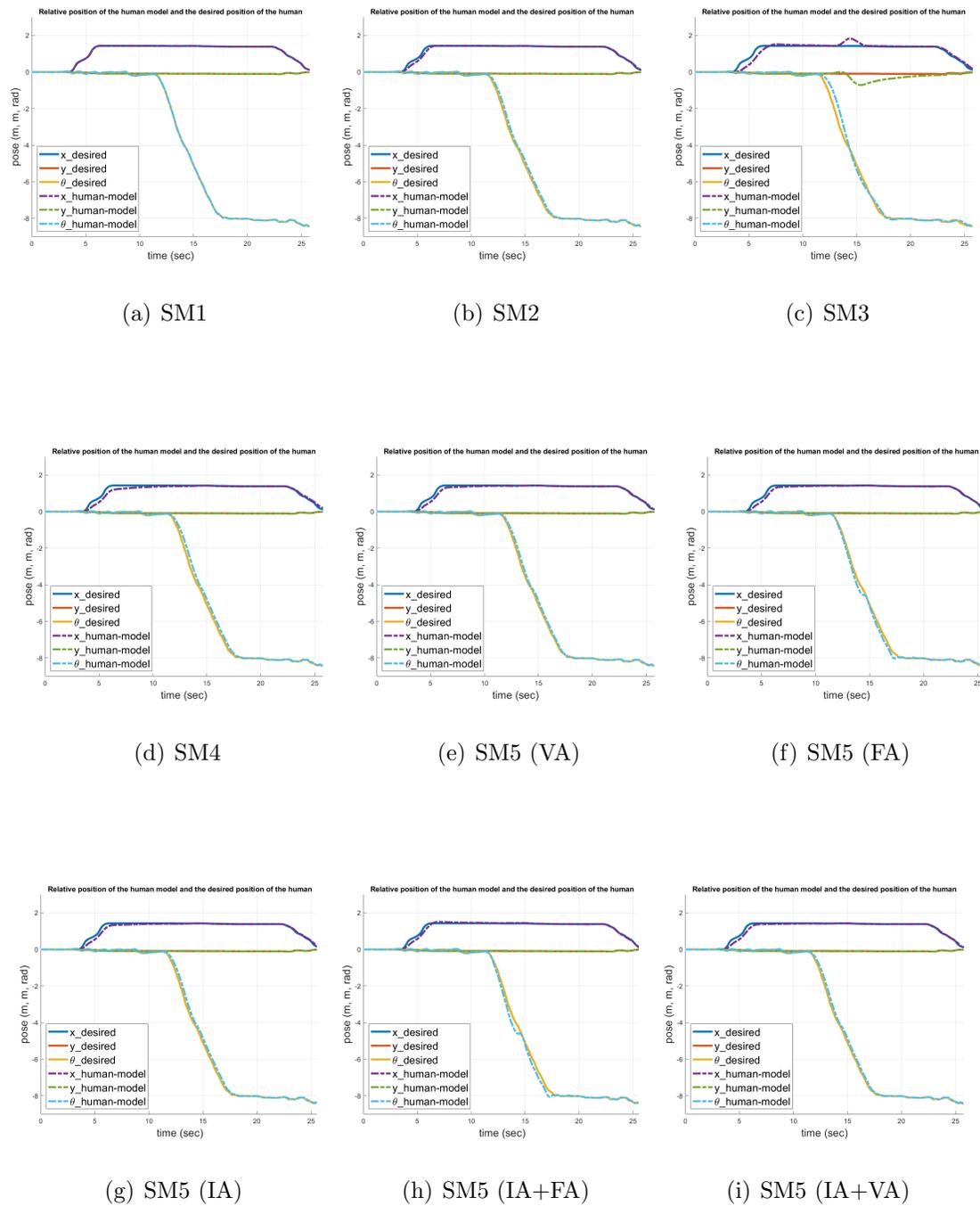


Figure 6.8: Comparison of the desired position and human model's position of Subject 2 while performing the Test2 (Righties) in each SM.

6.7 Discussion

The obtained results from both subjects in SM1 to SM4 of both test scenarios showed a very similar trend in terms of their effect on the deviation of the human model's position from the desired position (recorded from IMU), as can be seen through Table. 6.2 and Figs. 6.5, 6.6, 6.7, and 6.8. Therefore, as any analysis on the response of the system in one test can be extended to other test scenarios and subjects, in the initial part of the discussion section, the SM 1-4 and their effects on the trajectory are explained by studying a sample of the tests on one of the subjects (subject 1 in test 1 (Lefties)). In the second part of the discussion, a thorough comparison between different augmentation methodologies across all the tests and subjects is presented.

It is worth clarifying that among three inertial axes of movement in each designed test scenario, two of them experience higher magnitude of motion, and one of them is less involved in the motion (the axis that experience minimal displacement in each test). The two axes that are experiencing higher motion are hereafter called "dominant" axes and the axis that is less involved is called "non-dominant" axis. Dominant axes are Y and θ axes for test1 (Lefties) scenario, and X and θ axes for test2 (Righties) scenario.

6.7.1 Observations over different Simulation Modes (SM1-4)

Fig. 6.5 shows the trajectory following in each SM of the first subjects in the test1 (Lefties). Table. 6.2 represents the deviation of the human model's trajectory from the human's desired position in terms of RMSE. As it can be seen from both table and figure, in **SM1** without the effect of impairment, dynamics of robot, and any type of active control, the PID velocity control loop of the human model is properly

tuned to accurately drive the human model to follow the desired trajectory of the healthy human as recorded by the IMU (tuned gains are available in Table. 6.1). In **SM2**, the effect of impairment makes the simulated human model lag behind the desired trajectory recorded by the IMU from the subject. This lagging behind the desired trajectory is more noticeable along the axes that the movement is mainly about (dominant axes). In **SM3**, for the purpose of therapy and appropriate training, a basically simplified model of SoloWalk gait rehabilitation robot is added that gets into contact with the human model through a contact model. Since the robot's admittance controller is kept off, the robot is still a passive component that has to be dragged by the human model due to the presence of the contact model. Thus, as it could be expected, the opposing dynamics of the robot against the model with motor impairment result in a drastic deviation in the human model's trajectory from the desired trajectory. As it can be seen in Table. 6.2, the RMSE increases in SM2 due to the addition of the disability model and in SM3 due to the imposed dynamic of the robot. By activating the admittance controller in **SM4**, the burden imposed by the robot dynamics can be compensated while assisting the human model in following the desired trajectory by reducing the apparent dynamics of the robot as felt by the human. However, the robot's admittance controller alone is still not fully capable of compensating the impairment effect and any lag added to the system not showing in the form of force input, because as can be seen in Table. 6.2 or Fig. 6.5, it cannot bring the trajectory tracking error to the level of SM1.

6.7.2 A comparison over different augmentation methodologies; Simulation Modes (SM4-5)

As seen in Table. 6.2 in terms of RMSE, once the admittance controller was added in **SM4** the deviation from the desired trajectory is reduced. By adding the intention detection system in **SM5** the error reduces even further, especially toward the dominant axes. As expected, the motion along these dominant axes are more affected by the addition of the impairment modeling and the resulting lag (i.e. Y and θ axes in test1 (Lefties) scenario). Therefore, the contribution of the intention detection system and the admittance controller is most noticeable in these dominant directions, as can be observed in both tests of both subjects. In all the tests, by augmenting the admittance through all three VA, FA, and IA methodologies in SM5, a noticeable decrease of deviation along dominant axes is observable in Table. 6.2. Among these three methods, VA and FA methodologies, if meticulously tuned, were better capable of reducing the RMSE along the dominant axis in comparison to IA method. One of the reasons for this is that by using VA and FA methodologies, any attempt to move that was detected by the intention detection system can be directly utilized to provide assistance to the human model. Thus, prior to the initiation of the motion from the human side, just by detecting the human's attempt, the robot would guide the human toward the detected direction. This results in faster rise time and better assistance in case of true detection. Nonetheless, any mis-detection is reflected as a velocity or force along the misidentified axis, which the robot would drive the human model toward and cause a deviation from the desired trajectory. The effect of mis-detection can be observed by paying attention to the RMSE along the non-dominant axis. Since there is less movement along the non-dominant axis, the RMSE along this axis was already quite small in SM4 of all the tests and subjects. When moved to SM5, the effect of any mis-detection of the intention detection system would be more

perceptible along this non-dominant axis. Thus, an occasional increase of RMSE along the non-dominant axis can be expected in case of mis-detection. Table. 6.2 shows an occasional increase of RMSE along the non-dominant axes when the robot is augmented by either VA or FA methodologies, which is an indicator of the false detection.

On the other hand, although AI method was less capable of reducing the RMSE along the dominant axes, it was also less affected by mis-detection errors in comparison to VA and FA methodologies. This is due to the fact that a wrong detection would only make the robot more susceptible to any movements along the mis-detected axis in this method, and it would not try to forcefully drive the robot and consequently guide the human toward the detected direction unless a motion toward that direction is initiated by the human themselves. Thus, IA is less affected by false predictions, and the augmented admittance controller would be able to decrease the RMSE along all axes. Yet, due to the same effect, IA is not capable of initiating the motion on the subject's behalf, contrary to VA and FA.

Since one of the main objectives of this thesis was to help the subjects with motor impairment during the initiation of the motion phase, two additional augmentation methods were considered that were built using a combination of the IA with VA and FA methodologies. The results are presented in Table. 6.2 and Figs. 6.5, 6.6, 6.7, and 6.8. The main goal was to use the VA or FA's ability to further reduce the impairment effect and provide assistance during the initiation of motion phase, and to use the IA methodology to reduce the effect of false detection and smooth the control input to output translation. As it can be seen in Table. 6.2, IA+FA performed better in comparison to IA+VA in terms of the RMSE reduction ratio along the dominant axes. However, the mis-detection error has a noticeable higher effect along the non-dominant axis, as can be seen among all tests and all subjects. Although

IA+VA methodology could not reduce the RMSE along the dominant axes as much as IA+FA could, it also had less impact on the non-dominant axes. The reason behind this behaviour can be seen by taking a closer look at the dynamic equation of the admittance control (Eq. 6.6). Since Impedance is the relation between an input force and an output acceleration, lowering the impedance will result in the same input force causing significantly larger accelerations. Velocity Input, on the other hand, bypasses the apparent dynamics equations and is directly exerted to the controller as a desired trajectory. Therefore, it is not amplified by lowering the impedance and works in parallel to it. This effect is also perceivable by comparing the trajectory following when the controller is individually augmented by VA to that of FA method, and by also comparing the results of augmenting by IA+VA to IA+FA, in both tests of both subject as it is illustrated in Figs. 6.5, 6.6, 6.7, and 6.8.

During the tuning phase, it has been observed that the performance of the controller when augmented by either VA or FA methodology, is highly dependant on the predefined input velocity (x_d) or input force (F_d), respectively. Since the study at hand was intended as a feasibility investigation for augmenting the control of assistive robotic systems with an intention detection system, predefined velocities (based on the average velocity of each subject) and forces (proportional to velocity) were utilized in the augmented admittance control. However, for future studies, it is proposed to investigate and implement a more sophisticated and intelligent procedure to estimate and tune the input velocity or input force for more proper integration of VA or FA methodologies in admittance controller.

In order to ensure safety, first of all, Majority Vote post-processing step was used that removes spurious mis-classification to smooth operation. As observed in the simulation results, even when there are false predictions with intention detection system, because these falses are usually not consistent, they get outvoted by the

rest of the ten predictions used by the majority vote. Therefore, minimizing the negative impact of the false predictions. Further, it has been observed that the predictions have more noticeable negative impact when applied through FA method. The most safe options (IA and VA+IA respectively) are therefore chosen as the best possible options for integration of intelligent direction intention detection system to the admittance controller. Some other proposals are as follows: Properly modulating the gain through which the assistance of intention detection system is added to the admittance controller can improve the safety of the device. Since each model at the end produces an estimate of the probability of each class that represent a particular task (The class with the highest score is considered the most probable one), one of the proposals here is to use this confidence estimate as a measure of the gain. Choosing this gain proportional to confidence level of detection system, results in a lower impact by the controller when the predictions are less accurate or mis-classified.

In case of Solowalk, the worst case scenario might happen when the detected direction is opposite of the true intended direction, although the probability of this scenario is very low because the muscles should generate completely conjugate signals that should be detectable by intelligent intention detection system, the proposal is to brute force high-level safety precautions that compare the detected direction commands with the provided force commands from the user-robot interface (or any other communication sensors). In case of any contrast between these two commands, the controller can provide emergency stops to avoid any dangers.

6.8 Chapter Summary

A simulation environment was developed that used the healthy subjects' data (IMU and sEMG) as the reference desired trajectory and the indicator of the users' intentions, respectively. In this simulation a basic model of gait training robot with characteristics of SoloWalk was modeled and three different methodologies (VA, FA, and IA) plus two combination of these methodologies (IA+VA and IA+FA) for augmenting the admittance controller based on the direction detection system were investigated and evaluated. Although, VA and FA methodologies, if meticulously tuned, were better capable of reducing the RMSE along the dominant axis in comparison to IA method, IA was less or almost not affected by the false predictions of the intention detection system. Overall, it is possible to conclude that, IA methodology is the most dependable methodology to be used as the augmentation strategy specifically when the intention detection system's performance is not very reliable. However, since IA is not capable of initiating the motion on the subject's behalf by itself, it is less viable to assist in the initiation of the motion phase compared to VA and FA methodologies. Thus, IA+VA and IA+FA methodologies are introduced. Based on the results, IA+VA has shown the best performance in terms of reducing the RMSE and smoothing the trajectory following behaviour, while also being less impacted by false detection of the intention detection system compared to VA, FA, and IA+FA methodologies.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

With the overarching goal of providing additional support for walking during the therapy and rehabilitation of people with disabilities, particularly subjects diagnosed with cerebral palsy (CP), this study developed a direction detecting system and investigated its applicability for augmenting an HRI system. sEMG signals that convey muscle activity are collected from eight muscles of nine subjects during five trials of at least nine distinct gestures/activities in one knee/foot-gesture (Sitting Position) and three walking-related scenarios (Standing Position, Attempted Motion, and Dynamic Motion). The features of properly pre-processed sEMG signals are then extracted either by Feature Engineering or Feature Learning methodology in C-ML and DL frameworks, respectively. The inter-subject robustness and models' adaptation capability are evaluated by employing these learning frameworks in various implementation strategies. Among C-ML models, the LDA classifier in the Sitting position scenario (with more than 90% accuracy) was demonstrated to be able to reliably facilitate foot-related intelligent assistive devices and joystick-type controllers. The overall

prediction accuracy among walking-related scenarios, though not as remarkably high as in the Sitting Position scenario, could reach up to 75%.

CNN model of the DL approach was able to achieve better detection accuracy than the C-ML approach when implemented either in the Subject-Exclusive or the Generalized strategy. The specifically designed CNN model combined with AdaBN resulted in an advanced transfer learning method which had the ability to build upon previous participants' data and adapt to the new users' specific features to learn their specific manner of performing a particular task. This came with the final goal of enhancing the detection accuracy and alleviating the data generation burden imposed on a single individual. This methodology showed better detection accuracy in Sitting Position and Dynamic Motion scenarios, and it is expected to perform better as the pre-training data set enlarges due to the essential need of DL methods for more data. On a different note, due to the complex nature of the Attempted Motion scenario, it showed better prediction accuracy when trained in a Subject-Exclusive manner either when C-ML approach or DL approach were employed. This is attributed to the varying and disparate gestures performed by different subjects in the Attempted Motion scenario. Therefore, the reasonable assumption is that, Subject-Exclusive implementation strategy should perform better when employed either by C-ML or DL approaches during a walk-direction learning task on the target population (people with motor impairment). This is because they might possess distinct postures and perform tasks in a very dissimilar manner to other people with motor impairment.

Finally, a simulation environment was developed that used the healthy subjects' data (IMU and sEMG) as the reference desired trajectory and the indicator of the users' intentions, respectively. Within the developed simulation environment a model of gait training robot with the characteristics of SoloWalk that interacts with an impaired human model were used. Three different methodologies (VA, FA, and IA) plus

two combinations of these methodologies (IA+VA and IA+FA) were investigated and evaluated for augmenting the robot's admittance controller based on the direction detection system. Although, VA and FA methodologies, if meticulously tuned, showed better performance in reducing the RMSE along the dominant axis in comparison to IA method, these methodologies produced undesirable motion in case of wrong intention detection along the non-dominant axis. IA was less or almost not affected by the false predictions of the intention detection system. Therefore, IA was identified as the most dependable methodology to be used as the augmentation strategy particularly if the intention detection system's performance is not very reliable. Yet, since IA is not capable of initiating the motion on the subject's behalf, it is less viable to assist in the initiation of the motion phase compared to VA and FA. Because providing assistance to initiate the motion was one of the main objectives of this thesis, a combination of AI with VA and FA methods are investigated to reduce the effect of false detection and to provide assistance during the initiation of the motion phase. Based on the results, IA+VA has shown the best performance in terms of reducing the RMSE, while not causing fluctuations in the trajectory and being less impacted by the false detection of the intention detection system compared to VA, FA, and IA+FA methodologies.

Overall, it has been shown that the proposed DL algorithm (either employed in Subject-Exclusive or Subject-adaptive method) is accurate and robust enough to handle a complex, three-dimensional task to increase the cooperation between the robot and the human. As a result, the augmented admittance control enhances the response time and the accuracy of trajectory following.

7.2 Recommendations and Future Work

To be employed as the human intention detection system for Robotic-Assisted Walking, a couple of more investigations are required to ensure the developed methodology's robustness and functionality under different circumstances. The following list is suggested to further advance this line of research:

- Adding Hip muscles to the list of monitored muscles, or fuse other types of sensors like EEG, IMU, force underneath the feet or Joint kinematics to enrich the learning models for intention detection system.
- Increasing the number of subjects during training and testing in order to create a more robust model with a more reliable detection accuracy that is immune to the stochastic characteristics of the learning models.
- Due to the observed close correlation between some gestures (motions of the joints are naturally coupled), it is suggested that in future work a platform for a more isolated joint motion be developed that can decouple joint motions from each other for more exclusive feature extraction and class recognition.
- Collecting data from subjects with some level of disability or individuals with CP and evaluating the performance of each intention detection model. There are several challenges foreseen for application to real subjects with CP. Since CP is a neurological disorder, the main cause of the problem starts in nervous system. Thus, the main challenge in detecting users intentions is to properly separate the sEMG signals that are relevant to intended direction from irrelevant sEMG signals (due to Dystonia, tone abnormalities, spasm, tremor in different muscles, weakness of muscles, etc.). Therefore, it is expected that when this system is being applied on target population, more diligent and rigorous pre-processing

steps be required, or more complex and deep NNs be needed to increase the feature extraction capabilities. It is expected to have Subject-Exclusive implementation strategy outperform the other strategies because of the diversity of neurological effects which are specific to each patient. Subject-adaptability is expected to be an important and strong feature of this application to be investigated with target population.

- Customizing the developed intelligent intention detection system to support individuals with various types of CP, e.g. enhance HRI for individuals with Monoplegia-CP affecting their upper limb. This system can give them the ability to imply their intentions using their lower limb, which they most probably prefer to use. Moreover, since the intention detection system can be trained for each leg separately, in the case of individuals with Hemiplegia-CP, two different models can be generated to detect the spastic muscle activity patterns and take actions accordingly (e.g. lowering SoloWalk's induced motion when spastic motion is detected).
- Evaluating the developed intelligent intention detection system to benefit other groups of patients. For instance, (i) patients diagnosed with stroke who need rehabilitation from an early stage of diagnosis to regain autonomy, (ii) patients with geriatrics who need regular rehabilitation to reduce muscle atrophy and other body systems complications, and (iii) patients with Multiple Sclerosis (MS).
- Evaluating the incorporation of recurrent neural network (RNN) or Long Short Term Memory networks (LSTM) as an algorithm of intention detection system.
- The control chapter of this study is a proof of concept of the applicability of augmenting an admittance controller with an intention detection system. A

more thorough investigation is proposed in order to arrive at the proper manner of manipulating the admittance controller's parameters for the IA methodology. In addition, a more sophisticated and intelligent procedure is suggested to be investigated to estimate and tune the input velocity or input force for more proper integration of VA or FA methodologies in admittance controller.

- Modulating the augmentation gain (through which the assistance of the intention detection system is added to the admittance controller) based on the confidence level of the detection system. This can result in a lower impact by the augmented controller when the probability of a prediction is low and the system is prone to mis-classification.
- Enforcing high-level safety precautions that compare the detected direction commands with the provided force commands from the user-robot interface of the SoloWalk (or any other communication sensors). In case of any contrast between these two commands, the controller can provide emergency stops to avoid any dangers.
- Developing a real-time experiment that might highlight some unforeseen effects related to the human-robot interaction and provide more qualitative performance measurements in terms of ease of use, comfort level, and impact level from the users' perspective.
- Utilizing sEMG data to calculate the users' fatigue level or progress level to adeptly change the control systems' parameters to provide better and more intuitive assistance.
- Utilizing the trained intention detection system in an exoskeleton type robot in parallel to the SoloWalk to provide further assistance to the user in initiation and continuation of motion.

List of References

- [1] A. Meyer-Heim and H. J. van Hedel, “Robot-assisted and computer-enhanced therapies for children with cerebral palsy: current state and clinical implementation,” in *Seminars in pediatric neurology*, vol. 20, pp. 139–145, Elsevier, 2013.
- [2] S. Morris, G. Fawcett, L. Brisebois, and J. Hughes, “A demographic, employment and income profile of Canadians with disabilities aged 15 years and over,” no. 89, pp. 1–25, 2017.
- [3] D. L. Damiano, “Activity, activity, activity: rethinking our physical therapy approach to cerebral palsy,” *Physical therapy*, vol. 86, no. 11, pp. 1534–1540, 2006.
- [4] A. Mohebbi, “Human-robot interaction in rehabilitation and assistance: a review,” *Current Robotics Reports*, pp. 1–14, 2020.
- [5] K. J. Sullivan, D. A. Brown, T. Klassen, S. Mulroy, T. Ge, S. P. Azen, and C. J. Winstein, “Effects of task-specific locomotor and strength training in adults who were ambulatory after stroke: results of the steps randomized clinical trial,” *Physical therapy*, vol. 87, no. 12, pp. 1580–1602, 2007.
- [6] T. Yan, M. Cempini, C. M. Oddo, and N. Vitiello, “Review of assistive strategies in powered lower-limb orthoses and exoskeletons,” *Robotics and Autonomous Systems*, vol. 64, pp. 120–136, 2015.
- [7] M. Qbilat, A. Iglesias, and T. Belpaeme, “A proposal of accessibility guidelines for human-robot interaction,” *Electronics*, vol. 10, no. 5, p. 561, 2021.
- [8] A. A. Morbi, *Design, control, and implementation of a robotic gait rehabilitation system for overground gait training*. PhD thesis, Carleton University, 2014.

- [9] J. O. Wobbrock, S. K. Kane, K. Z. Gajos, S. Harada, and J. Froehlich, “Ability-based design: Concept, principles and examples,” *ACM Transactions on Accessible Computing (TACCESS)*, vol. 3, no. 3, pp. 1–27, 2011.
- [10] N. Amankwah, M. Oskoui, R. Garner, C. Bancej, D. G. Manuel, R. Wall, P. Finès, J. Bernier, K. Tu, and K. Reimer, “Original quantitative research-cerebral palsy in canada, 2011–2031: results of a microsimulation modelling study of epidemiological and cost impacts,” *Health promotion and chronic disease prevention in Canada: research, policy and practice*, vol. 40, no. 2, p. 25, 2020.
- [11] T. M. O’Shea, “Diagnosis, treatment, and prevention of cerebral palsy in near-term/term infants,” *Clinical obstetrics and gynecology*, vol. 51, no. 4, p. 816, 2008.
- [12] C. Bayon, R. Raya, S. L. Lara, O. Ramirez, J. Serrano, and E. Rocon, “Robotic therapies for children with cerebral palsy: a systematic review,” *Transl Biomed*, vol. 7, no. 1, p. 44, 2016.
- [13] N. Hogan, H. I. Krebs, B. Rohrer, J. J. Palazzolo, L. Dipietro, S. E. Fasoli, J. Stein, R. Hughes, W. R. Frontera, D. Lynch, *et al.*, “Motions or muscles? some behavioral factors underlying robotic assistance of motor recovery,” *Journal of rehabilitation research and development*, vol. 43, no. 5, p. 605, 2006.
- [14] Z. Warraich and J. A. Kleim, “Neural plasticity: the biological substrate for neurorehabilitation,” *PM&R*, vol. 2, no. 12, pp. S208–S219, 2010.
- [15] S. Hussain, S. Q. Xie, and P. K. Jamwal, “Adaptive impedance control of a robotic orthosis for gait rehabilitation,” *IEEE transactions on cybernetics*, vol. 43, no. 3, pp. 1025–1034, 2013.
- [16] R. Wu, H. Zhang, T. Peng, L. Fu, and J. Zhao, “Human-robot interaction and demonstration learning mode based on electromyogram signal and variable impedance control,” *Mathematical Problems in Engineering*, vol. 2018, 2018.
- [17] L. Tiberio, A. Cesta, and M. Olivetti Belardinelli, “Psychophysiological methods to evaluate user’s response in human robot interaction: a review and feasibility study,” *Robotics*, vol. 2, no. 2, pp. 92–121, 2013.
- [18] C. Mollaret, A. A. Mekonnen, F. Lerasle, I. Ferrané, J. Pinquier, B. Boudet, and P. Rumeau, “A multi-modal perception based assistive robotic system for

- the elderly,” *Computer Vision and Image Understanding*, vol. 149, pp. 78–97, 2016.
- [19] B. D. Argall and A. G. Billard, “A survey of tactile human–robot interactions,” *Robotics and autonomous systems*, vol. 58, no. 10, pp. 1159–1176, 2010.
- [20] E. J. Rechy-Ramirez and H. Hu, “Bio-signal based control in assistive robots: a survey,” *Digital Communications and networks*, vol. 1, no. 2, pp. 85–101, 2015.
- [21] “Gaittronics inc..” <http://gaittronics.com/solowalk/>. Accessed: 2021-02-17.
- [22] K. Dautenhahn, “Methodology & themes of human-robot interaction: A growing research field,” *International Journal of Advanced Robotic Systems*, vol. 4, no. 1, p. 15, 2007.
- [23] K. W. Ong, G. Seet, and S. K. Sim, “An implementation of seamless human-robot interaction for telerobotics,” *International Journal of Advanced Robotic Systems*, vol. 5, no. 2, p. 18, 2008.
- [24] L. Bi, C. Guan, *et al.*, “A review on emg-based motor intention prediction of continuous human upper limb motion for human-robot collaboration,” *Biomedical Signal Processing and Control*, vol. 51, pp. 113–127, 2019.
- [25] M. Rashid, N. Sulaiman, A. P. A. Majeed, R. M. Musa, A. F. A. Nasir, B. S. Bari, and S. Khatun, “Current status, challenges, and possible solutions of eeg-based brain-computer interface: A comprehensive review,” *Frontiers in neuro-robotics*, vol. 14, 2020.
- [26] S. Polak, Y. Barniv, and Y. Baram, “Head motion anticipation for virtual-environment applications using kinematics and emg energy,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 36, no. 3, pp. 569–576, 2006.
- [27] M. B. I. Reaz, M. S. Hussain, and F. Mohd-Yasin, “Techniques of emg signal analysis: detection, processing, classification and applications,” *Biological procedures online*, vol. 8, no. 1, pp. 11–35, 2006.
- [28] F. Zhang and H. Huang, “Decoding movement intent of patient with multiple sclerosis for the powered lower extremity exoskeleton,” in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 4957–4960, IEEE, 2013.

- [29] N. Nazmi, M. A. Abdul Rahman, S.-I. Yamamoto, S. A. Ahmad, H. Zamzuri, and S. A. Mazlan, "A review of classification techniques of emg signals during isotonic and isometric contractions," *Sensors*, vol. 16, no. 8, p. 1304, 2016.
- [30] K. R. Lyons and S. S. Joshi, "A case study on classification of foot gestures via surface electromyography," in *Proc. Annu. Conf. Rehabil. Eng. Assist. Technol. Soc. Amer.*, pp. 1–5, 2015.
- [31] A. Phinyomark, P. Phukpattaranont, and C. Limsakul, "Feature reduction and selection for EMG signal classification," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7420–7431, 2012.
- [32] B. Hudgins, P. Parker, and R. N. Scott, "A new strategy for multifunction myoelectric control," *IEEE transactions on biomedical engineering*, vol. 40, no. 1, pp. 82–94, 1993.
- [33] K. Englehart, B. Hudgins, P. A. Parker, and M. Stevenson, "Classification of the myoelectric signal using time-frequency based representations," *Medical engineering & physics*, vol. 21, no. 6-7, pp. 431–438, 1999.
- [34] K. Englehart, B. Hudgin, and P. A. Parker, "A wavelet-based continuous classification scheme for multifunction myoelectric control," *IEEE Transactions on Biomedical Engineering*, vol. 48, no. 3, pp. 302–311, 2001.
- [35] K. Englehart and B. Hudgins, "A robust, real-time control scheme for multifunction myoelectric control," *IEEE transactions on biomedical engineering*, vol. 50, no. 7, pp. 848–854, 2003.
- [36] M. Atzori, A. Gijsberts, I. Kuzborskij, S. Elsig, A.-G. M. Hager, O. Deriaz, C. Castellini, H. Müller, and B. Caputo, "Characterization of a benchmark database for myoelectric movement classification," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 1, pp. 73–83, 2014.
- [37] I. Kuzborskij, A. Gijsberts, and B. Caputo, "On the challenge of classifying 52 hand movements from surface electromyography," in *2012 annual international conference of the IEEE engineering in medicine and biology society*, pp. 4931–4937, IEEE, 2012.
- [38] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A.-G. M. Hager, S. Elsig, G. Giatsidis, F. Bassetto, and H. Müller, "Electromyography data for non-invasive naturally-controlled robotic hand prostheses," *Scientific data*, vol. 1, no. 1, pp. 1–13, 2014.

- [39] A. Gijsberts, M. Atzori, C. Castellini, H. Müller, and B. Caputo, “Movement error rate for evaluation of machine learning methods for semg-based hand movement classification,” *IEEE transactions on neural systems and rehabilitation engineering*, vol. 22, no. 4, pp. 735–744, 2014.
- [40] H. T. T. Vu, D. Dong, H.-L. Cao, T. Verstraten, D. Lefeber, B. Vanderborght, and J. Geeroms, “A review of gait phase detection algorithms for lower limb prostheses,” *Sensors*, vol. 20, no. 14, p. 3972, 2020.
- [41] B. Hu, E. Rouse, and L. Hargrove, “Fusion of bilateral lower-limb neuromechanical signals improves prediction of locomotor activities,” *Frontiers in Robotics and AI*, vol. 5, p. 78, 2018.
- [42] S. Bai, J. Z. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” *arXiv preprint arXiv:1803.01271*, 2018.
- [43] K.-H. Park and S.-W. Lee, “Movement intention decoding based on deep learning for multiuser myoelectric interfaces,” in *2016 4th International Winter Conference on Brain-Computer Interface (BCI)*, pp. 1–2, IEEE, 2016.
- [44] K.-T. Kim, K.-H. Park, and S.-W. Lee, “An adaptive convolutional neural network framework for multi-user myoelectric interfaces,” in *Asian Conference on Pattern Recognition (ACPR)*, pp. 788–792, IEEE, 2017.
- [45] M. Atzori, M. Cognolato, and H. Müller, “Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands,” *Frontiers in Neurorobotics*, vol. 10, pp. 1–10, 2016.
- [46] W. Geng, Y. Du, W. Jin, W. Wei, Y. Hu, and J. Li, “Gesture recognition by instantaneous surface EMG images,” *Scientific Reports*, vol. 6, no. 1, pp. 6–13, 2016.
- [47] W. Wei, Y. Wong, Y. Du, Y. Hu, M. Kankanhalli, and W. Geng, “A multi-stream convolutional neural network for semg-based gesture recognition in muscle-computer interface,” *Pattern Recognition Letters*, vol. 119, pp. 131–138, 2019.
- [48] X. Zhai, B. Jelfs, R. H. Chan, and C. Tin, “Self-recalibrating surface emg pattern recognition for neuroprosthesis control based on convolutional neural network,” *Frontiers in neuroscience*, vol. 11, p. 379, 2017.

- [49] Y. Du, W. Jin, W. Wei, Y. Hu, and W. Geng, “Surface EMG-based inter-session gesture recognition enhanced by deep domain adaptation,” *Sensors (Switzerland)*, vol. 17, no. 3, pp. 6–9, 2017.
- [50] Z. Ding, C. Yang, Z. Tian, C. Yi, Y. Fu, and F. Jiang, “semg-based gesture recognition with convolution neural networks,” *Sustainability*, vol. 10, no. 6, p. 1865, 2018.
- [51] H. Chen, Y. Zhang, G. Li, Y. Fang, and H. Liu, “Surface electromyography feature extraction via convolutional neural network,” *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 1, pp. 185–196, 2020.
- [52] U. Côté-Allard, C. L. Fall, A. Drouin, A. Campeau-Lecours, C. Gosselin, K. Glette, F. Laviolette, and B. Gosselin, “Deep learning for electromyographic hand gesture signal classification using transfer learning,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 4, pp. 760–771, 2019.
- [53] N. Hogan, “Impedance control: An approach to manipulation,” in *American Control Conference*, pp. 304–313, IEEE, 1984.
- [54] D. P. Ferris, K. E. Gordon, G. S. Sawicki, and A. Peethambaran, “An improved powered ankle-foot orthosis using proportional myoelectric control,” *Gait & posture*, vol. 23, no. 4, pp. 425–428, 2006.
- [55] L. Finch, H. Barbeau, and B. Arsenault, “Influence of body weight support on normal human gait: development of a gait retraining strategy,” *Physical Therapy*, vol. 71, no. 11, pp. 842–855, 1991.
- [56] G. Colombo, M. Wirz, and V. Dietz, “Driven gait orthosis for improvement of locomotor training in paraplegic patients,” *Spinal cord*, vol. 39, no. 5, pp. 252–255, 2001.
- [57] H. Schmidt, S. Hesse, R. Bernhardt, and J. Krüger, “Hapticwalker—a novel haptic foot device,” *ACM Transactions on Applied Perception (TAP)*, vol. 2, no. 2, pp. 166–180, 2005.
- [58] S. Hesse, A. Waldner, and C. Tomelleri, “Innovative gait robot for the repetitive practice of floor walking and stair climbing up and down in stroke patients,” *Journal of neuroengineering and rehabilitation*, vol. 7, no. 1, p. 30, 2010.

- [59] C. Schmitt and P. Métrailler, “The motion maker™: a rehabilitation system combining an orthosis with closed-loop electrical muscle stimulation,” in *International workshop on functional electrical stimulation*, no. CONF, pp. 117–120, 2004.
- [60] M. Bouri, B. Le Gall, and R. Clavel, “A new concept of parallel robot for rehabilitation and fitness: the lambda,” in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 2503–2508, IEEE, 2009.
- [61] M. Girone, G. Burdea, M. Bouzit, V. Popescu, and J. Deutsch, “Orthopedic rehabilitation using the” rutgers ankle” interface,” *Studies in health technology and informatics*, pp. 89–95, 2000.
- [62] K. J. Chisholm, *Design and control for a gait rehabilitation robot*. PhD thesis, Carleton University, 2010.
- [63] K. Chisholm, K. Klumper, A. Mullins, and M. Ahmadi, “A task oriented haptic gait rehabilitation robot,” *Mechatronics*, vol. 24, no. 8, pp. 1083–1091, 2014.
- [64] I. L. Petersen, W. Nowakowska, C. Ulrich, and L. N. S. A. Struijk, “A novel semg triggered fes-hybrid robotic lower limb rehabilitation system for stroke patients,” *IEEE Transactions on Medical Robotics and Bionics*, pp. 1–1, 2020.
- [65] J. Kathryn and M. Syranosian, “Autoambulator improves functionality for healthsouth’s rehab patients.”
- [66] D. Aoyagi, W. E. Ichinose, S. J. Harkema, D. J. Reinkensmeyer, and J. E. Bobrow, “A robot and control algorithm that can synchronously assist in naturalistic motion during body-weight-supported gait training following neurologic injury,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 3, pp. 387–400, 2007.
- [67] J. F. Veneman, R. Kruidhof, E. E. Hekman, R. Ekkelenkamp, E. H. Van Asseldonk, and H. Van Der Kooij, “Design and evaluation of the lopes exoskeleton robot for interactive gait rehabilitation,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 3, pp. 379–386, 2007.
- [68] S. K. Banala, S. H. Kim, S. K. Agrawal, and J. P. Scholz, “Robot assisted gait training with active leg exoskeleton (alex),” *IEEE transactions on neural systems and rehabilitation engineering*, vol. 17, no. 1, pp. 2–8, 2008.

- [69] S. Lee and Y. Sankai, "Power assist control for leg with hal-3 based on virtual torque and impedance adjustment," in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 6–pp, IEEE, 2002.
- [70] S. M. Taslim Reza, N. Ahmad, I. A. Choudhury, and R. A. R. Ghazilla, "A fuzzy controller for lower limb exoskeletons during sit-to-stand and stand-to-sit movement using wearable sensors," *Sensors*, vol. 14, no. 3, pp. 4342–4363, 2014.
- [71] M. Peshkin, D. A. Brown, J. J. Santos-Munné, A. Makhlin, E. Lewis, J. E. Colgate, J. Patton, and D. Schwandt, "Kineassist: A robotic overground gait and balance training device," in *International Conference on Rehabilitation Robotics*, pp. 241–246, IEEE, 2005.
- [72] Y. Stauffer, Y. Allemand, M. Bouri, J. Fournier, R. Clavel, P. Métrailler, R. Brodard, and F. Reynard, "The walktrainer—a new generation of walking reeducation device combining orthoses and muscle stimulation," *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 17, no. 1, pp. 38–45, 2008.
- [73] C. Bayon, O. Ramírez, J. I. Serrano, M. Del Castillo, A. Pérez-Somarriba, J. M. Belda-Lois, I. Martínez-Caballero, S. Lerma-Lara, C. Cifuentes, A. Frizera, *et al.*, "Development and evaluation of a novel robotic platform for gait rehabilitation in patients with cerebral palsy: Cpwalker," *Robotics and autonomous systems*, vol. 91, pp. 101–114, 2017.
- [74] C. Bayón, T. Martín-Lorenzo, B. Moral-Saiz, Ó. Ramírez, Á. Pérez-Somarriba, S. Lerma-Lara, I. Martínez, and E. Rocon, "A robot-based gait training therapy for pediatric population with cerebral palsy: goal setting, proposal and preliminary clinical implementation," *Journal of neuroengineering and rehabilitation*, vol. 15, no. 1, p. 69, 2018.
- [75] S.-K. Wu, G. Waycaster, and X. Shen, "Electromyography-based control of active above-knee prostheses," *Control Engineering Practice*, vol. 19, no. 8, pp. 875–882, 2011.
- [76] S. Pfeifer, H. Vallery, M. Hardegger, R. Riener, and E. J. Perreault, "Model-based estimation of knee stiffness," *IEEE transactions on biomedical engineering*, vol. 59, no. 9, pp. 2604–2612, 2012.
- [77] H. Huang, T. A. Kuiken, R. D. Lipschutz, *et al.*, "A strategy for identifying locomotion modes using surface electromyography," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 1, pp. 65–73, 2008.

- [78] D. Jin, J. Yang, R. Zhang, R. Wang, and J. Zhang, "Terrain identification for prosthetic knees based on electromyographic signal features," *Tsinghua Science and Technology*, vol. 11, no. 1, pp. 74–79, 2006.
- [79] D. Tkach, R. D. Lipschutz, S. B. Finucane, and L. J. Hargrove, "Myoelectric neural interface enables accurate control of a virtual multiple degree-of-freedom foot-ankle prosthesis," in *International Conference on Rehabilitation Robotics (ICORR)*, pp. 1–4, IEEE, 2013.
- [80] "Neural control of prosthetic." <https://nrel.bme.unc.edu/research/>. Accessed: 2021-02-17.
- [81] C. Morbidoni, A. Cucchiarelli, S. Fioretti, and F. Di Nardo, "A deep learning approach to emg-based classification of gait phases during level ground walking," *Electronics*, vol. 8, no. 8, p. 894, 2019.
- [82] A. Rainoldi, G. Melchiorri, and I. Caruso, "A method for positioning electrodes during surface emg recordings in lower limb muscles," *Journal of neuroscience methods*, vol. 134, no. 1, pp. 37–43, 2004.
- [83] S. Kyeong, W. Shin, M. Yang, U. Heo, J. rou Feng, and J. Kim, "Recognition of walking environments and gait period by surface electromyography," *Frontiers of Information Technology and Electronic Engineering*, vol. 20, no. 3, pp. 342–352, 2019.
- [84] H. J. Hermens, B. Freriks, C. Disselhorst-Klug, and G. Rau, "Development of recommendations for semg sensors and sensor placement procedures," *Journal of Electromyography and Kinesiology*, vol. 10, no. 5, pp. 361 – 374, 2000.
- [85] C. J. De Luca, L. Donald Gilmore, M. Kuznetsov, and S. H. Roy, "Filtering the surface EMG signal: Movement artifact and baseline noise contamination," *Journal of Biomechanics*, vol. 43, no. 8, pp. 1573–1579, 2010.
- [86] S. M. Khan, A. A. Khan, and O. Farooq, "Selection of features and classifiers for EMG-EEG-Based upper limb assistive devices - A review," *IEEE Reviews in Biomedical Engineering*, vol. 13, pp. 248–260, 2020.
- [87] A. D. Chan and G. C. Green, "Myoelectric control development toolbox," *Conference of the Canadian Medical & Biological Engineering Society (CMBES) Proceedings*, vol. 30, 2007.

- [88] L. H. Smith, L. J. Hargrove, B. A. Lock, and T. A. Kuiken, “Determining the optimal window length for pattern recognition-based myoelectric control: Balancing the competing effects of classification error and controller delay,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 19, no. 2, pp. 186–192, 2011.
- [89] R. Menon, G. Di Caterina, H. Lakany, L. Petropoulakis, B. A. Conway, and J. J. Soraghan, “Study on Interaction between Temporal and Spatial Information in Classification of EMG Signals for Myoelectric Prostheses,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 10, pp. 1832–1842, 2017.
- [90] R. Merletti, D. Farina, H. Hermens, B. Freriks, and J. Harlaar, “European recommendations for signal processing methods for surface electromyography,” *European recommendations for surface electromyography. Roessingh Research and Development, Enschede, Netherlands*, pp. 57–70, 1999.
- [91] I.-S. Hwang and L. D. Abraham, “Quantitative emg analysis to investigate synergistic coactivation of ankle and knee muscles during isokinetic ankle movement. part 1: time amplitude analysis,” *Journal of Electromyography and Kinesiology*, vol. 11, no. 5, pp. 319–325, 2001.
- [92] J. Potvin and S. Brown, “Less is more: high pass filtering, to remove up to 99% of the surface emg signal power, improves emg-based biceps brachii muscle force estimates,” *Journal of Electromyography and Kinesiology*, vol. 14, no. 3, pp. 389–399, 2004.
- [93] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer Series in Statistics, New York, NY, USA: Springer New York Inc., 2001.
- [94] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152, 1992.
- [95] A. H. Al-Timemy, R. N. Khushaba, G. Bugmann, and J. Escudero, “Improving the performance against force variation of emg controlled multifunctional upper-limb prostheses for transradial amputees,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 6, pp. 650–661, 2016.
- [96] L. Breiman, “Bagging predictors,” *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

- [97] T. K. Ho, “Random decision forests,” in *international conference on document analysis and recognition*, vol. 1, pp. 278–282, IEEE, 1995.
- [98] M. V. Liarokapis, P. K. Artemiadis, K. J. Kyriakopoulos, and E. S. Manolakos, “A learning scheme for reach to grasp movements: On emg-based interfaces using task specific motion decoding models,” *IEEE journal of biomedical and health informatics*, vol. 17, no. 5, pp. 915–921, 2013.
- [99] G. Song, Y. Wang, M. Wang, and Y. Li, “Lower limb movement intent recognition based on grid search random forest algorithm,” in *Proceedings of the 3rd International Conference on Robotics, Control and Automation*, pp. 225–229, 2018.
- [100] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.
- [101] L. Mason, J. Baxter, P. L. Bartlett, and M. R. Frean, “Boosting algorithms as gradient descent,” in *Advances in neural information processing systems*, pp. 512–518, 2000.
- [102] M. Claesen and B. De Moor, “Hyperparameter search in machine learning,” *arXiv preprint arXiv:1502.02127*, 2015.
- [103] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, “Scikit-learn: Machine learning in python,” *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.
- [104] “Metrics for classification scikit-learn.” https://scikit-learn.org/stable/modules/model_evaluation.html. Accessed: 2021-03-05.
- [105] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, “Efficient processing of deep neural networks: A tutorial and survey,” *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.
- [106] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [107] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

- [108] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*, pp. 448–456, PMLR, 2015.
- [109] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, “Progressive neural networks,” *arXiv preprint arXiv:1606.04671*, 2016.
- [110] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou, “Revisiting batch normalization for practical domain adaptation,” *arXiv preprint arXiv:1603.04779*, 2016.
- [111] J. A. Kleim and T. A. Jones, “Principles of experience-dependent neural plasticity: implications for rehabilitation after brain damage,” *Journal of speech, language, and hearing research*, 2008.
- [112] E. T. Wolbrecht, V. Chan, D. J. Reinkensmeyer, and J. E. Bobrow, “Optimizing compliant, model-based robotic assistance to promote neurorehabilitation,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 16, no. 3, pp. 286–297, 2008.
- [113] M. A. Dimyan and L. G. Cohen, “Neuroplasticity in the context of motor rehabilitation after stroke,” *Nature Reviews Neurology*, vol. 7, no. 2, pp. 76–85, 2011.
- [114] N. Colonnese and A. Okamura, “M-width: stability and accuracy of haptic rendering of virtual mass,” *Robotics*, vol. 41, pp. 41–48, 2013.
- [115] L. Marchal-Crespo and D. J. Reinkensmeyer, “Review of control strategies for robotic movement training after neurologic injury,” *Journal of neuroengineering and rehabilitation*, vol. 6, no. 1, pp. 1–15, 2009.
- [116] C. Ott, R. Mukherjee, and Y. Nakamura, “Unified impedance and admittance control,” in *IEEE International Conference on Robotics and Automation*, pp. 554–561, IEEE, 2010.
- [117] D. A. Lawrence, “Impedance control stability properties in common implementations,” in *IEEE International Conference on Robotics and Automation*, pp. 1185–1190, IEEE, 1988.
- [118] A. Peer and M. Buss, “Robust stability analysis of a bilateral teleoperation system using the parameter space approach,” in *International Conference on Intelligent Robots and Systems*, pp. 2350–2356, IEEE, 2008.

- [119] N. Hogan, “Stable execution of contact tasks using impedance control,” in *IEEE International Conference on Robotics and Automation*, vol. 4, pp. 1047–1054, IEEE, 1987.
- [120] D. E. Whitney, “Force feedback control of manipulator fine motions,” 1977.
- [121] B. Corteville, E. Aertbeliën, H. Bruyninckx, J. De Schutter, and H. Van Brussel, “Human-inspired robot assistant for fast point-to-point movements,” in *IEEE International Conference on Robotics and Automation*, pp. 3639–3644, IEEE, 2007.
- [122] Y. Zhuang, S. Yao, C. Ma, and R. Song, “Admittance control based on emg-driven musculoskeletal model improves the human–robot synchronization,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1211–1218, 2018.
- [123] G. Kang, H. S. Oh, J. K. Seo, U. Kim, and H. R. Choi, “Variable admittance control of robot manipulators based on human intention,” *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 3, pp. 1023–1032, 2019.
- [124] R. Zhang, F. Hoflinger, and L. M. Reind, “Calibration of an imu using 3-d rotation platform,” *IEEE sensors Journal*, vol. 14, no. 6, pp. 1778–1787, 2014.

Appendix A

Experiments

A.1 Gestures Performed during Different Scenarios

Table A.1: List of Gestures and Activities performed in each Scenario

Scenario	Gestures
SittingPosition	Ankle Dorsiflexion, Ankle Plantarflexion, Ankle Inversion, Ankle Eversion, Ankle LateralRotation, Ankle MedialRotation, Knee Abduction, Knee Adduction, Knee Flexion, Knee Extension, Rest
StandingPosition	SideStep, CrossStep, StepForward, StepForward2, StepBackward, StepBackward2, Hip InternalRotation, Hip ExternalRotation, Rest
AttemptedMotion	SideStep, CrossStep, Walk Forward, Walk Backward, Rotate LeftSide, Rotate RightSide, Sit, Stand, Rest
DynamicMotion	SideStep, CrossStep, Walk Forward, Walk Backward, Rotate LeftSide, Rotate RightSide, Sit, Stand, Rest

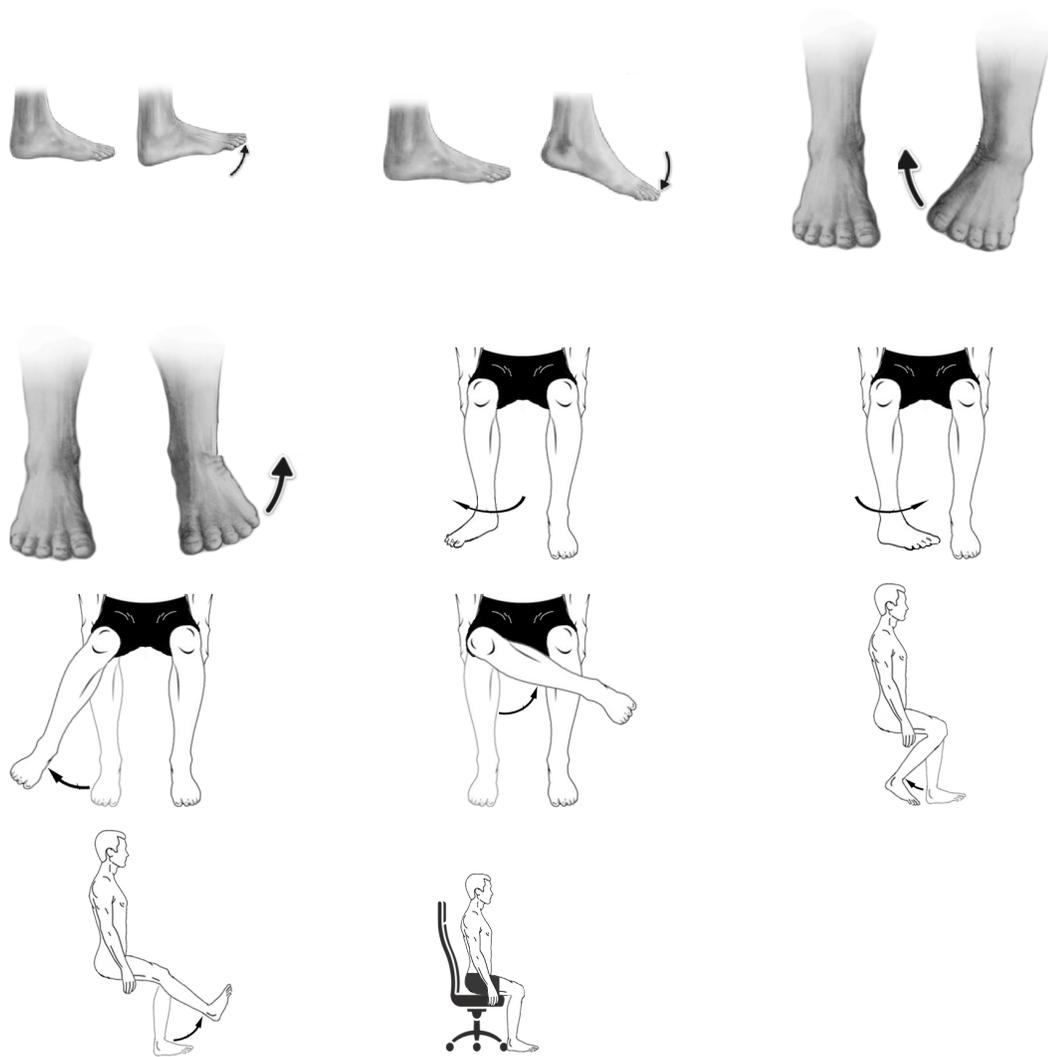


Figure A.1: Static gestures to be performed in Sitting Position Scenario

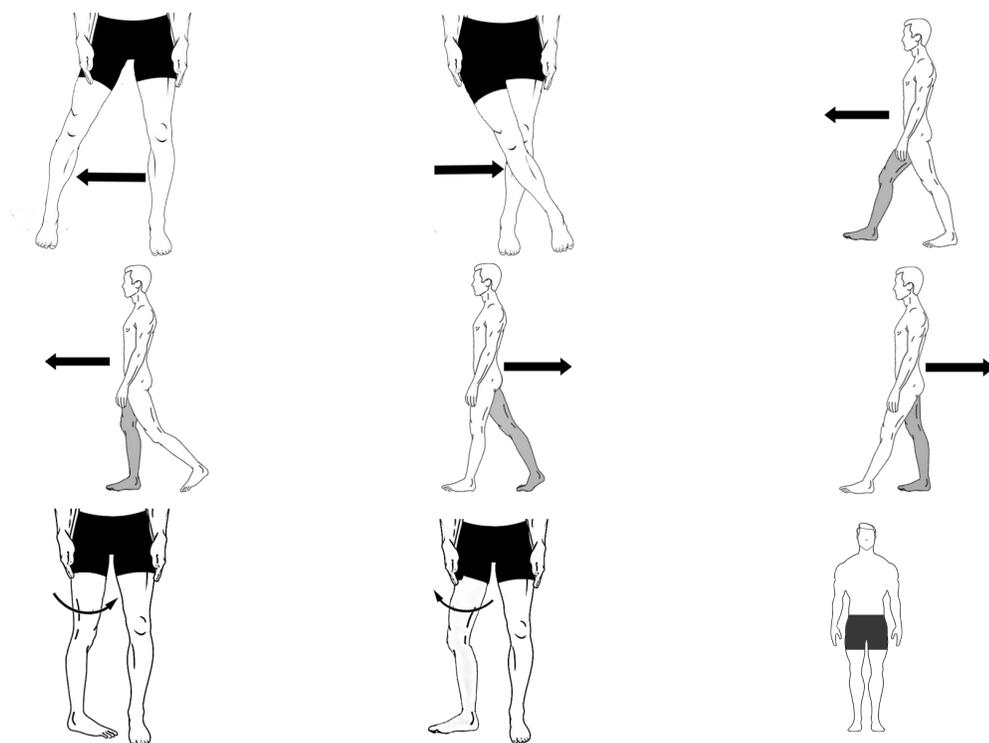


Figure A.2: Static gestures to be performed in Standing Position Scenario

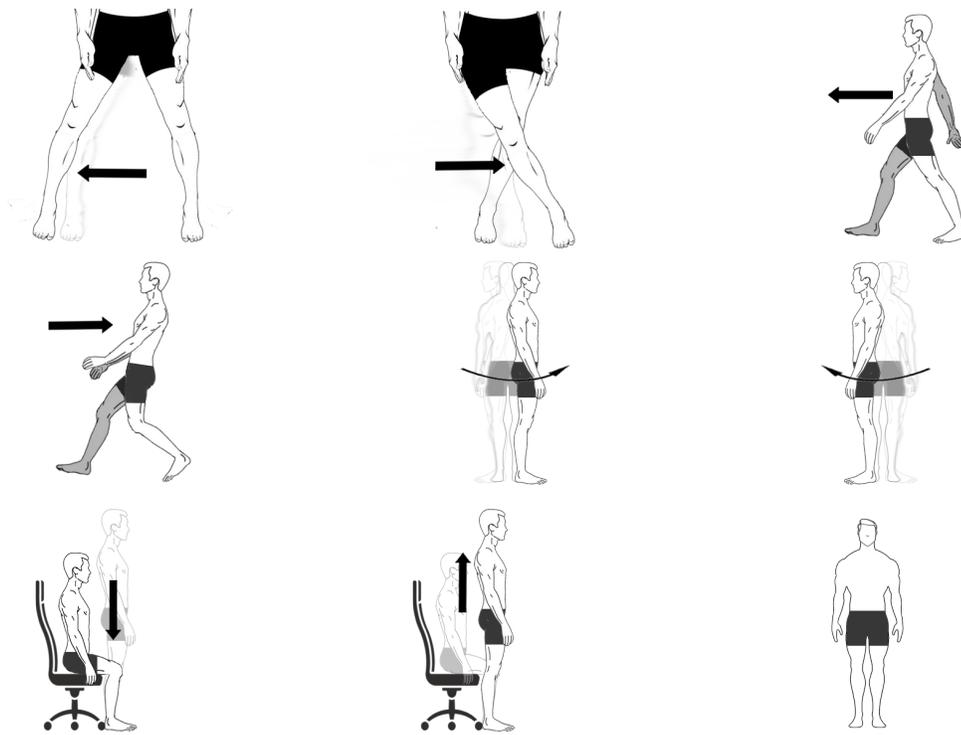


Figure A.3: Self-prevented gestures to be performed in Attempted Motion Scenario

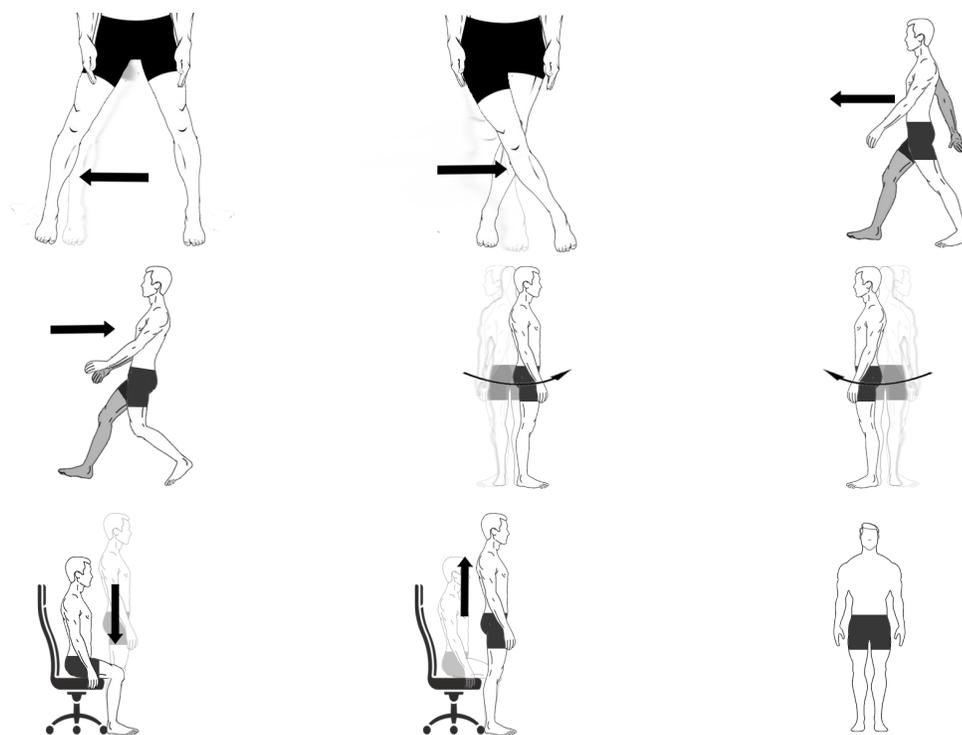


Figure A.4: Dynamic activities to be performed in Dynamic Motion Scenario

A.2 Ethics Clearance



Office of Research Ethics
4500 ARISE Building | 1125 Colonel By Drive
Ottawa, Ontario K1S 5B6
613-520-2600 Ext: 4085
ethics@carleton.ca

CERTIFICATION OF INSTITUTIONAL ETHICS CLEARANCE

The following research has been granted clearance by the Carleton University Research Ethics Board-B (CUREB-B). CUREB-B is constituted and operates in compliance with the Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans (TCPS2).

Ethics Clearance ID: Project # 114417

Project Team Members: Ms. Hasti Khiabani (Primary Investigator)

Mahdokht Golmo-hammadi (Student Researcher)
Mauricio Ledon (Student Researcher)
Nicholas Berezny (Student Researcher)
Mojtaba Ahmadi (Research Supervisor)

Study Title: Sensor-based intention detection for human movements and their effect on improving human machine interaction.

Funding Source: (If applicable):

Effective: October 26, 2020

Expires: October 31, 2021

This certification is subject to the following conditions:

1. Clearance is granted only for the research and purposes described in the application.
2. Any modification to the approved research must be submitted to CUREB-B via a Change to Protocol Form. All changes must be cleared prior to the continuance of the research.

3. An Annual Status Report for the renewal or closure of ethics clearance must be submitted and cleared by the renewal date listed above. Failure to submit the Annual Status Report will result in the closure of the file. If funding is associated, funds will be frozen.
4. During the course of the study, if you encounter an adverse event, material incidental finding, protocol deviation or other unanticipated problem, you must complete and submit a Report of Adverse Events and Unanticipated Problems Form.
5. It is the responsibility of the student to notify their supervisor of any adverse events, changes to their application, or requests to renew/close the protocol.
6. Failure to conduct the research in accordance with the principles of the *Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans 2nd edition* and the *Carleton University Policies and Procedures for the Ethical Conduct of Research* may result in the suspension or termination of the research project.

IMPORTANT: Special requirements for COVID-19:

If this study involves **in-person research interactions with human participants**, whether on- or off-campus, the following rules apply:

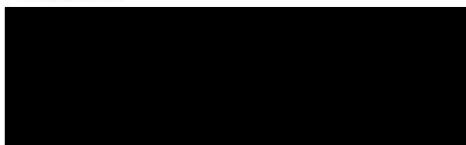
1. The study **requires prior approval of the relevant Dean's Office**. See [Principles and Procedures for On-campus Research at Carleton University](#) and note that this document applies both to on- and off-campus research that involves human participants. Please **contact your Dean's Office for information** about obtaining this approval.
2. You must submit a copy of that approval to the Office of Research Ethics prior to starting any in-person research activities.
3. If the Dean's approval requires any significant change(s) to any element of the study, you must notify the Office of Research Ethics of such change.

Upon reasonable request, it is the policy of CUREB, for cleared protocols, to release the name of the PI, the title of the project, and the date of clearance and any renewal(s).

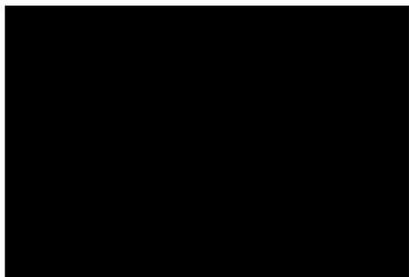
Please email the Research Compliance Coordinators at ethics@carleton.ca if you have any questions.

CLEARED BY:

Date: October 26, 2020



Bernadette Campbell, PhD, Chair, CUREB-B



Natasha Artemeva, Co-Chair, PhD, Vice Chair, CUREB-B

Appendix B

Custom-made 3d-printed parts

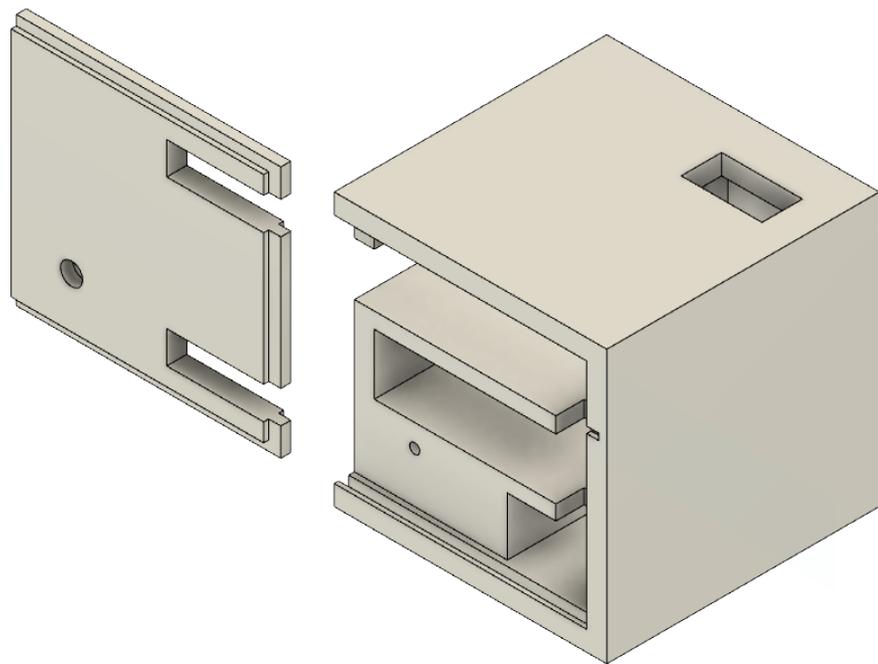


Figure B.1: The base box that holds the IMU sensor and the required batteries.

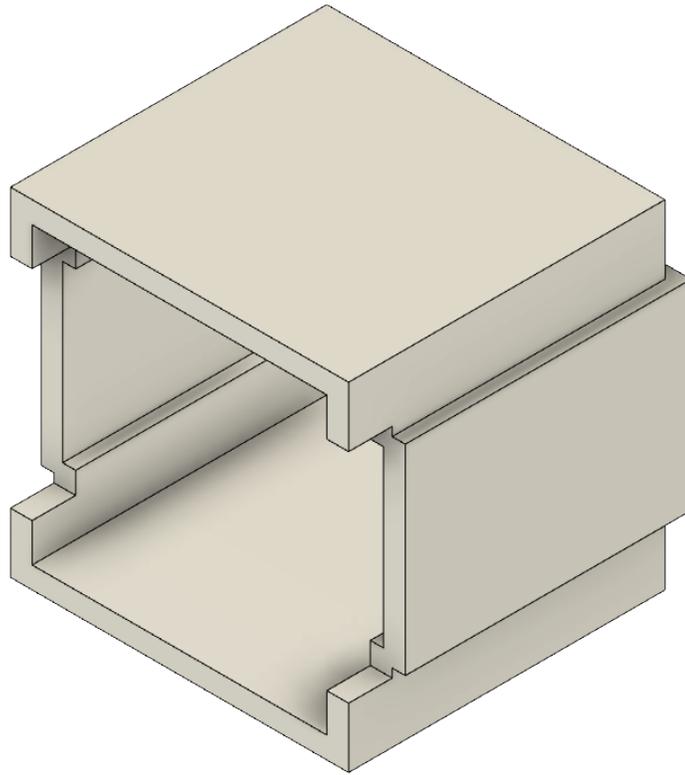


Figure B.2: The calibration unit box that encapsulates the base box and contains space for attaching a strap that covers the base box and attaches it to human body.

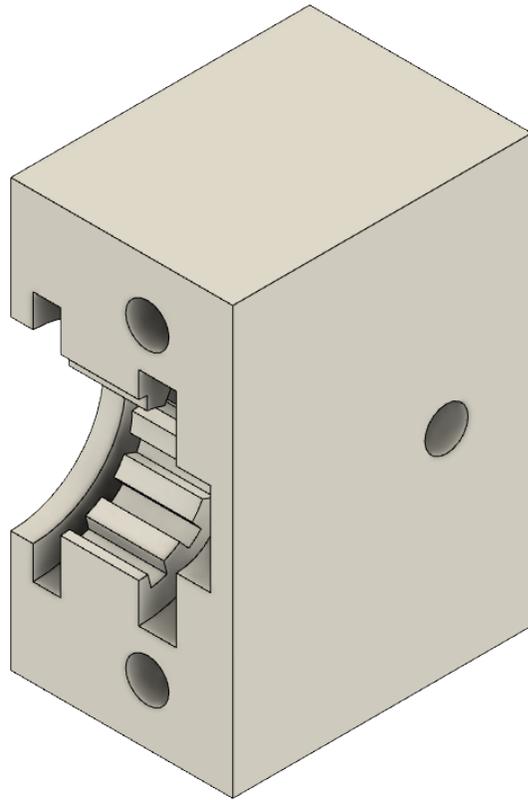


Figure B.3: The calibration box that encapsulates that IMU's base box

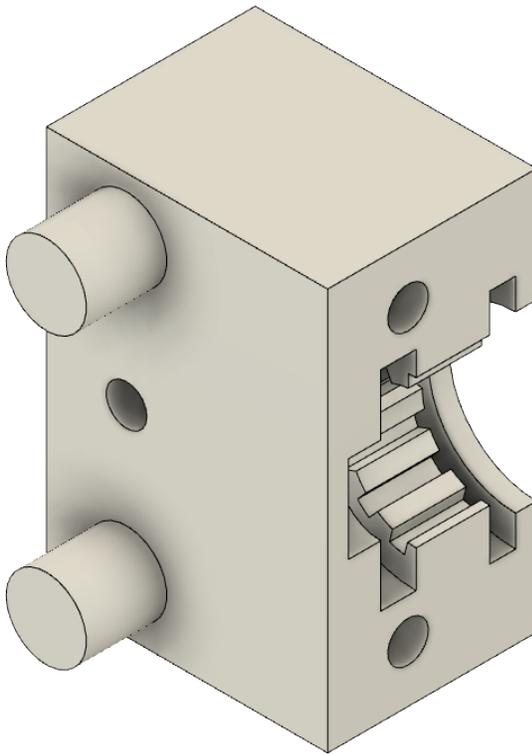


Figure B.4: The second part of the gear connector that also being used to attach the calibration box to the motor

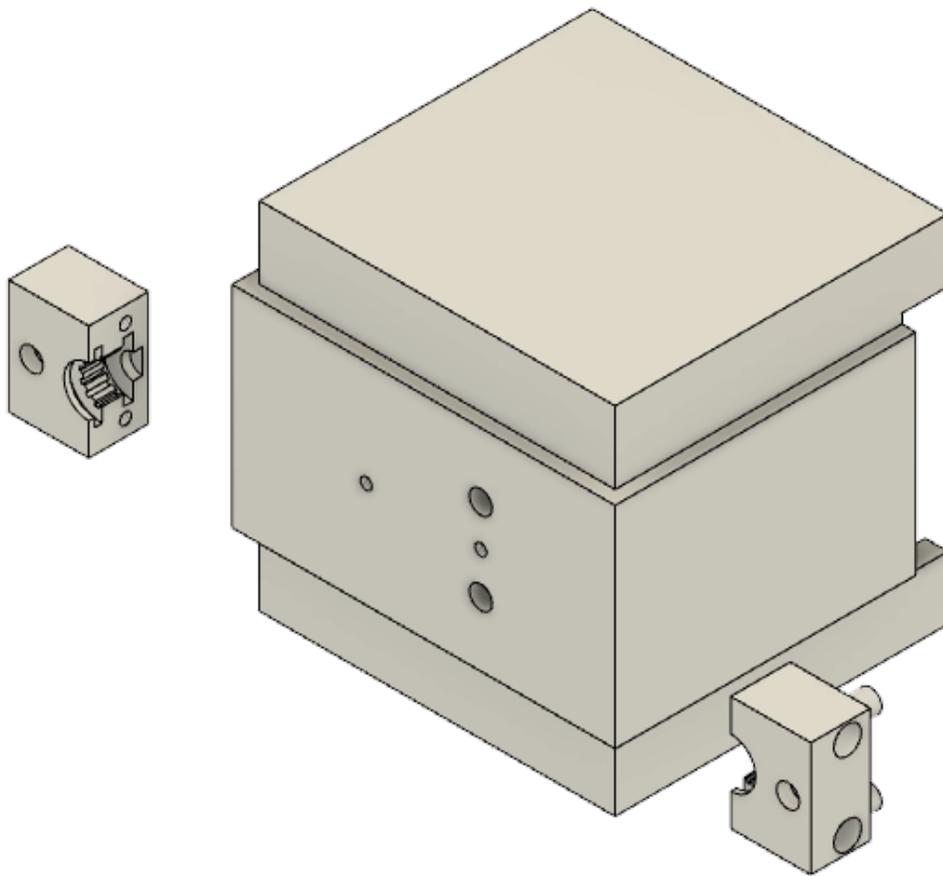


Figure B.5: An assembly of the calibration box with the the gear connectors.

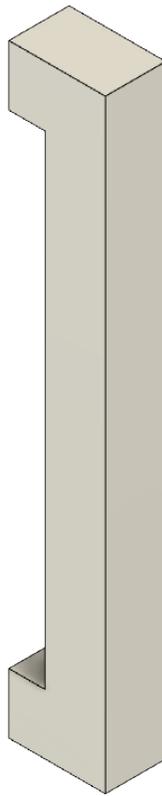


Figure B.6: The strap connectors to the Base IMU-two of these parts are manually glued to the final assembled system to hold the strap that covers the IMU base box.

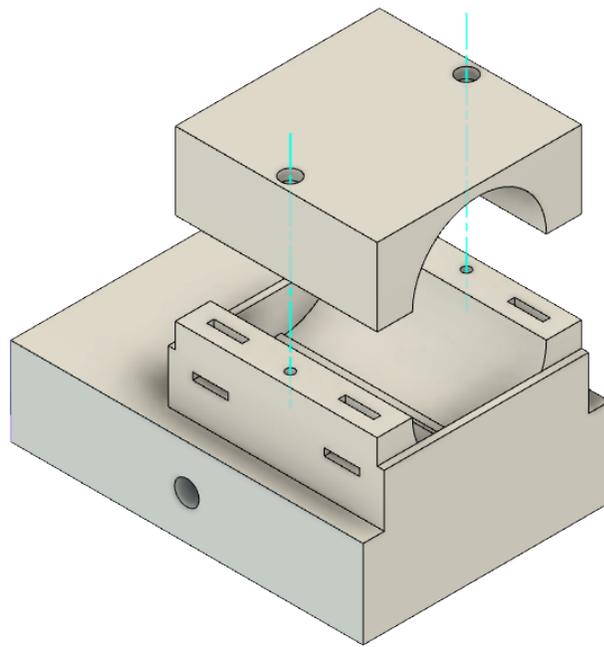


Figure B.7: The parts that hold the motor still and is being used to fix the whole calibration system to a fix table.

Appendix C

IMU Calibration and Fusion Methodology

C.1 IMU Calibration

An Inertial Measurement Unit (IMU) consists of several inertial sensors. Any sensor in an IMU (including accelerometer, gyroscope, and magnetometer) outputs a measured value that is different from the true value. The relation between the measure and the true values of each sensor can be modeled via an error model formula. This model includes the linear bias (offset), a non-unitary scaling matrix, a misalignment error matrix (which captures the non-orthogonality of the sensor channels, and the rotation between the frame of the sensor and IMU module(body)). Using the method introduced by Zhang et al [124], the unknown parameters of the Accelerometer and Gyroscope in the utilized IMU of this study have been identified. A calibration box has been designed, 3D-printed and assembled in a calibration setup (including stationary table, rotary means, ...) in order to calibrate the IMU sensor. The custom made 3d-printed parts involved: IMU box, calibration box, the parts to connect the IMU and calibration box to the gears, the motor holder and straps. All CAD models are presented in Appendix B. Since the calibration procedure has been performed in

this setup which is fixed to a reference table, the error model parameters work for the particular mounting position and orientation of reference frame of the calibration setup. Therefore, if for the experiment one was to move the IMU box and mount it to the user, there would exist mounting errors and other sources of error due to the change. As a result, the participants are asked to start the tests from a point where the IMU is in the calibration box so as to make sure that the error parameters are set to the correct values corresponding to the initiation pose.

The built in Levenberg-Marquardt (LM) algorithm in *MATLAB ver. R2020a*, which takes the advantages of gradient descent and the Gauss-Newton methods, is used to solve the optimization problems.

C.2 IMU Sensor Fusion

An attitude and heading reference system (AHRS) consists of a 9-axis system that uses an accelerometer, gyroscope, and magnetometer to compute orientation. Various IMU fusing filters are developed to produce a smoothly changing estimate of the device orientation, while correctly estimating the direction of north. Accurate magnetic field measurements are essential for a full sensor fusion and the determination of heading (the north) and orientation. Noise sources and manufacturing defects degrade a magnetometer's measurement. The most striking of these are hard iron effects. Hard iron effects are stationary interfering magnetic noise sources. Soft iron effects are more subtle. They arise from objects near the sensor which distort the surrounding magnetic field. Since the magnetometer in IMU is highly biased with soft-iron and hard-iron effects during the indoor data collecting nature of the project. Fusing the sensor with magnetometer to detect an absolute measurement with respect to inertial

frame is not practical. The fusion of the accelerometer and gyroscope produces also significantly smooths estimates of the motion, however it cannot correctly estimate the direction of north. If the imu fusing filter does not process magnetometer data, it simply assumes the device's X-axis is initially pointing northward. The motion estimate given by the imu fusion filter is relative to the initial estimated orientation. The solution that has been proposed for this problem is to use a fix box as the initial starting point that has been also used for calibration.

Several papers by Xiaoping Yun et al proposed methods to use IMU attached to a foot that could be used to track position through dead reckoning and integral drift corrected for each time the foot hit the ground. The same methodology is used to calculate the position of the user that is wearing the IMU on pelvis. The algorithm to fuse the IMU sensors are available at <https://x-io.co.uk/gait-tracking-with-x-imu/>.

Appendix D

Simulation Environment

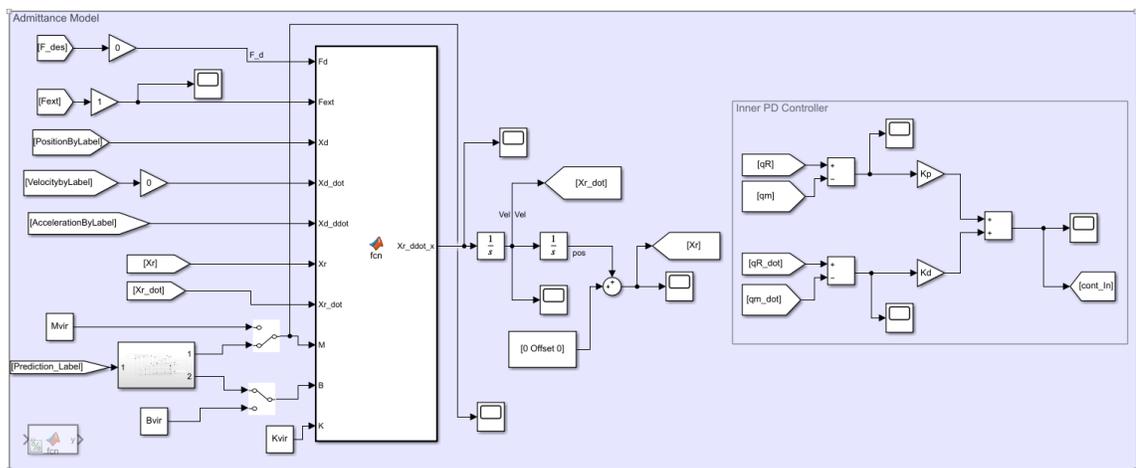


Figure D.1: Simulation block of admittance-controller

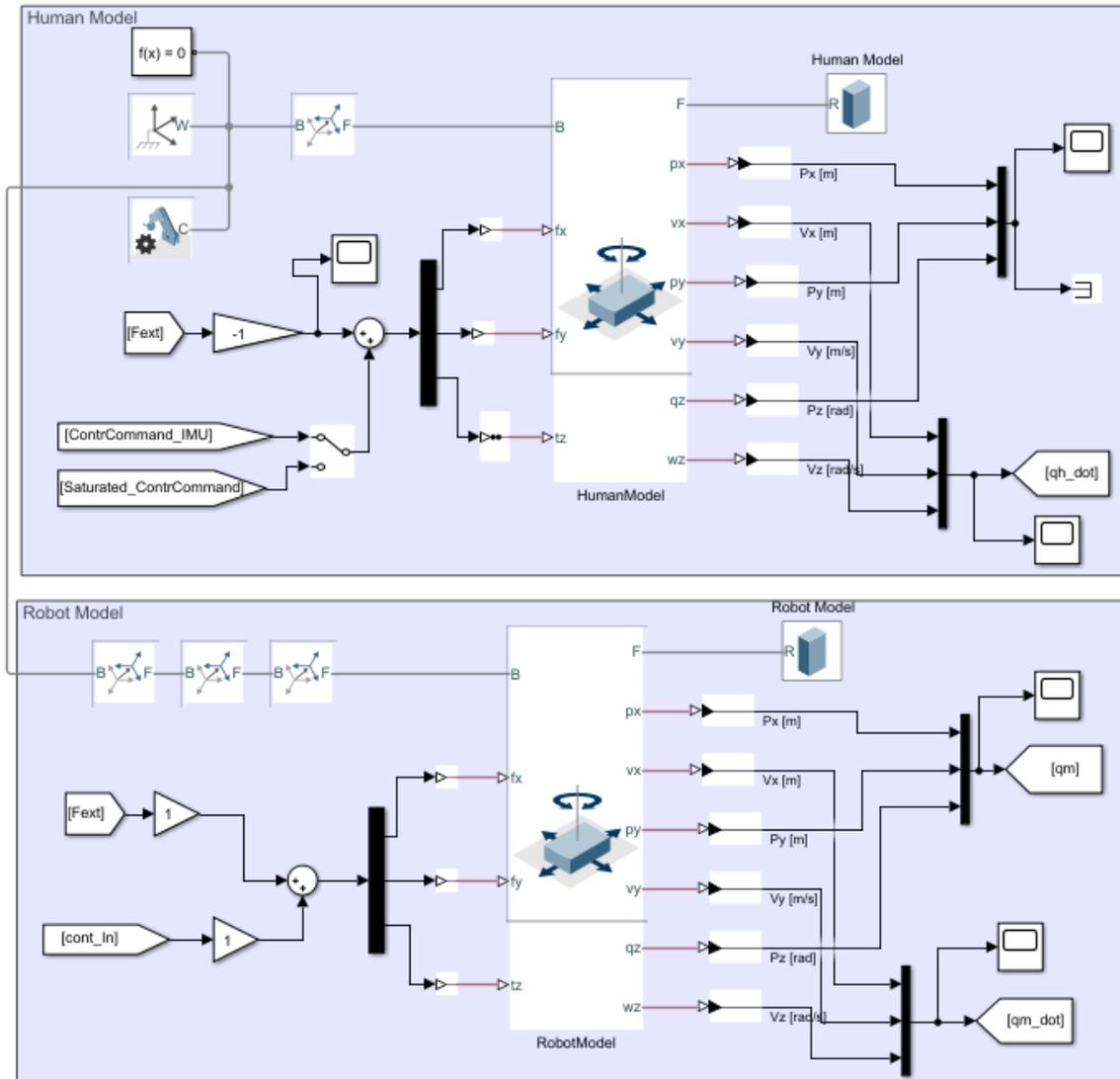


Figure D.2: Simulation block of Human Model and Robot Model

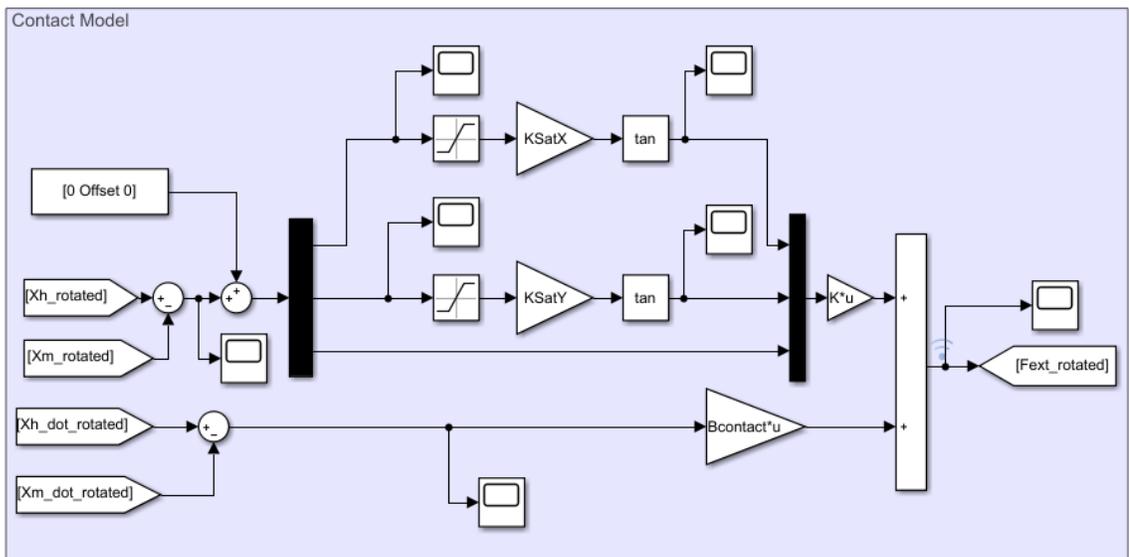


Figure D.3: Simulation block of Contact Model

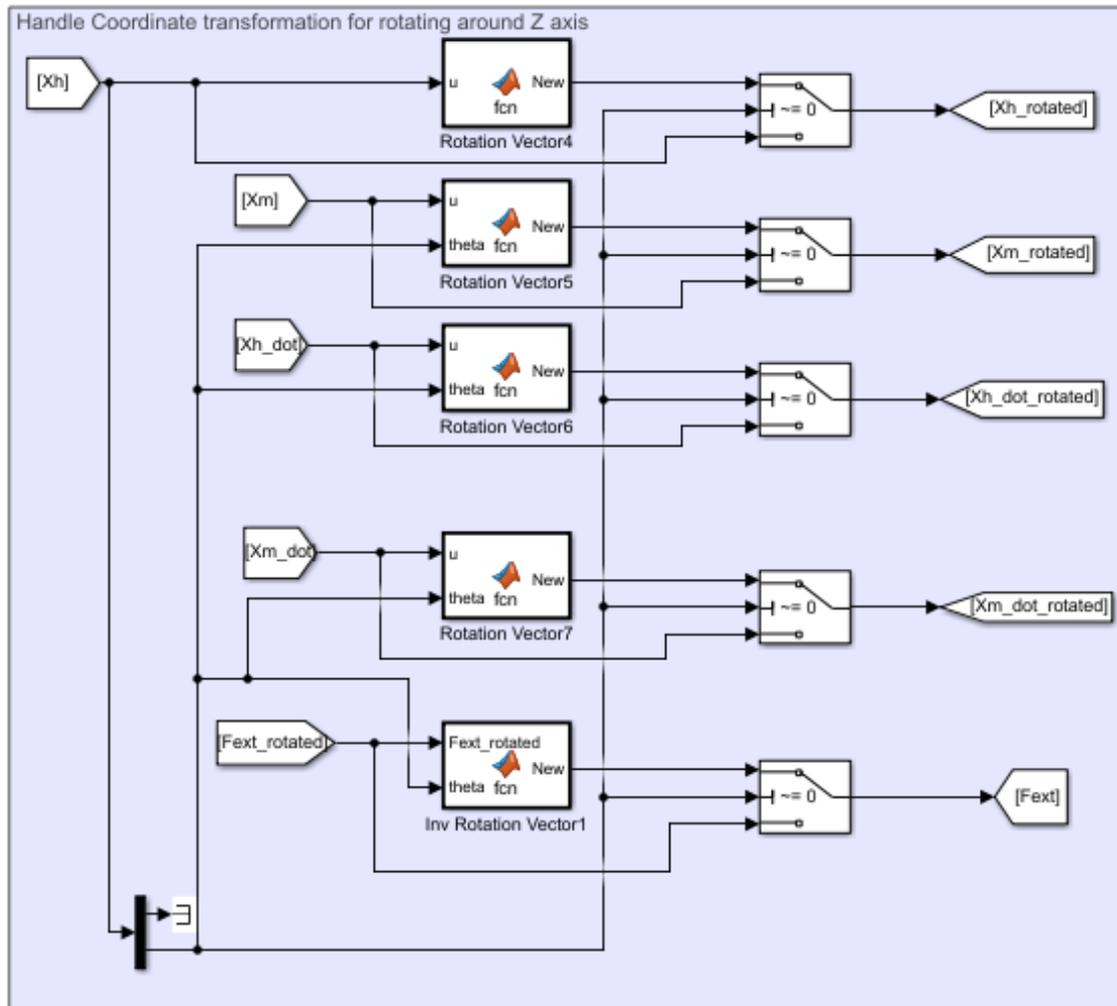


Figure D.4: Simulation block of Coordinate Transformation

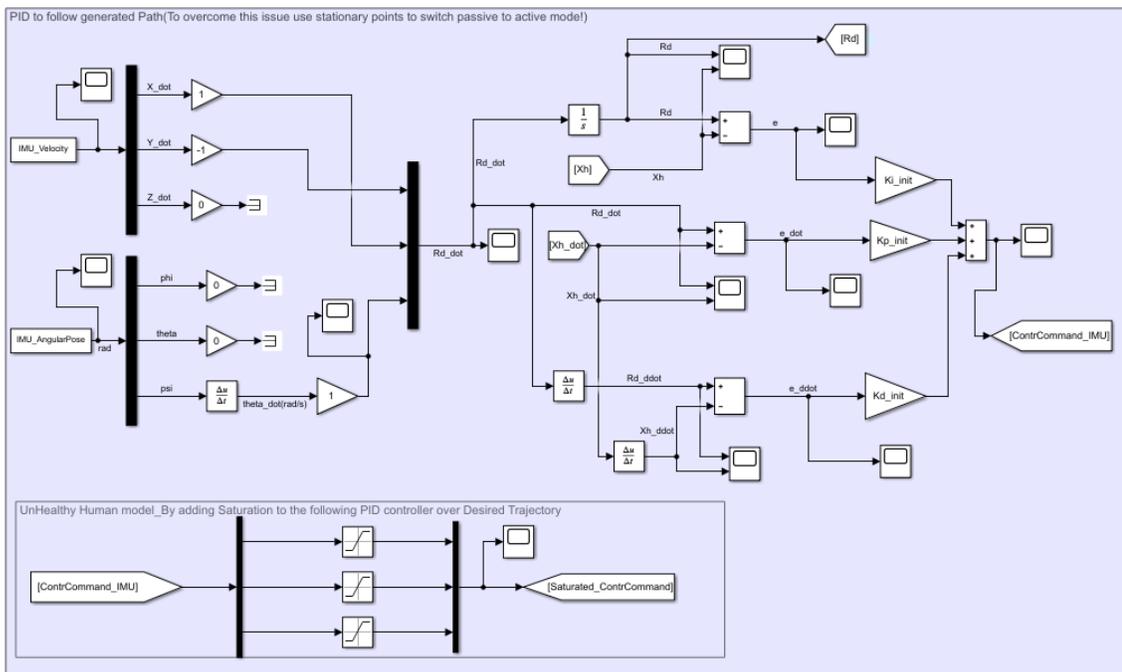


Figure D.5: PID control of IMU data and the unhealthiness model using saturation

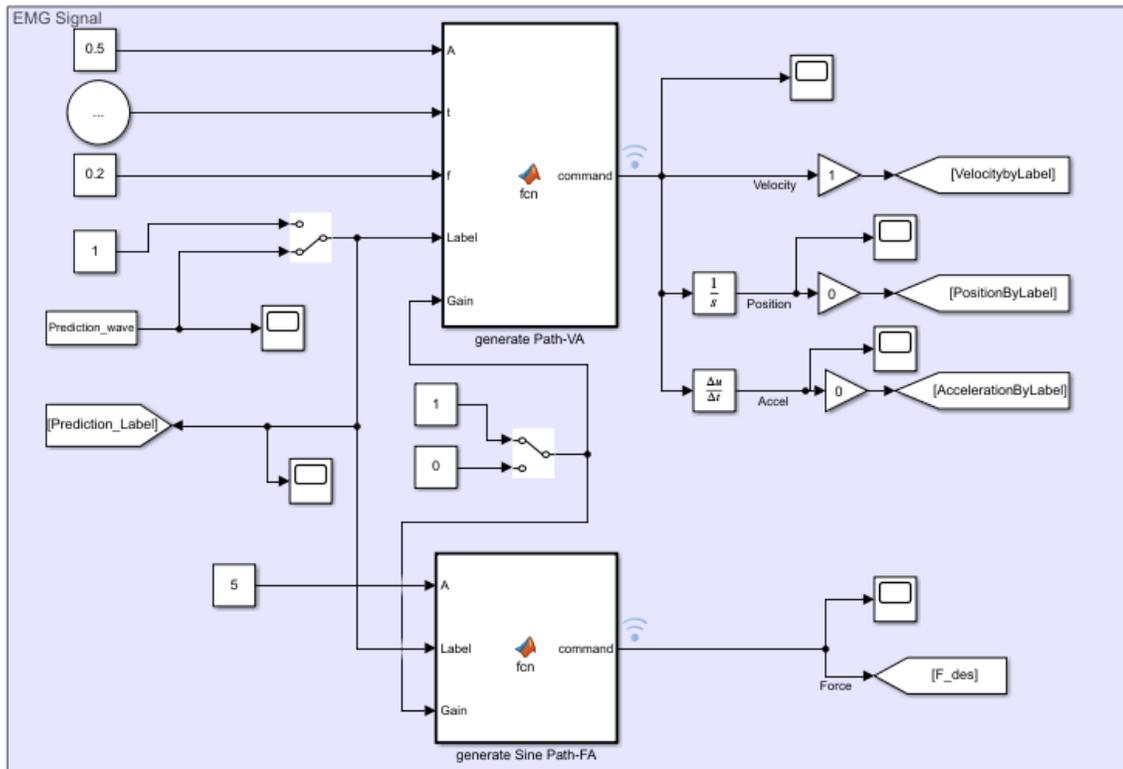


Figure D.6: sEMG-based command generation block

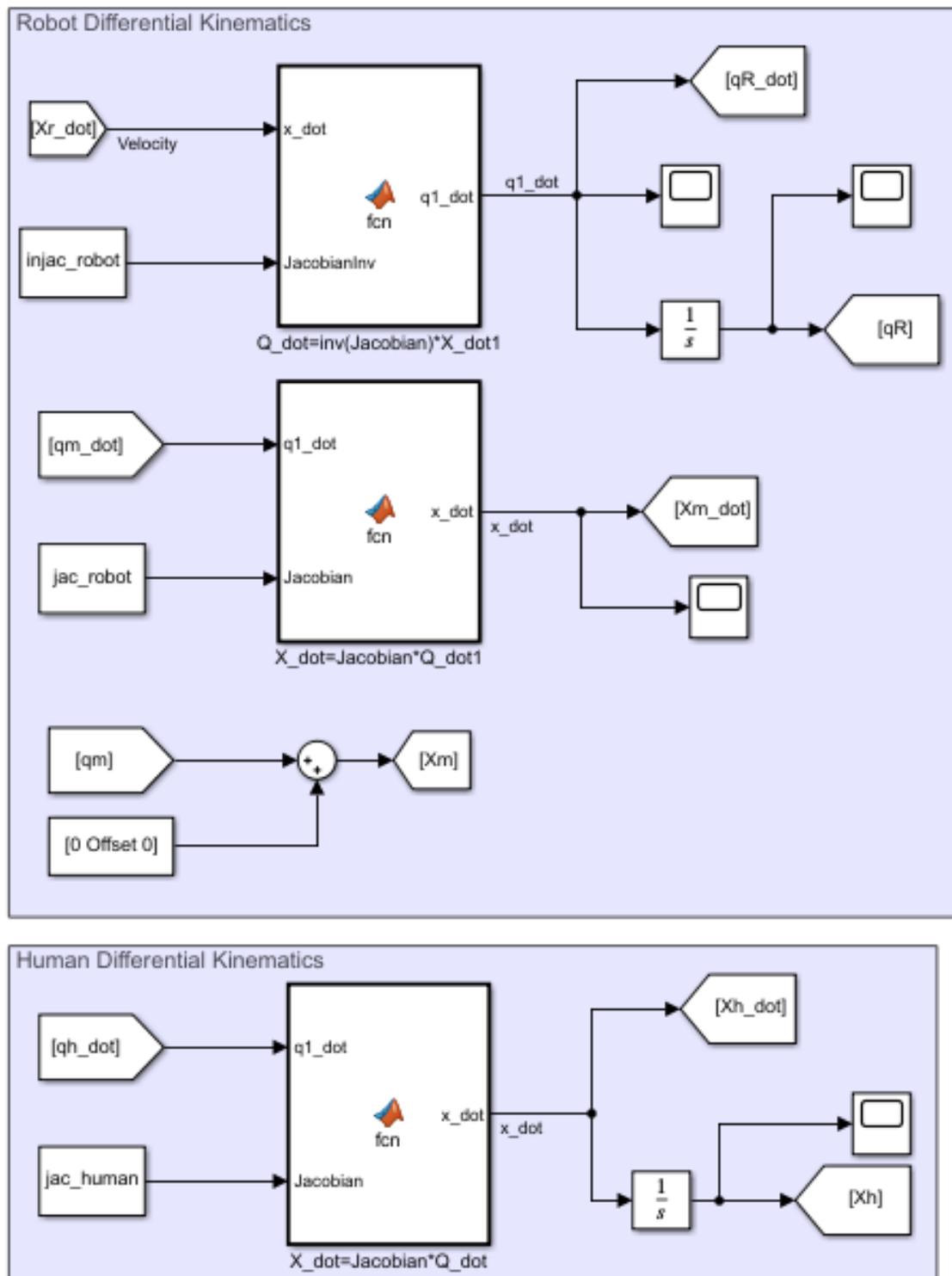


Figure D.7: Simulation Block of Human and Robot Differential Kinematics

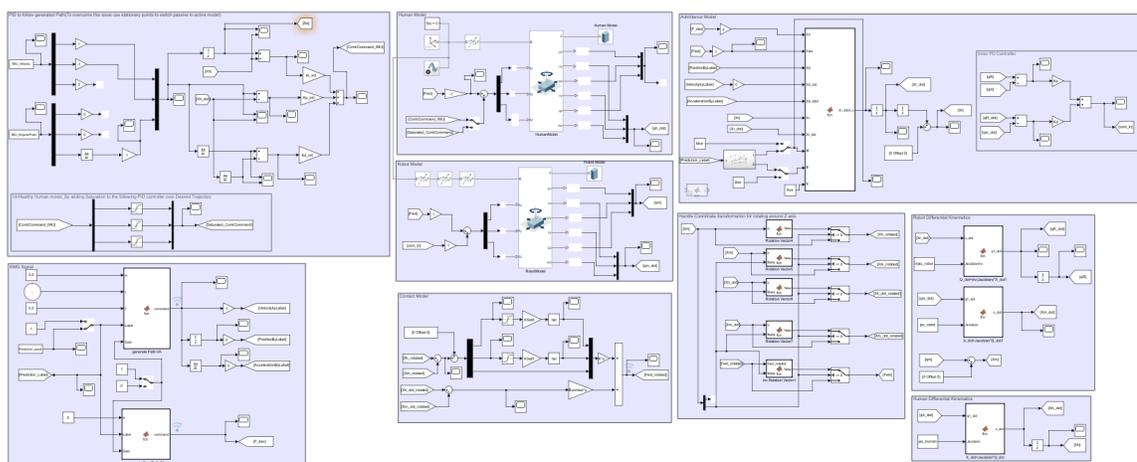


Figure D.8: All simulation blocks together