Development of Robotic Ankle Rehabilitation System to Enhance Human Machine Interaction

Thesis submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy
in
Biomedical Engineering

by
Palayil Baby Jephil

to

School of Biomedical Engineering
Faculty of Engineering and Information Technology
University of Technology Sydney
NSW - 2007, Australia

February 2023

AUTHOR'S DECLARATION

I, *Palayil Baby Jephil*, declare that the thesis is submitted for the partial fulfilment of the requirements for the award of Doctor of Philosophy in the *School of Biomedical Engineering, Faculty of Engineering and Information Technology* at the University of Technology Sydney, Australia.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

All the work described in this thesis was carried out under the supervision of Dr Steven Su. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others except as specified and summarised in the Statement of Contributions.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature: Production Note: Signature removed prior to publication.

Date: 22nd February 2023

ACKNOWLEDGEMENTS

First and foremost, I want to thank A. Prof. Steven Su, who has been my primary supervisor and advisor. He was helpful in the completion of this thesis, and without him, none of the work detailed here would have been possible. Prof. Su is to be commended for taking the time to explain and clarify the aims of my study. Prof. Su's collaborative chances have made this study quite fruitful.

I'd like to express gratitude to Professor Negin Shariati Moghadam, my co-supervisor, for her unflinching dedication to my research. Her expertise and leadership have facilitated my research management in several ways.

In addition, I'd like to express my gratitude to Mark Watford of the School of Sport Science for providing the information I needed to successfully complete this research. His thoughts about rehabilitation and user communication assisted in identifying research needs in the discipline.

My deepest gratitude goes out to everyone in the School of Biomedical Engineering, in the Engineering Department, and in the IT Department. The school's kind and helpful staff created an ideal research investigation setting. The key to the success of the study lies in the school and faculty's provision of resources.

I would like to take this opportunity to thank UTS for financially supporting my research via the provision of scholarships. I am thankful to the UTS Techcelerator programme for allowing me to show my findings to a wider audience.

I'd want to express my appreciation to my colleagues for all the times they've helped me out, given me advice, or given me constructive criticism.

Finally, I'd like to express my appreciation to my wife and family for their unwavering encouragement during the years of my study.

ABSTRACT

Ankle injuries are quite prevalent and are one of the leading factors that might prevent a person from engaging in daily activities. These injuries may result from running, falling, a stroke or several other neuronal injuries. Regaining complete ankle functionality after severe injuries requires therapy. Rehabilitation is one of the required methods that might expedite patients' recoveries from ankle injuries. In recent years, rehabilitation has included several technologies, such as robot-assisted rehabilitation. Even though research has shown the advantages of robot-based rehabilitation, there are several factors that must be addressed to have a better experience while engaging with the robot. In robotic rehabilitation, the utilisation of physiological signals has shown to be advantageous.

This thesis proposes methodologies to improve human-machine interaction by enabling control of the robot via physiological signals, reducing computational time, providing an adaptive control system and improving user safety. Initially, a mechanical structure for the robot which the users can easily use was designed. To control the robot via physiological signals, classifiers were used to recognise the intended movement using the user's electromyogram (EMG) and electroencephalogram (EEG) signals. A combination of features was used where classifying the EMG signal achieved a 98.9% accuracy. IMU sensors have also been used in parallel with EMG data to guarantee the user's safety while executing the exercises. Ankle plantarflexion and dorsiflexion torque and joint angle are also estimated using EMG. Parameters of the nonlinear mathematical models are determined for each ankle motion pattern using Swarm Techniques. Once the action of interest has been determined, the activation functions acquired from a single EMG channel are used in conjunction with pre-existing nonlinear models to estimate joint torque and joint angle. Results from the experiments proved the proposed strategy to be effective.

Movement classification based on EEG enabled the user to operate the robot by imagining the desired movement. A novel channel reduction strategy using ICA was proposed where EEG channels were decreased to 50% to achieve the same accuracy as using all EEG channels. An MRMR features reduction

approach was studied to lower the size of features. The findings demonstrated that employing just the dominant features to classify the signal had a negligible effect on accuracy.

After establishing the ankle joint's angle, this thesis's last module discusses a control system to adapt to the user's movement. Reinforcement learning, a self-learning machine-learning method, is utilised to train an agent to drive the robot. For this function, a deep deterministic policy gradient method was applied. The algorithm tunes a PID controller such that its parameters were changed in real-time based on the user's movement. The results demonstrate that the robot can follow the user's movement with less than 2% error.

STATEMENT OF CONTRIBUTIONS

The work presented in this thesis could not have happened without many outstanding individuals whose efforts made the contents presented in this thesis feasible. The multidisciplinary nature of this project necessitates that some of the data offered here be gathered with the assistance of others. My supervisor thought up the experiments in this thesis, with additional input from other colleagues.

In the following, I credit the contribution of other people presented by the chapter:

Chapter 3: The initial Housing for the ankle Robot was developed by Dr Tao Zhang (PhD) and Jessica Wind from the mechanical engineering faculty in collaboration with my research. They aided me in the implementation of my design into the Solid Works model. The design was further modified with the assistance of Jessica Wind.

Chapter 4: Torque collection experiments were performed with the assistance of A. Prof Mark Watford from the School of Sports Science.

Chapter 6: Dr Tao Zhang (PhD) from the school of mechanical engineering created the robot's MATLAB simulation mechanical model.

PUBLICATIONS

[1] P. B. Jephil *et al.*, "Estimation of Ankle Joint Torque and Angle Based on S-EMG Signal for Assistive Rehabilitation Robots," in *Biomedical Signal Processing*, Series in BioEngineering, 2020, ch. Chapter 2, pp. 31-47. (Chapter 4)

AWARDS

[1] Most Feasible Prototype Award, UTS Techcelerator 2020

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Ankle joints are important structural components of the human body since it helps to transmit the body's weight and the forces created by touching the ground. It does this by forming an intersection between the legbones and the foot, enabling the foot to move in several directions. A few different things could bring on an ankle injury, and the aftereffects of such an injury could significantly impact a person's day-to-day life. Some issues with the ankle can be traced back to a sports injury, while others are due to a stroke or neuronal injury. A sprain is the most frequently occurring type of ankle injury. Ankle injuries during sporting activities are caused when the ankle joint is subjected to movement outside of its normal range of motion. It is estimated that 230,000 people in Australia seek medical attention for injuries related to their ankles each year [1]. Even though a foot injury does not threaten a person's life, it can severely limit their ability to participate in various day-to-day activities. Young adults and athletes have a higher risk of sustaining an ankle injury. As a result, getting the appropriate treatment and proper rehabilitation for these injuries as soon as possible is essential for a speedier recovery.

Physiotherapy techniques, such as joint manipulation, muscle stretching, massages, and active and passive exercises, are the cornerstones of the conventional approaches to ankle rehabilitation [2]. A patient does not actively make any effort to move their ankle while they are participating in the passive exercise. The patient would need the assistance of a therapist to move their ankle joint. On the other hand, active exercises are those in which both the patient and the therapist play an active part. The patient attempts to flex and extend their ankle during an active exercise. Because many of the exercises are repetitive and very demanding, they have the potential to put a great deal of strain on the

therapists and cause them to become physically exhausted [3]. The patient's potential to resume day-to-day activities is directly affected by these exercises, which is one of the primary goals of treatment. The ability of the practitioner to apply adequate force and acceleration using their hands is another disadvantage of hands-on physiotherapy. This means that more effort and exertion may be required to help the patient, which is another disadvantage of hands-on physiotherapy. In some instances, the physiotherapist may face difficulties in performing their job due to the size of the patient. To combat these constraints, researchers are exploring the possibility of using robots to enhance the efficacy of rehabilitation.

Robotics has a lot of possibilities in rehabilitation, including the ability to monitor the progression of rehabilitation, the ability to perform various exercises using the same device, etc. Rehabilitation programs that include robots are becoming more common. According to Valadao et al., (2015) [4], using robotics in rehabilitation accelerates a patient's recovery without placing a physical strain on the therapist responsible for the treatment. Several pieces of research have shown that using robots with different control systems including impedance, admittance, and adaptive control offers beneficial outcomes [5-7]. On the other hand, the control method for making sure to have an effective way for the user to interact with the robot, which is required for successful ankle rehabilitation, has not been researched explicitly for active rehab exercises.

Robotic devices can be used to carry out a wide variety of exercises, each of which consists of the user providing the robot with detailed instructions regarding the movement to be carried out, which is defined as a "passive rehabilitation exercise." The patient's input is required for the robot to move during active rehabilitation exercises. This is done to encourage movement. Patients have a variety of options for interacting with these robots, including giving a physical command to the robot to perform a specific movement by pushing several buttons on its control panel. Patients can also use their physiological signals to control the robot. These signals include electromyograms (EMG), which show muscle activity, and electroencephalograms (EEG), which show brain activity. Signals from the electromyograph and electroencephalograph can be used to determine the intentions of human movement [8, 9]. Patients can have a more natural and beneficial interaction with the robots by using their physiological signals [10]. In addition, EMG signals can be utilised to predict human movements before the actual movement occurs [10]. On the other hand, patients with limited or no movement ability in their lower limbs, such as stroke patients, may benefit from

using EEG signals. Therefore, this thesis primarily focuses on using physiological signals to control the ankle robot.

The relationship between humans and robots has a long history and is only getting more profound and complex. The pace of technical progress and the flexibility of the interaction between people and assistive technologies have both accelerated in recent years. Therefore, there is a need for technological solutions within the rehabilitation industry. Therefore, the main objective of this study is to further develop an ankle rehabilitation robot to make gait training easier. To begin the process of developing a robot for the ankle, the first thing that needs to be done is to gain an understanding of how the joint in the ankle generates forces. Since EMG signals are nonlinear, they vary from one individual to the next. Despite the development of mathematical modelling techniques such as the Hill muscle model, to calculate the forces acting on the ankle, it is still difficult to estimate the forces accurately [9, 11]. However, using EMG with an appropriate mathematical model is considered to provide a solution to estimate ankle joint torque accurately.

In order to improve human-machine interaction by making use of physiological signals, the first thing that needs to be done is to design a simple mechanical model of a robot in which patients can rest their ankles. After this, the individual's physiological information from EMG and EEG signals are analysed to determine how they interact with the robot. Therefore, the robot would respond to these signals as control signals when they are sent to it. Machine learning strategies and optimisation algorithms have been implemented to determine human movement intent in light of the fact that physiological signals are inherently noisy and can vary significantly between individuals. Despite the advances that have been made in machine learning techniques, very little research has been done on developing an adaptive rehabilitation robot. The demand for physiotherapists has increased exponentially over the past ten years as a result of the growing amount of stroke patients. According to several studies, using robots in the rehabilitation process results in quicker recovery. Even though earlier research has used physiological signals to estimate human intention, very few studies have investigated the possibility of improving the robot's movement stability in relation to human intention. Within this project's scope, we investigate methods for enhancing computational efficiency and classification accuracy to enhance human-machine interaction. Specifically, our goal is to give the robot, a more natural response regarding the desired motion of the ankle joint.

To accomplish this goal with this project, we have combined a number of algorithms, including support vector machine, reinforcement learning, independent component analysis, and signal feature extraction methods. This thesis focuses primarily on four issues: creating a classification algorithm for a fast response using EMG signal fused with inertial measurement unit sensors for added safety of the user, exploring a channel reduction and feature reduction algorithm for EEG signals to provide high classification accuracy and improve computational efficiency; to use mathematical models to predict ankle joint torque and to design a control algorithm that can respond to the patient's changing ankle position. The study presented in this PhD project is an innovative methodology because it could help patients affected by stroke and ankle injuries to get back to their normal day-to-day activities as soon as possible.

1.1 AIMS AND SCOPE OF THESIS

The primary focus of this study is to control an ankle robot using physiological signals in an effective and efficient way to enhance human-machine interaction. The following aims have been carried out to accomplish this objective:

- To develop a simple mechanical model of the robot to enhance patient comfort and usability.
- Evaluating the reaction time of the robot with regards to human intent using the EMG signal.
- Identifying a simple control algorithm for selecting between EMG and IMU to improve user safety.
- To determine the required number of EEG channels and features to obtain high accuracy.
- To use mathematical models to predict the torque and position of the ankle using EMG signals.
- To investigate the performance of an adaptive controller so that it can adapt to the patient's movement.

During the study of this project, the subsequent research questions were raised:

- What is the influence of time on the robot when using the EMG signal?
- What signal features are to be selected to classify physiological signals accurately?
- How can the number of EEG channels be reduced efficiently?
- Which feature reduction method is to be used to improve computation efficiency without decreasing classification accuracy?
- How can the torque and position of that ankle be estimated from EMG signals?
- What parameters are required to develop an adaptive PID controller?
- What would be the effect of EEG channel reduction and feature reduction on the accuracy of the data?

1.2 NOVELTY, CONTRIBUTIONS, AND SIGNIFICANCE

This thesis presents research demonstrating novel methods for estimating movement intentions based on physiological signals and discusses strategies to improve human-machine interaction. We improve the human-robot interaction by increasing classification accuracy, decreasing the time needed to perform computations, and developing a method for adaptive control. The following is a description of the novel aspects and significant contributions made in this research project:

A. Providing a concept for the mechanical structure of the robot:

- An evaluation of psychological considerations of human-machine interaction.
- A braking mechanism to ensure user safety.
- Proposal of a novel design that is easy to use and portable.

B. Providing an effective method to enhance user safety using EMG together with IMU sensors.

- Systematic evaluation of the window size is required to extract features from the EMG signal without affecting the classification accuracy.
- An effective switching algorithm for selecting between EMG or IMU sensor as the robot's control signal.
- Using the IMU as a ground truth to observe ankle orientation with respect to their movement intent.

C. Proposal of a novel channel reduction method for EEG signal to improve the computation efficiency.

- Proposing a novel ICA method to identify dominant EEG channels.
- Proposing frequency domain features to improve EEG classification accuracy.
- Proposal of using MRMR method to find dominant features and reduce feature set dimensions.

D. Providing a method to predict ankle joint torque using EMG signal.

- Proposing the use of nonlinear equations with partial swarm optimisation methods to estimate ankle position for plantar flexion and dorsiflexion.

E. Proposal of an adaptive controller for the ankle robot.

- To develop an adaptive PID controller to move based on the user's intent using reinforcement learning.

1.3 THESIS OVERVIEW

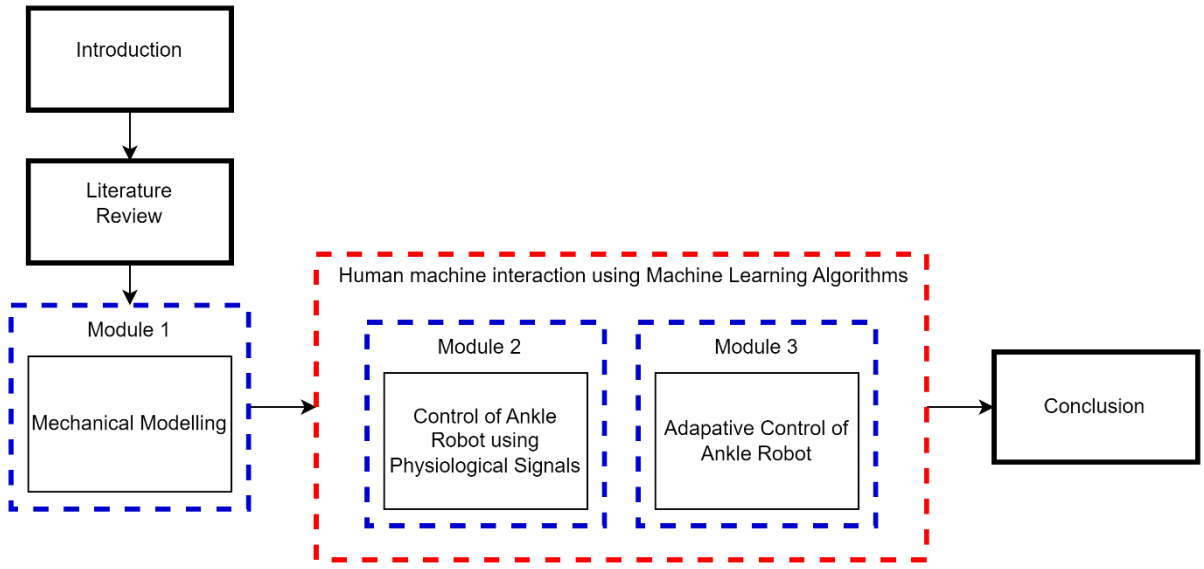


Figure 1.1. Thesis Overview.

In this thesis, 'Development of Robotic Ankle Rehabilitation System to Enhance Human-Machine Interaction,' an ankle robot is offered to improve patient recovery efficiency and enhance interaction with the robot. Three modules are present in this thesis, representing chapters 3 to 6 and shown in figure 1.1. In the first module of this chapter, we will discuss the physiological considerations that need to be considered when designing an ankle robot. We investigate the factors that affect the relationship between humans and robots. In light of this consideration, a design has been proposed that can be utilised by patients risk-free while also being easily movable in and around a working space. Because rehabilitation should be individualised and adaptable to each patient's requirements, the designed model includes two distinct components: a platform on which the patient's ankle will be kept and a separate leg support that will offer comfort to the patient. Furthermore, safety mechanisms have been taken into consideration to prevent ankle injury.

The second module uses Electromyogram (EMG), which are the electrical signals produced in the muscle. An efficient set of time domain features has been selected to determine the patients' intent. The classification of five distinct movements of the ankle that are discussed in this chapter is ankle plantar flexion,

ankle dorsiflexion, ankle inversion, and ankle eversion, as well as the ankle rest state. A cubic support vector machine (SVM) is utilised to classify the patient's intent. Different window sizes have been tested to study the robot's response times and determine how accurately it can classify the data. After the EMG data has been classified, we use an IMU to track the ankle's orientation. The signals from the IMU can also be categorised according to the same movement intentions. Signal to noise ratio is utilised to identify whether the obtained EMG is good quality. IMU will be used to control the robot when the signal noise remains high.

An investigation is conducted into a method that uses an EMG signal to estimate the torque produced by the ankle as well as the position of the ankle. The first step towards finding a solution was to determine the user's intentions by analysing their EMG signals. After the indent has been identified, the signal is run through nonlinear mathematical models, and then the particle swarm optimisation method is used to attain the values for the unknown variables in the mathematical equation. The torque value is determined by solving the nonlinear equation, which is then compared to the actual torque. To find the location of the ankle, a linear equation is formed between the ankle position and the ankle joint torque.

Electroencephalogram (EEG) signal is the other physiological signal utilised in this chapter's second module. An EEG is an electrical signal generated by the scalp. EEG signals are used to classify five distinct ankle movements. In this module, we investigate the effectiveness of three frequency domain features. After the dominant frequency domain features have been identified, an independent component analysis is performed in order to remove any artifacts that may be present in the signal, such as eye movements. This analysis is also used to determine which EEG channels are most prevalent. The precision of the SVM classifier is investigated with regard to the aforementioned EEG channels. After completing channel reduction, we minimize the size of the feature matrix by employing the MRMR methodology. After the features have been reduced, the accuracy of the classifier is measured once more so that its effect can be evaluated.

Creating an adaptive controller for the ankle robot is the focus of the final module in this project. This chapter uses reinforcement learning (RL) to change the parameter of the robot in response to variations in the ankle position. Deep deterministic policy gradient RL method is utilised for training the agent. The robot is guided by means of a mechanism known as a proportional, integral, and

derivative (PID) controller. The controller takes its values from the three actions that are output by the agents. The motion of the ankle robot is modelled in a simulation, which is then trained and put into use. This adaptive control approach can improve recovery in an efficient manner and enhance robot interaction. The adaptive controller considers the position values obtained in the fourth module.

This thesis contains seven chapters. The first chapter of this thesis provides an overview of the project and its significance. This chapter also includes the project's research objectives and research question. The second chapter provides a background for the current technologies and recent advancements in the field. The chapter provides an evaluation of the literature and is followed by the research gap in ankle robot rehabilitation. This project's primary contribution is from chapters 3 to 6. Each of these chapters begins with a brief introduction, which is followed by a description of the methodologies that include the history of the methods used and the rationale for selecting the method or algorithm. The third chapter examines the mechanical aspect of robots. The fourth and fifth chapters focus on using physiological signals for robot control and demonstrate a straightforward method for estimating ankle-generated forces using patrial swarm optimisation (PSO) and support vector machines. The sixth chapter presents an adaptive controller for the robot ankle. Chapter seven presents the conclusion and future directions for this research.

Data was obtained with the consent and approval of the Research council through the human ethics application (Application ID ETH17-1758). The application was specifically approved for the analysis of Human Movement through the examination of biomechanical and physiological signals. Each individual's data was collected on a voluntary basis. Strict confidentiality measures were implemented to ensure the privacy of the collected data, and no sharing of the data occurred on any platform. The gathered information will solely be utilized for research purposes, and all identifiable participant information will be completely eliminated from all aspects of this study.

1.4 COVID IMPACTS AND LIMITATIONS

Although the study project investigates identifying five basic ankle movements, ankle dorsiflexion, plantarflexion, inversion, eversion, and ankle state of rest, only three ankle states would be used for real-time implementation. This is

because only one DC motor is available, which can only spin in one degree of freedom (DOF). Therefore, implementation in real time would be limited to ankle dorsiflexion, plantarflexion, and the resting condition of the ankle. The mechanical model illustrated in Chapter 3 is based on the current DC motor with a single DOF. Similarly, chapter 4 only accounts for dorsiflexion and plantarflexion for estimating the ankle joint torque. This is owing to the limitation of the available isokinetic dynamometer (Biodex System 4, New York, United States), which only measures continuous torque for ankle dorsiflexion and plantarflexion. This is necessary information for estimating ankle joint torque. Due to these hardware constraints, the adaptive control scheme outlined in Chapter 6 is confined to a single DOF movement.

Covid-19's restrictions and lockdown impacted the research project as well. Access to the required data and lab resources was restricted due to governmental rules and regulations. This had a negative effect on the overall study progress and significantly slowed the achievement of the set objectives and goals. Due to the ongoing pandemic, one of the limitations was the collection of data from diverse subjects. Maintaining social distance was challenging as two or three personnel were required to be in close proximity while collecting physiological data. Due to this, only five healthy subjects' data were collected for EMG signal classification, and three healthy subjects' data were collected for EEG signal classification. Since EEG required more training time for motor imagery tasks, only three subjects were included in the data collection. Due to the COVID-19 pandemic, the study's laboratory facilities were shut down. Testing the developed algorithm and simulation model on the actual ankle robot was significantly delayed. Due to these limitations, the mechanical model presented in the thesis has not been physically constructed, so the effectiveness of the working model cannot be evaluated on a subject. Due to the restrictions of the hospital clinician, the proposed methods have not been evaluated on real-time patients with ankle injuries. Regardless of the hurdles faced by this limitation, the quality of research has not been compromised.

2 LITERATURE REVIEW

2.1 ROBOTIC REHABILITATION

Since the early 2000s, the concept of robot-assisted rehabilitation has been proposed. The number of patients in need of rehabilitation continues to rise. By catering to the growing population, robotic devices could potentially solve the problem. Several clinical studies have demonstrated robotic rehabilitation's efficacy and therapeutic potential [12]. A study by Krebs et (2003) [13] demonstrates the effectiveness of robotic therapy for rehabilitation. The devices could operate passively, in which the exercises are limited to a specific range of motion, or actively, in which the exercises are guided in response to the user input. Robotic rehabilitation improves treatment and maximises the best possible recovery outcome [13]. Physiotherapists have developed a series of rehabilitation exercises to treat patients effectively. End-effector rehabilitation robots, such as MIT-Manus and Haptic Walker, could simulate walking and stair climbing [14, 15], demonstrating positive outcomes in robotic gait rehabilitation. Ankle robots would assist in estimating stiffness and improving motor control for gait function.

In recent years, the use of physiological signals such as electromyogram and electroencephalogram signals has gained importance as they can provide feedback on the effectiveness of treatments. Several studies on physiological signal control have increased with the development of physiological signal acquisition devices. Recent studies have utilised surface electromyogram (sEMG) as one of their essential parameters. Possibilities such as movement estimation and prediction for rehabilitation demonstrate the potential value of sEMG treatments in rehabilitation [16]. Using EMG could help us estimate the patient's movement

intention before the ankle movement, as it occurs before muscle contraction [8, 9]. EMG control could aid in estimating the required forces for robotic control [17]. Moreover, robots can simultaneously assess the recovery stage based on physiological signals and provide patients with feedback [18]. This feedback can be provided via visual or audible cues or muscle stimulation.

Using machine learning algorithms has improved the rehabilitation process. They help to facilitate the creation of controllers that can achieve patient-cooperative control [19]. Artificial Intelligence (AI) for movement classification for optimal control has proven effective [20]. Incorporating multiple physiological signals into machine learning algorithms can help patients adapt to challenging exercises, thereby improving lower limb rehabilitation performance. Combining machine learning algorithms with adequate control strategies could improve the therapeutic efficacy of rehabilitation robots. Deep Neural Networks, k-Nearest Neighbors (KNN), Support Vector Machine (SVM) and Artificial Neural Networks (ANN), are all examples of machine learning algorithms that have been widely used to classify human movement patterns with an accuracy of at least 90% [21, 22]. The initial step in classifying human movements based on physiological signals is extracting valuable signal features. EMG and EEG features are primarily based on time and frequency domain features. Selecting the most pertinent features is crucial to improving classification accuracy. To estimate joint forces, researchers have combined machine learning algorithms and optimisation techniques such as Genetic algorithm (GA) and Particle Swarm optimisation techniques [23].

The subsequent literature will examine the various parameters affecting robotic rehabilitation, such as machine learning algorithms, control strategies, and sensing mechanisms, and will discuss the research gap that impedes the clinical application of rehab robots for ankle rehabilitation.

2.2 ELECTROMYOGRAM SIGNALS

When a muscle contracts, an electrical signal is generated that can be extracted and analysed for any disease or muscular dystrophy. An electromyogram (EMG) is a physiological signal used to investigate the motor functions of humans. EMG measures muscle activity directly, revealing how the motor cortex's commands are executed in relation to the intended movement [24]. Two properties make EMG an indispensable signal for robotic rehabilitation. The

EMG signal can be executed before the actual movement and linked to the brain signal. Therefore, EMG signals could be a practical basis for rehabilitation therapy and scientific research. The brain's motor cortex is responsible for muscle activity. The brain sends movement information to the spinal cord, which then transmits it to the skeletal muscles. The motor unit of the muscles comprises of neurons and muscle fibres which is the smallest unit of a skeletal muscle. Based on muscle size, the human body may have anywhere from 100 to 1000 motor units. Bigger muscles have more motor units.

During muscle contractions, the motor units and their stimulation frequency are influenced by velocity and force. The signal produced by the muscles depend on the total number of motor units involved in achieving a movement goal. Therefore, EMG can directly demonstrate the force and speed with which a muscle contracts. It is also essential to note that these two variables are not the only ones capable of altering the EMG signal. EMG patterns may be affected by muscle exhaustion, motor unit synchronisation, and the potential generated by individual muscle fibres. Surface EMG signals are a standard method for measuring muscle activity since they are non-invasive and do not need the assistance of a medical professional. Therefore, it is readily applicable to research studies. One of the important steps in sEMG analysis is separating relevant features from signal noises. Surface EMG is prone to signal noise which is one of its drawbacks. EMG signal is also affected by the quality of its hardware.

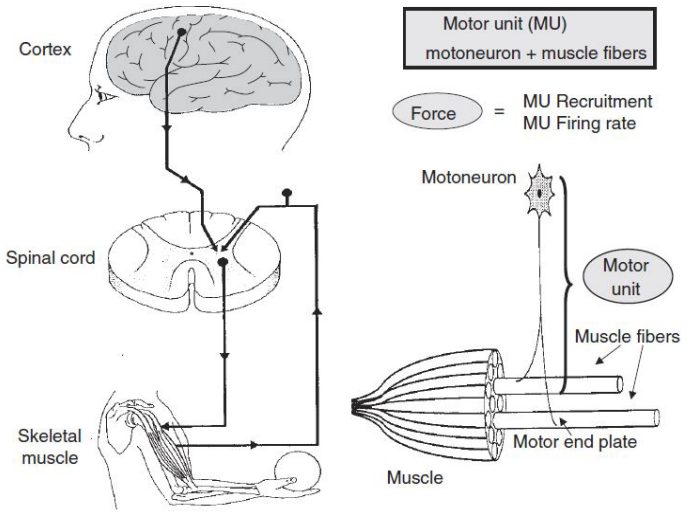


Figure 2.1. Illustration of muscle movement [24]

2.3 ELECTROENCEPHALOGRAM SIGNALS

Hans Berger invented electroencephalography in 1924 and applied it to humans [25]. An electroencephalogram (EEG) signal detects electrical activity caused by the brain due to several neural activities. The fundamental unit of the human nervous system is the neurons undergoing the six stages. The resting potential of a neuron is approximately a few millivolts. Following a stimulation, neurons create an action potential that opens voltage-gated sodium and potassium channels. Once the threshold is reached, the action potential stage occurs. The change in this potential can be measured using electrodes kept at the scalp of the brain. Due to EEG's poor signal resolution, the electrodes' placement is even more crucial for mitigating adverse effects. The international 10-20 system is often employed for comprehensive scalp coverage. Bony landmarks are used in the prefrontal, frontal, central, parietal, occipital, and auricular areas of the head to build lines of sensors, as shown in Figure 2.3. The 10-20 system and alternative configurations that increase the number of EEG electrodes are available to enhance signal resolution for research projects.

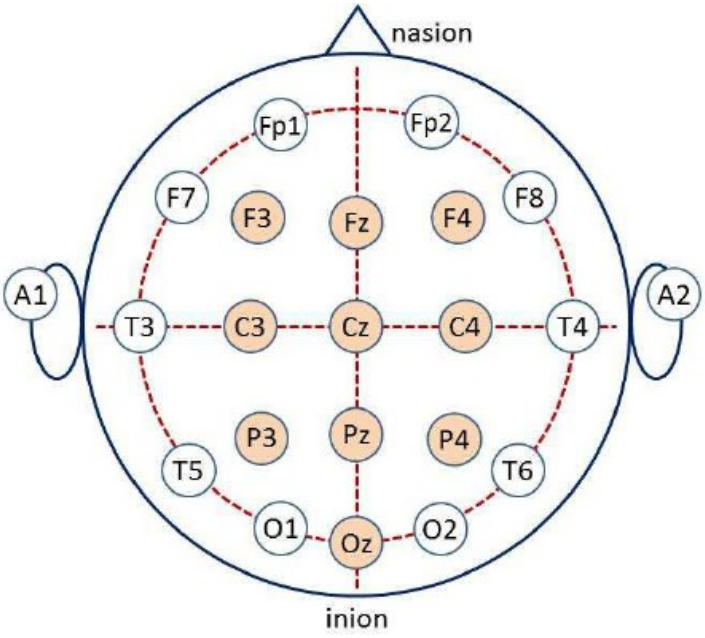


Figure 2.2. International 10-20 EEG electrode placement [26]

In recent years, signals derived from the brain have been widely employed for patient device control via the brain-computer interface (BCI). BCI is a system

in which EEG signals are collected, processed, and then used to instruct a computer to perform a task. This permits the application of a BCI system in multiple fields, including rehabilitation, assistive device control, and communication. We can now analyse EEG patterns with reduced noise using advanced machine learning and digital signal processing methods. This innovation has significantly impacted the diagnosis of epilepsy and Alzheimer's disease [27]. EEG can continuously monitor changes throughout rehabilitation training. Continuous monitoring, precise detection, and reliable prediction of brain activity are all made possible by modern EEG data processing. This thesis investigates the use of EEG to detect ankle movement intent.

2.4 ANKLE BIOMECHANICS

Before designing an ankle robot, it is essential to understand the ankle joint's movement mechanism. The ankle joint is comprised of thirty-three small joints that provide multi-axial motion to facilitate human gait [28]. There are six primary motions of the ankle namely, dorsiflexion and plantarflexion in the sagittal plane, inversion and eversion in the frontal plane and abduction and adduction in the transverse plane [29]. These six movements revolve around three axes at the ankle joint, permitting clockwise and counter-clockwise ankle movement. The motion of the ankle joint is shown in the diagram below.

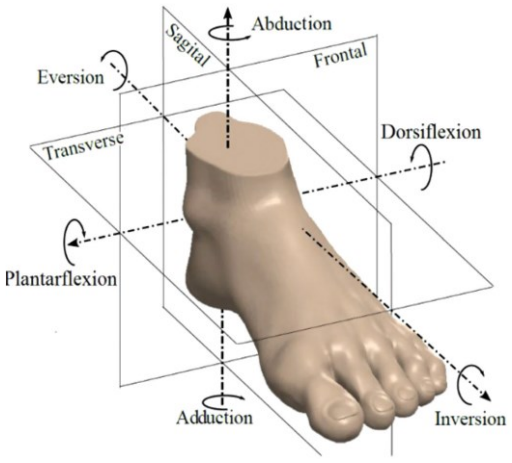


Figure 2.3. Illustration of the Ankle Joint movement [3]

The range of motion, also known as ROM, for each of the movements varies significantly from one individual to the next. These differences can be attributed to factors such as cultural norms, traditions, and an individual's day-to-day living activities. The ankle movement, specifically dorsiflexion and plantarflexion, which frequently occurs in the sagittal plane is primarily responsible for human gait movement [28]. The range of motion for basic ankle movements is outlined in Table 2.1.

Movement	Range of Motion in Degrees
Plantarflexion	37.6 – 45.8
Dorsiflexion	20.3 – 29.8
Inversion	14.5 – 22.0
Eversion	10.0 – 17.0
Adduction	22.0 – 36.0
Abduction	15.4 – 25.4

Table 2.1. Ankle Range of motion [30, 31]

2.5 SENSOR PLACEMENT

Understanding the ankle mechanics would aid in selecting the appropriate muscles for EMG sensor placement during rehabilitation. Due to the fact that the majority of ankle movements involve dorsiflexion and plantarflexion, it is necessary to examine the muscles involved in the aforementioned movements. Identifying and collecting EMG data from these muscles is crucial to robotic rehabilitation. The placement of EMG electrodes is provided by the "SENIAM" (Surface EMG for Non-Invasive Assessment of Muscles) recommendation [32]. A study conducted by Mademli et al. [33] demonstrates how the position of the EMG electrode can affect the estimation of the user's intent and the ankle position. The study's results indicate that the ankle position estimation could vary by 10 degrees, indicating that the location of the electrode will affect the outcome. Altering the placement of electrodes will significantly alter the signal strength for the same muscle activity.

For plantarflexion, the soleus and gastrocnemius muscles are primarily responsible [32]. At the same time, the anterior tibialis muscle is involved in

dorsiflexion, which raises the ankle. For detecting ankle inversion and eversion, the SENIAM standard suggests placing electrodes on the Peroneus Longus, Peroneus Brevis, and anterior muscle. Three EMG electrodes are generally used to classify ankle dorsiflexion and plantarflexion to control the ankle robot [17, 34]. The electrodes were attached to the gastrocnemius and anterior tibialis (TA) muscles. Other studies have only used one electrode on GAS and one electrode on TA [35, 36]. Three electrodes have demonstrated greater accuracy in classifying human movement intention. On the other hand, adding more electrodes could increase the computational complexity of the classification algorithm.

2.6 MACHINE LEARNING FOR REHABILITATION ROBOTS

The introduction of machine learning has led to solving computationally difficult-to-model problems. It has contributed to the evolution of all facets of life, including healthcare services. Pattern recognition consists primarily of feature extraction and classification. The accuracy of the classification is determined by choosing the most appropriate features and classification algorithm. Complex analyses of bio signals are possible with machine learning algorithms. Types of machine learning include supervised learning, in which the user assigns labels to the training data, unsupervised learning, in which the training data is not provided with its respective outcome; and self-supervised learning, such as reinforcement learning, in which the learning agent observes its environment to determine the outcome. The following literature examines machine algorithms and feature extraction techniques, as well as their implications for physiological signals, in great detail.

2.6.1 Feature Extraction

The accuracy of a classifier depends on the extraction of data features. It is the procedure of converting signals into feature vectors. The extraction of features from physiological signals such as EMG or EEG is divided into three main categories: time-domain features (TD), frequency domain features (FD), and time-frequency domain features (TFD) [37]. A signal's amplitude is the foundation of time-domain characteristics. There is no need for signal transformation to implement time features, so they are simple to implement. These features do not necessitate the signal input's transformation. Root mean square (RMS), Mean

absolute value (MAV), wavelength (WL), zero crossing (ZC), slope sign change (SSC), and autoregressive (AR) coefficient models have been suggested as time-domain features for EMG signals by [21]. Nevertheless, ZC and SSC exhibit few frequency domain properties. However, the signal does not undergo frequency domain conversion. Extracting MAV, ZC, and WL features for recognizing walking patterns results in an 80% classification accuracy [38]. Studies have also shown that adding features such as maximum amplitude and standard deviation (SD) can improve the overall classification performance [39]. Other characteristics such as skewness, kurtosis, and moving entropy have also been used to classify EMG. In the presented studies, RMS has been demonstrated to be a standard characteristic for all classifications because it provides a quantitative measure.

Frequency domain features concentrate primarily on the frequency components of signals. Regarding the classification of EMG signals, only a small number of studies have utilised these pattern recognition features. Features such as median and mean power frequency have been widely utilized for the classification of muscle fatigue [40]. EEG signal characteristics are more pronounced in the frequency domain, and studies have neglected the other domains to minimise the loss of input data [41, 42]. Multi-dimensional signal processing techniques, such as the Fast Fourier transform (FFT) and the wavelet transform (WT), the power spectrum, the energy spectrum, and the spectral power, have been used to classify mental tasks. Total power, short-term fourier transform, and signal-to-noise ratio have been utilised to diagnose epilepsy and convulsions [43]. FFT is the most frequently extracted feature from EEG because it enables the observation of the various frequency bands. Numerous studies have found that combining time and frequency domain characteristics improves classification performance.

Time-frequency domain features are generally utilized to detect system properties that vary with time based on the marginal spectrum [44]. No studies have used this property for classification or pattern recognition. Features such as discreet wavelet transform have been used to denoise physiological signals. This is because time-frequency domain features have very high dimensions. Feature reduction techniques, such as feature projection and feature selection, have helped reduce the data's dimensions [21]. Based on the literature, we can presume that time-domain features are optimal for classifying EMG signals. Whereas frequency domain features such as FFT and wavelet transform are optimal features from EEG classification. Table 2.2 below summarizes the most common signal processing and pattern classification features.

Time Domain Features	Frequency domain features	Time-Frequency domain features
Integrated EM	Mean frequency	Discrete Wavelet Transform
Mean Absolute Value	Median frequency	Continuous Wavelet Transform
Root Mean Square	Total power	
Variance	Frequency ratio	
Waveform length	Power spectrum ratio	
Zero crossing	The power spectrum deformation	
Slope sign change	Variance of central frequency	
Kurtosis	Energy	
Skewness	Signal to noise ratio	
Moving Approximate Entropy moving	Spectral moment	
Simple square integral	Signal to motion artifact ratio	
Average amplitude change	Wavelet decomposition	
Mean absolute value slope	Wavelet decomposition difference	
Auto-regressive coefficients	Modified mean frequency	
Standard deviation	Modified median frequencies	
Integral absolute value	Short-Time Fourier transform	
Variance		
Maximum amplitude		

Table 2.2. Common features used for physiological signal classification [21]

2.6.2 Feature Reduction Methods

The process of converting a given data into smaller features that can assist in classifying the signal is referred to as feature extraction. However, the key purpose of the features reduction method is to reduce the number of existing features by choosing only the most critical features from the provided data set. Since the cost of computation could be increased by having a large feature set, selecting only the most important features decreases the time needed for computation and increase the speed of the classifier. Having too many features can increase the correlation between the variables that make up the feature, which in turn can lower the accuracy of the classifier [45]. Techniques for reducing the number of features that are used include maximum relevance and minimum redundancy (MRMR), Relief-F, neighbour's component analysis (NCA), chi2, and principal component analysis (PCA). These are just some of the more common ones. This method can help rank the features and as a result, enables us to select the features that have the highest rank.

MRMR method is a feature selection algorithm that reasonably balances relevance and redundancy [46, 47]. It utilises mutual information between features to determine the relevance of a specific feature. NCA is a technique in which weights are assigned to the data set's features [48]. The feature weights are then optimised based on classification accuracy. Relief-F selects important features by applying weights to a set of features using statistical methods [49]. They can recognise conditional relationships between features and offer a clear classification overview. Chi-Square is a statistical test for determining whether two feature variables depend on their target class [50]. The features are discarded if the target classes are independent of their variable. PCA is one of the most frequently employed feature reduction algorithms, in which the algorithm searches for patterns in a feature set and compares and examines the similarities and differences between each feature variable [51]. It can be used in situations where identifying the prominent features is difficult.

Even though all methods can reduce the size of the features set, selecting the proper method is critical for obtaining good classification results. The MRMR and Chi-Square algorithms provide faster training, whereas the Relief-F and NCA algorithms offer only moderate training speed. MRMR and NCA algorithms can achieve a high level of classification accuracy. Recent studies [52, 53] have demonstrated the benefits of employing MRMR for feature reduction in biomedical applications. Literature also indicates that NCA required manual tuning of its

parameters, whereas Chi-Square and Relief-F had slightly lower accuracies than MRMR. On the other hand, information is lost when discarding features using PCA as a method for feature reduction. Since the MRMR algorithm is shown to have the highest accuracy in this thesis, we would use it to reduce EEG features.

2.6.3 Support Vector Machines (SVM)

SVM is a supervised learning algorithm used in machine learning for linear classification and pattern recognition [54]. It is based on an algorithm for supervised learning. Maximizing the hyperplane's margin between each data set improves the classification accuracy. SVM is primarily utilised to distinguish between two data sets. Figure 2.3 illustrates the working of SVM. SVM uses a kernel where the hyperplane can be used to aggregate sample data for multiclass classification. Thousands of data samples can be optimised by SVM for rapid problem-solving on data sets. Consequently, SVM could be utilised to analyse a vast array of biological data. Using EMG data, Khokhar, Xiao & Menon [55] predicted a person's intent within 250 milliseconds using SVM. The objective was to categorise a person's wrist movements. The classifier accurately predicted 19 intentions with an accuracy of 88% and 13 intentions with a precision of 96%. Using EMG, SVM has demonstrated its capacity to optimise training set data for vast dimensional spaces. As the number of classes to be classified decreases, SVM could achieve a 97% accuracy rate [56]. Gao et al. [57] classified six ankle movements for ankle rehabilitation using SVM with 12 features. The study demonstrated that SVM outperformed other algorithms in certain situations, such as walking uphill and on level ground. Several studies have also used SVM to classify mental tasks using EEG signals [58, 59]. As the number of classes and complexity of the data increase, SVM accuracy tends to decrease dramatically. Therefore, SVM is the most appropriate classifier for large classification datasets.

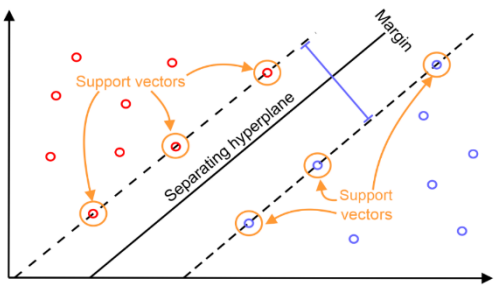


Figure 2.4. Illustration of SVM algorithm

2.6.4 Linear Discriminant Analysis (LDA)

As a popular supervised learning technique, LDA can be used for both classification and dimensionality reduction [60]. This classifier attempts to minimise interclass variance by increasing the distance between projected classes. LDA utilises eigenvalue, which corresponds to eigenvectors, to define the single hyperplane. Figure 2.4 illustrates the working of LDA. The relevant hyperplane is then used for classification. LDA has been employed extensively in tissue analysis, speech recognition, and text classification [61, 62]. When performing real-time classification, LDA is computationally efficient [63]. LDA could still be used for classification when data for classification are limited. Englehart and Hudgins [63] classified the four gait phases using nine EMG sensors and six force sensors. LDA was combined with SVM to achieve a 97% accuracy rate. In a recent study, healthy subjects participated in exercises such as walking uphill, downhill, and on level ground, and the LDA classification accuracy was 93.4% [38]. When classifying mental tasks or analysing epileptic seizures, a modification of the LDA algorithm was required for accurate prediction and analysis [64]. Using LDA to classify comparatively requires more features. LDA provides high accuracy for multidimensional data classification only when combined with other machine learning algorithms. Even though LDA can classify with limited training data and is computationally efficient, research indicates that using LDA alone does not increase accuracy. If computational efficiency is the most critical factor, then LDA would be the optimal classification method.

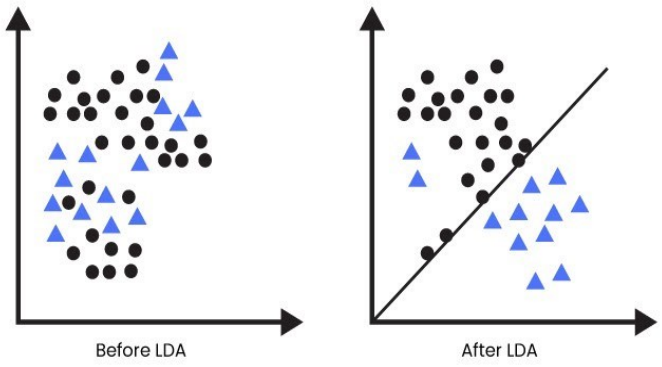


Figure 2.5. Illustration of working of LDA

2.6.5 k-Nearest Neighbour (KNN)

KNN is a classification process that assigns labels to unlabelled data based on the labels of the majority of the data's neighbours [65]. The distance metric that is utilised during data training has a direct influence on the performance of the classifier. The classification is done based on data points with the highest probability, which is based on the nearest neighbours of the test sample [66]. Figure 2.5 illustrates the working of k-NN. The KNN algorithm has memory and time complexity issues because the accuracy of the classifier is dependent on the data samples. Kim et al. [66] conducted research contrasting the effectiveness of KNN and LDA for classifying wrist motion. According to the findings, KNN had a higher level of prediction accuracy than LDA, which was 84.9%.

Archer et al. [39] utilised a hybrid classifier consisting of a 6-level decision tree combined with the KNN algorithm to categorise the activities involving ankle motion. For the purpose of this study, an ankle motion capture was performed using a two-second window and an overlapping one-second window. The training data contained 23 different gyroscope and accelerometer features. The findings show that a hybrid classifier can be utilised to classify gait activities for complex movements like walking on an inclined plane. Although KNN has reliably classified the data that was provided, it requires a significantly larger amount of memory than other models when applied to large data sets. KNN has also been utilised in classifying mental tasks by means of EEG. It was discovered that KNN is considerably quicker than ANN and SVM when classifying data. KNN has not been utilised solely as a classification algorithm for pattern classification; rather, it is combined with various other algorithms to achieve higher accuracy levels.

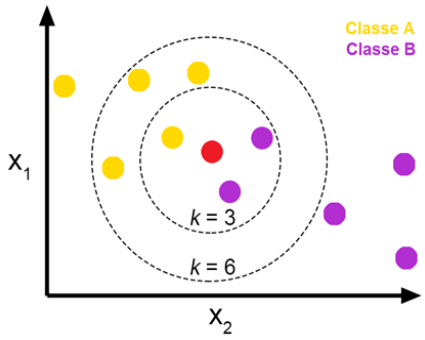


Figure 2.5. Illustration of KNN classification algorithm

2.6.6 Artificial neural networks (ANN)

ANN is an algorithm for learning that has its roots in biological neural networks and is based on them as its foundation [67]. Each node in an ANN communicates with the other nodes in the network, which is similar to neuronal connections in the brain. These nodes are equipped with weights, enabling the system to generate the intended output. In processing a wide variety of physiological signals, ANN has been widely utilised. The ability of artificial neural networks to produce nonlinear functions in response to linear inputs makes them particularly well-suited for pattern classification [68]. The classification of signals is the most common purpose of ANN. There are many different kinds of ANNs, but some of the most common ones are feedforward neural networks, back-propagation neural networks, recurrent neural networks and convolutional neural networks.

2.6.6.1 Feedforward Neural Network (FFN)

The simplest kind of ANN is the feed-forward neural network. The network attempts to obtain features from the input and generate a line or hyperplane between distinct classes [69]. FFN is a widely utilised data representation algorithm. Relationships between various inputs and their intended outcomes may be described using FNN. Figure 2.6 illustrates the working of FFN. A multilayer feedforward network was used to detect motion, with the two force sensors serving as gait classification inputs [70]. Ten hidden layers with a sigmoid activation function comprised the network. The research revealed that while walking on a flat surface, the accuracy rate was 99%. Similar research was demonstrated using FNN where EMG was used to control the robot's gait movement [71].

Using EMG, researchers could predict muscle activation during walking [70, 71]. Additionally, the FFN algorithm has demonstrated its ability to replicate human physiological signals. Estimating the ankle angle is one of the complex tasks involved in ankle rehabilitation. Zhou et al. [72] combined FNN and functional electrical stimulation to demonstrate its efficacy in estimating ankle position. Despite the study's proof of FNN's viability, the predicted ankle always contained a 2 to 5 % margin of error. Studies have also shown the effectiveness of FNN for neurorehabilitation of the ankle using a noisy EEG signal to detect movement intent [42, 73]

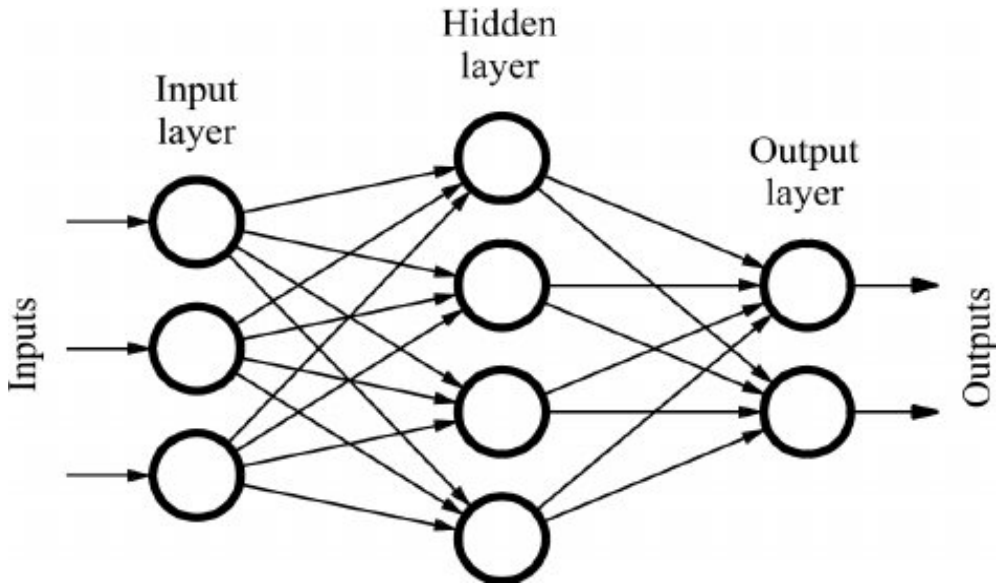


Figure 2.6. Illustration of FNN.

2.6.6.2 Recurrent Neural Network (RNN)

During RNN learning, data is transferred from one node to the next within the same layer or the layer beneath it, which maps into a learning procedure [74]. The data transfer in RNN is bidirectional between each neuron, and these neurons can perform nonlinear operations. Each neuron's output is proportional to the sum of its inputs. Every neuron in an RNN may be able to compute complex functions based on the data it receives. The RNN learning technique is put to use in several applications, including classification and time-series prediction. Even with a relatively small network size, this mechanism is able to classify patterns. Using EEG as a diagnostic tool for diseases such as Alzheimer's was one of the earliest applications of RNN [75]. With only a three-layer RNN network, achieving results with a high degree of accuracy was possible. RNN was used in the process that Li, Hayashibe, et al. [76] proposed as a method for patient-specific muscle modelling. The EMG could be used to monitor fatigue using the proposed model. RNN was not used on its own in this experiment; rather, it was combined with a nonlinear autoregressive exogenous model (NARX). The ability to estimate joint torque for the purpose of rehabilitation was demonstrated by the combination of NARX and RNN. The model demonstrated a strong performance while maintaining high accuracy and stability. RNN-based modelling performs better

than a typical classifier-based approach when it comes to the classification of EMG. RNN, when combined with long short-term memory, has demonstrated its effectiveness in classifying EEG signals for the recognition of emotions [41]. Combining RNN with different optimisation algorithms is one way to increase its efficiency. This is despite the fact that RNNs can classify signals using only a small network.

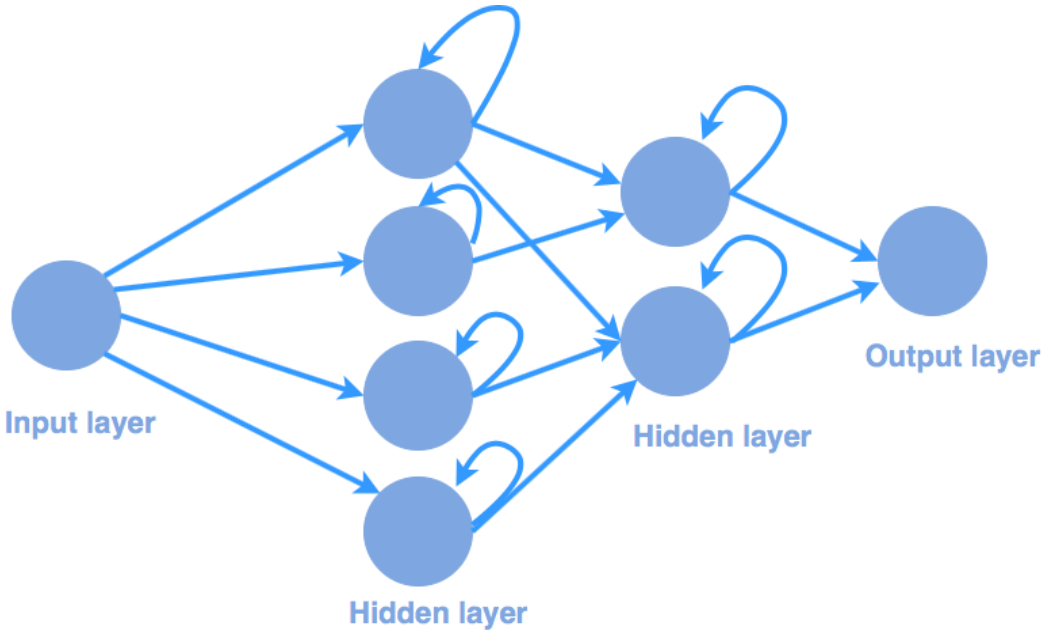


Figure 2.7. Illustration of RNN.

2.6.6.3 Convolutional Neural Networks (RNN)

CNN is a neural network that can maintain a spatial relationship between input data points. CNN can feed specific nodes in the network with small portions of the data rather than all of the nodes in the network. This property assists CNN in obtaining features from the data that is provided. CNNs are built to process data into multiple arrays, each comprised of multiple stages. CNN's are most commonly utilised in the image processing industry, where they are utilised to either blur or sharpen an image. By performing mathematical manipulation in either a linear or nonlinear form, the network is able to convert the input to the output that the user desires. When training for EMG pattern recognition, CNN has proven to be more effective than other methods [77]. In the experiment that Zhai et al. [77] conducted, CNN was used to classify hand movements based on

EMG signals. The experiment's findings demonstrated that classification accuracy was significantly improved compared to SVM.

Xia, Hu & Peng, [78] came up with the idea of a recurrent CNN, which was able to model the time steps of each movement of the upper limb by making use of EMG. The signals were collected at a sampling rate of 200 Hz while the proposed model was being used, and the data were segmented by using a window of 150 milliseconds. The model could accurately classify a time-varying signal and estimate the movement with high precision. The idea behind CNN has also been used in conjunction with EEG to identify seizures as they occur [79]. In Thomas et al. [79] study, CNN was used to obtain the features from the data, and SVM was used for the classification of features that were extracted. While the data was being tested on 156 subjects, the results showed an accuracy of 83.36%.

According to the research that has been done, to train an ANN, a very large amount of input data is necessary. This is despite the fact that using ANN is very effective in signal classification and feature extraction. Comparatively speaking, it is more complex than SVM, LDA, and KNN, all of which allow the classifier to make predictions with a small amount of data.

2.6.7 Reinforcement Learning (RL)

States, actions, and rewards are foundational components of RL learning. The agent is placed in a setting with multiple states. The condition describes the present state of the environment. A reward is given to the agent for being in a specific state, moving to another state, or remaining in a specific state. In RL, the learning agent is not instructed on what action to take; rather, it learns from the reward it receives after performing an action. RL algorithms are currently used to solve control-related issues such as power management, robotic control, rehabilitation robot control, etc. Since physiological signals are time-varying, it is challenging to establish human-robot cooperation using them.

Tamei & Shibata [80] developed an RL algorithm to observe the EMG signals of a user in order to optimise a force estimation. The objective of this experiment was to create a cooperative transfer of a specific load between locations. Cooperatively, the robot and the user could transfer the load to achieve the shared goal. This study demonstrates the effect of RL on developing a human-robot interactive system. Combining neural networks and RL has demonstrated a

99% accuracy rate in classifying EMG data for predicting elbow, finger, and hand movements [81]. Using these methods have shown to achieve a high level of accuracy using only four features. Using Q-learning, the neural network was used to approximate functions. RL was used concurrently to predict whether an attempt was successful or unsuccessful. Bingjing et al. [7] employed RL as an interactive gait rehabilitation controller. RL is also used to tweak the parameters of the admittance controller by receiving input from force sensors placed between the human and the robot.

Even though RL has not been utilised to improve classification accuracy, it has made a smooth transition, which has enhanced human-robot interaction. The powerful learning ability of RL has very few applications in clinical or medical settings. At this point, no published research demonstrates using RL as a standalone classification algorithm. It is essential to have a firm grasp of the fact that RL is primarily utilised as a learning controller. This property could be used to imitate patterns, allowing robots to perform the same movements that physiotherapists do.

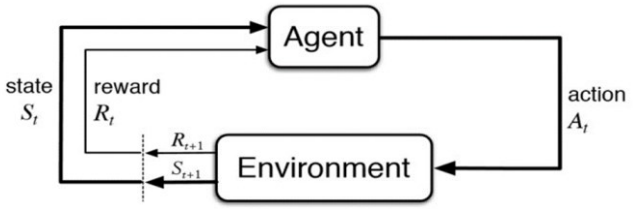


Figure 2.8. Workflow of Reinforcement Learning

2.7 OPTIMIZATION TECHNIQUES

In the optimization process, several potential solutions are evaluated against one another in a series of iterative steps until one is deemed optimal. Even though machine learning algorithms can anticipate a user's actions, it is possible that they will not be able to locate the user in the precise location that they have specified. When it comes to accurate movement, using optimisation techniques could help find mathematical models that could assist in providing estimates of parameters such as force and torque. Optimizing the parameters of machine learning algorithms would be possible with the help of these techniques.

2.7.1 Particle Swarm Optimisation (PSO)

By simulating the behaviour of a flocking swarm, PSO generates a large number of seemingly random solutions to an optimization problem.[82]. PSO is an algorithm based on social information sharing. Particles are the members of a population that search a specific space for the best solution. This search is based on the tendency of particles in a population known as a swarm to imitate the success of other particles. The swarm directly influences a particle's behaviour, and the particle could also influence the position of other particles in the swarm [82]. Each particle records its current optimal solution and its problem space, known as local best, and its neighbour's best solution, known as global best. PSO has been utilised to address issues involving EMG to force conversion. The classification performance of an algorithm may be affected by optimising kernel parameters. Using EMG, Nurhanim et al. [23] utilised PSO to estimate ankle torque. Several nonlinear mathematical models were used to estimate ankle joint torque, and PSO was used to determine the optimal mathematical model for estimating joint torque. By including a wavelet transform into PSO, a study showed the ability of the method to classify five different hand gestures with a 96.25% accuracy [83]. Modification of PSO can also be used to select the most helpful EMG features, thereby improving the overall accuracy of the classifier.

2.7.2 Genetic Algorithm (GA)

GA is an optimisation method in which the solution to a problem is found by applying recombination operators to data in such a way that essential information is preserved [84]. They are utilised when there are multiple optimal solutions to a search problem. GA does not require historical data in its search space, which may reduce computational speed requirements. GA provides the number of variations for each particle between generations or based on predetermined values. Therefore, the algorithm will display the optimal problem parameters after each iteration [84]. Although GA can be used to optimise machine learning classifiers, it is not fully automated because its parameters cannot be modified automatically. A study by Fischer et al. [85] for EMG classification demonstrates that as more features are extracted from a data set, using GA would make selecting the optimal features very efficient. GA was utilised to optimise the parameters of a mathematical model in which input EMG signals are used to estimate torque. In a study by Buongiorno et al. [86], using genetic algorithm (GA) and linear

optimisation methods for estimating shoulder and elbow joint movements were evaluated. The study results indicate that optimisation based on genetic algorithms were more effective for movement prediction. Compared to linear optimisation models, the error variance when predicting movement was significantly reduced.

2.7.3 Simulated Annealing (SA)

SA is a technique of global optimisation that can be applied to resolve combinatorial optimisation problems [87]. Annealing is a process in which a solid is heated to alter its physical or chemical properties, which is the inspiration for this algorithm; as a consequence, the iterative process of this algorithm is more time-consuming. In contrast to PSO and GA, this approach involves making only a minor adjustment to the initial problem. A comparison is made between the new solution and the previous one, and the new solution is chosen only if the probability of the given function has a higher value.

Parasuraman, Oyong & Jauw [88] estimated the joint torque and force for ankle rehabilitation robots using SA. This was done in order to design the robots better. During the experiment, a number of different mathematical models, both linear and nonlinear, were developed using EMG as an input. The goal of the algorithm was to choose a mathematical equation that was competent for making an accurate torque prediction based on the electromyogram signal. The characteristics of SA have been used in calibrating and simulating EMG-based models to understand the role that individual muscles play during walking and the ankle swing phase better [89]. As an EMG-based model optimisation technique, SA has not been utilised in many studies because the iterative process to find a solution is too time-consuming.

2.8 CONTROL SYSTEMS FOR REHABILITATION ROBOTS

2.8.1 Admittance Controller

Admittance control is a sort of control in which forces are delivered after detecting human motion. Recently, admittance controllers have been applied in rehabilitation robots for patient-robot interaction [7, 90]. The capacity to adjust its settings for adaptive control, which might aid in active training, is one of the key reasons why researchers choose admittance control. In rehabilitation, admittance control may be used to eliminate the synchronisation delay between humans and robots [91]. Based on the intended location, the controller might build a trajectory. A rehabilitation robot's non-stationary movement caused by EMG signals can be controlled using admittance control. The controller can measure human effort and translate it into the required joint movement. Bingjing et al. [7] applied RL to regulate the controller's settings. In this investigation, force sensor data was sent to the admittance controller. The controller was modified using RL to enable force-based interactive gait control. Zhuang et al. [91] modelled a PID controller using the ankle exoskeleton's admittance, achieving the required movement trajectory.

2.8.2 Impedance Controller

Impedance control is a method for shifting movement behaviour from extremely rigid to flexible. This control is typically used to define the state of the robot before a patient interacts with it [90]. Using the controller, robotic assistance is used to compensate for a patient's disability. The robot's force is dynamically adjusted using impedance control. Having a very high or very low impedance could make it very difficult for the patient to maintain control, or it could increase the risk of involuntary movement of the joints. A hybrid force-impedance control has demonstrated the potential for adaptive control in robotics based on the EMG signal [92]. Nizamis et al. [93] created a bionic arm with impedance control for human-machine interaction. The arm receives EMG signals from the affected limb, estimates torque, and uses impedance control to reach the desired position. The controller can also be used to produce constant torque impedance for varying stages of ankle rehabilitation following hemiplegic stroke [94]. The robot will only move when the detected force is below the threshold

value. After torque interaction, the study by Liu et al. [94] indicates that the foot pedal could easily track the patient's movement.

2.8.3 Proportional-Integral-Derivative (PID) Controller

Rehabilitation robots frequently utilise PID controllers. These are the most effective and straightforward controllers to implement. The controller's feedback mechanisms assist the system in producing the desired output [95]. It is a phase lead-lag compensator-based controller. The proportional controller regulates the action by delivering a signal that is proportional to the output error. Through low-frequency compensation, the integral controller minimises error. By compensating for high-frequency responses, the derivative controller contributes to improved precision. PID controllers regulate robotic velocity control. Zhuang et al. [96] improved human-robot synchronisation using a PID. PID was used to monitor the error generated by the admittance controller and to compensate for the error generated by the controller. When the ankle robot fails to achieve the desired position, PID controllers measure the ankle joint signals to compensate for the error. Since PID can be auto-tuned, it can be used as an adaptive control system.

2.9 INERTIAL MEASUREMENT UNIT (IMU) AND TORQUE SENSORS

Typically, an IMU sensor measures orientation, angular velocity, and magnetic field. The sensor consists of an accelerometer, gyroscope, and magnetometer, which can be used to observe orientation and determine travel direction. IMU, unlike EMG, can detect ankle position without requiring a machine learning algorithm. Kundu et al. [97] combined EMG and IMU to control a wheelchair using hand gestures. IMU was used to identify patterns, and the combination of EMG from two muscles helped achieve a 94% accuracy rate. Martinez-Hernandez, Mahmood & Dehghani-Sanij [98] attached 13 IMUs to an amputee's robotic leg, which helped classify walking on different terrains, such as on level ground or uphill. The combination of EMG and IMU has increased the precision and accuracy of movement from 80% (EMG alone) to 86% (EMG + IMU) [99]. Studies have demonstrated that using more IMUs can assist in segmenting

distinct phases of movement, which can help quantify movement limitations. IMU data can provide kinematic information about ankle movement.

Torque sensors are mechanical sensors that have been widely incorporated into robots for ankle rehabilitation. Typically, these sensors are mounted beneath the patient's feet to detect the force they apply. In addition to providing feedback, the sensors can be used to validate the classification algorithms' force estimates. Torque sensors placed between humans and robots can detect human intentions. The ankle and ankle robot interaction can be monitored using force sensors. Huang et al. [100] gathered torque and EMG sensor data to train a BPNN to estimate ankle joint torque. Training EMG data to its corresponding torque data has enhanced joint torque estimation [100]. Force sensors and EMG signals have recently been utilised to identify movement patterns. During rehabilitation, calibration of these sensors is crucial and can be challenging for therapists. In addition, Zhuang et al. [96] demonstrated that if motion intention is only based on torque sensors, it may cause a delay, resulting in poor human-robot interaction movement. These sensors can therefore be used to provide feedback and observe the patient's current ankle position relative to their desired position.

2.10 LITERATURE EVALUATION

Utilising robotics in rehabilitation can expedite the patient's recovery and reduce the workload of physiotherapists. The implementation of the Machine Learning algorithm and optimisation techniques aids in the estimation of the patient's intent to move. A patient's rehabilitation can be aided by creating a movement trajectory using physiological signals and sensors, such as IMU sensors. The majority of research focuses on classifying and predicting the intent of patients. Although robotic devices can mimic the therapist's action for passive exercises, these actions have yet to be demonstrated for EMG-based rehabilitation robots. Despite advancements in machine learning techniques, very little research has been conducted on creating an adaptive rehabilitation robot. In the past decade, the number of stroke patients and the demand for physiotherapists have increased. Using robots for rehabilitation has shown to improve recovery rates. Although previous studies have used physiological signals to estimate human intention, very little research has been done to improve the stability of a robot's movement in relation to human intention. Creating an adaptive control algorithm could positively assist the patient during physiotherapy.

The literature describes the current technologies employed to control rehabilitation robotics for active exercises. Several models have been implemented to predict the movement trajectory accurately; however, the parameters of each model should be altered for each individual. A significant research gap is finding an optimal control method to convert a non-stationary physiological signal to a stationary movement output that can replicate a physiotherapist's action. Reducing the number of sensors and improving the robot's response time are additional obstacles. These current gaps delay the commercialisation of an ankle robot based on physiological signals.

By providing constant repetitive exercise, the development and use of rehab robots based on the physiological signals of the ankle would transform rehabilitation into a more efficient process. Indirectly, it reduces rehabilitation costs by increasing recovery rates and decreasing physiotherapist dependence. This innovation could help physiotherapists monitor multiple patients simultaneously instead of treating each patient individual

3 MECHANICAL DESIGN FOR 1-DOF ANKLE REHABILITATION ROBOT

3.1 INTRODUCTION

A rehabilitation program requires a patient to undergo intense exercises and frequent training sessions. Although this could improve the patient's condition, conventional rehabilitation resources are limited. Rehabilitation robots for the ankle have demonstrated their efficacy and capacity to enhance a patient's condition [6]. However, these robots are primarily influenced by industrial robots in their design, which could reduce human-machine interaction and pose difficulties in clinical settings. Some disadvantages include stiff actuators, a high price, an intimidating design, and enormous magnets and actuators [101]. An example of an industrial based robot is the Lokomat System which is used for gait training [102]. The system's regular training regimen is inapplicable to all patients. In addition, the robot may be less interesting or inspiring than conventional modes of rehabilitation. The Lokomat robot employs stiff actuators that require a great deal of power to operate and has a very high production cost. Therefore, the mechanical structure of a rehabilitation robot is essential to its development. The ankle joint in humans is complex and capable of rotation in all three anatomical planes. Due to the complexity of the human ankle joint, rehabilitation robot design considerations should differ significantly from industrial robot design considerations. Platform-based and wearable robots are the two most common ankle rehabilitation robots. Platform-based robots are immobile and require additional support to hold the system in place beside the

user. Rutgers Ankle [103] is a six-DOF ankle robot which is a platform-based robot. The pneumatically operated robot had problems such as unwanted vibration and the inability to maintain high pressure [104]. In contrast, wearable robots do not require any support to hold the robot other than the user itself. An example of a wearable robot is the MIT Ankle Bot Roy et al. [105], which uses a pair of parallel linear drives to achieve six degrees of freedom.

However, some challenging factors, such as kinematic incompatibility, could result in misalignment between the ankle joint and the robot. Rehabilitation robots should adapt to each patient's needs and have a universal design. When designing a rehabilitation robot, the robot's stability, range of Motion (ROM), speed, strength, and ability to manually or automatically shut down the robot in the event of an unexpected event are among the most important safety factors to consider. Since both health care professionals and patients would use the robot, it should be designed with user experience criteria in mind, such as the robot's usefulness, accessibility, improved training results, ease of use, valuable outcome, easy learning for health professionals and physiotherapists, and the ability to adjust robot parameters for different tasks. This chapter proposes a mechanical design for a 1-DOF ankle robot capable of ankle plantarflexion and dorsiflexion. At UTS's Centre for Health Technologies, a new design will be developed for an already existing robot.

3.2 PLATFORM ASSISTED ROBOT REVIEW

This section reviews some of the current 1-DOF ankle rehabilitation robots and discusses their shortcomings and gaps. The ankle rehabilitation robot must utilise trajectory tracking position control to determine the patient's ankle joint angle from dorsiflex and plantarflex ROM movements. An early implementation of the ankle robot, where the robot was used to stretch the ankle towards plantarflexion and dorsiflexion for stroke patients, consisted of leg support, a footplate, a torque sensor motor, its gear head and housing for the robot [106]. Similar designs were implemented in the later years in different studies. Homma & Usuba [107] developed a design to fix the ankle joint in a specific position while simultaneously attempting to enable passive inversion and eversion in the footplate. Leg support was provided, which was kept very close to the ankle.

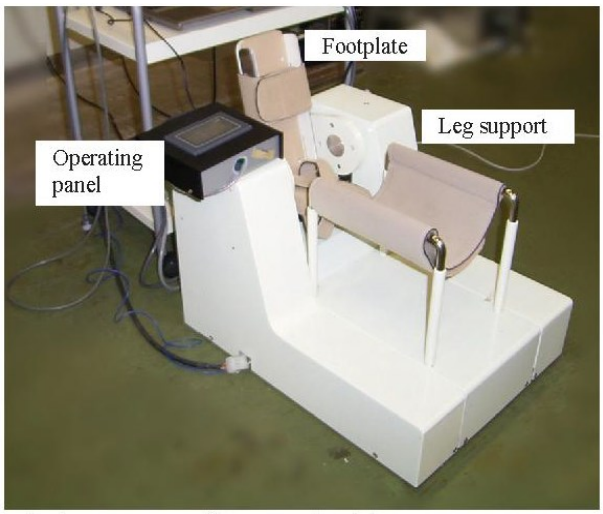


Figure 3.1. Ankle Robot Prototype [107]

The patient can only perform the exercises while lying down, which is a disadvantage of this design. Since the leg support is fixed, it cannot be adjusted to accommodate every patient. Cushioning the footplate and the leg support provides comfort for the user. Zhou et al. [108] designed a robot with a sliding platform to tailor it to patient requirements. The device looked big and had an overly complicated appearance which can also lead to potential hazards. Therefore, this project will use fixed chairs to maximise the available workspace, reduce potential dangers, and facilitate a more straightforward operation. In this design, leg support was provided for the calf muscles.

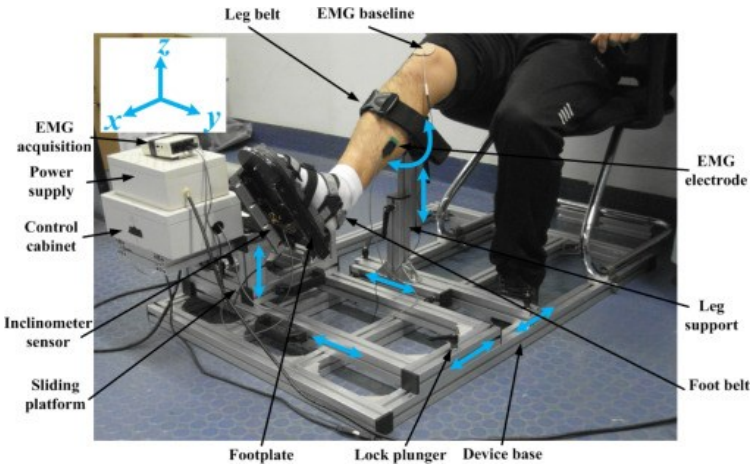


Figure 3.2. Ankle Robot Prototype [108]

Alternately, Zhou et al. [36] adjusted ankle flexion movements using a time belt and software position control to finalise the robot's position. This procedure appears laborious and time-consuming due to the use of the time belt. Ren et al. [18] designed a wearable robot for ankle rehabilitation for passive-controlled stretching. The motor and controller were attached to the user's ankle joint, making the device quite heavy. The device can only be utilised while lying down. A second prototype developed by Yao et al. [109] employs a similar assembly with the addition of a fixed leg support. This design limits the seating position of the user. Notables are the basic joint torque angle measurements that can be viewed on the metal plate adjacent to the foot platform and the plantarflexion ROM is restricted due to the platform's clearance from the ground.

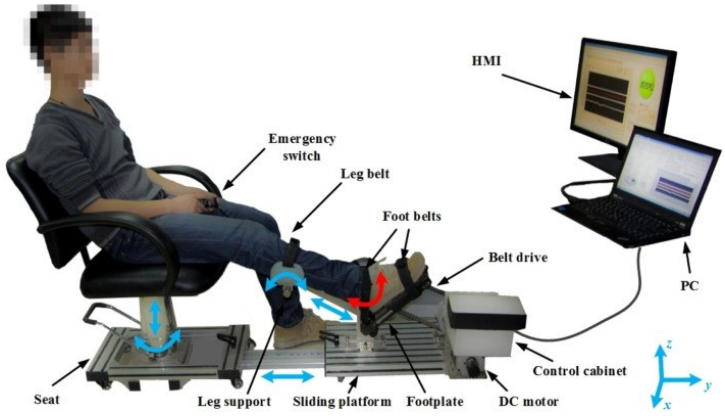


Figure 3.3. Ankle Robot Prototype [36]

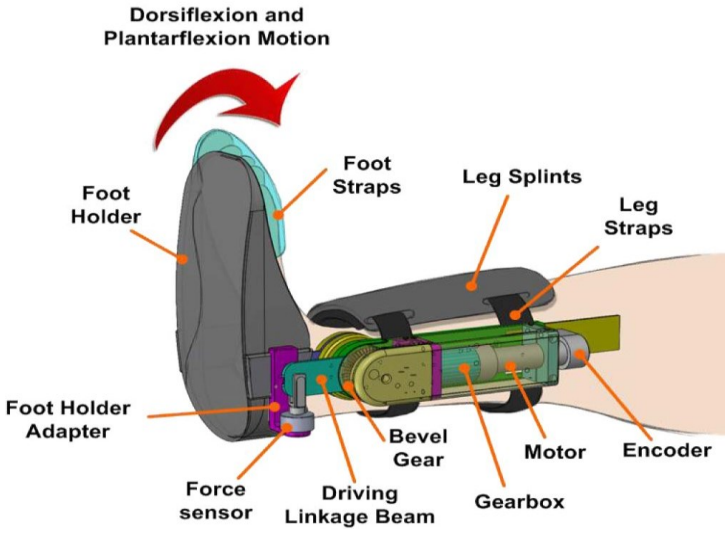


Figure 3.4. Wearable Ankle Robot Prototype [18]

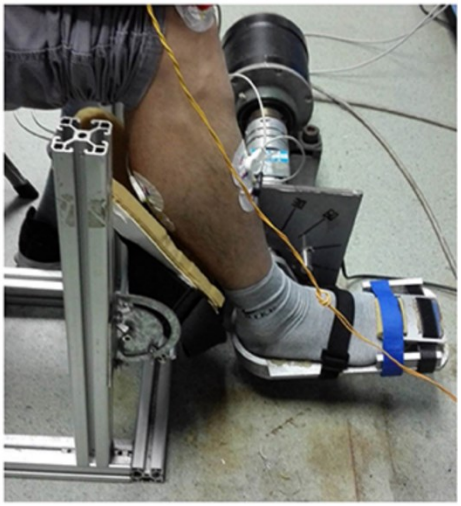


Figure 3.5. Ankle Robot [109]

While reviewing the platform robot-assistive device, no mechanical braking systems were exclusively disclosed. Software-based braking systems were proposed [36] as a 'kill switch' mechanism that activated when the platform moved outside its programmed parameters. The only mechanism proposed was a mechanical limit switch connected to an actuator and the platform's underside, which controls ROM and the platform's physical positioning. Homma and Usuba [107]inserted screws behind the footplate to restrict its movement. For this method, there was no evidence of testing or practical application of braking in the dorsiflex and plantarflex directions.

3.3 USER-BASED DESIGN CONSIDERATIONS

Currently, no rehabilitation robots can compete with human therapists when it comes to assisting patients with real-time assessments and feedback. The design's success depends on several factors, such as the development of rehabilitation engineering to create effective assistive devices. Equally important are enhancing the user experience and developing a reliable robot-human interface for healthcare providers and patients. Any technological advancement that relies on a human interface is inextricably linked to risk and unpredictability. There is a possibility that patients will not be motivated to use robot-based rehabilitation technology if it is not engaging.

Federici and Scherer [110] have proposed 12 criteria for evaluating and testing user engagement by evaluating robot and patient control strategies, end-user comfort, and the safety of assistive rehabilitation technology. The 12 criteria are mentioned below:

- How straightforward is it for therapists to learn?
- Simple to start and adjust the settings of the robot.
- A design that is universal and has a long lifespan.
- Less time-consuming than conventional therapy methods.
- Functionally dependable and solid.
- Adjustable to various tasks.
- Adaptable to the height and length of the patient.
- Capability to provide clear instructions.
- Capability to measure, track, and record patient progress.
- Portability and usability in existing areas.
- Can be operated independently by patients.
- Meets Privacy guidelines.

Successful use of the technology by the patient is contingent upon meeting the mentioned conditions. Robot-assisted physiotherapy focuses on enhancing system functionality and rehab time. Although the most effective way to ensure technological success is to consider the psychological effects on the user and integrate psychological strategies to enhance human-robot engagement and treatment outcomes, it is essential to consider the psychological effects on the user. A review by Langer et al. [111] proved that one's cognitive mood improves due to physical independence, which means that successful post-injury treatment also improves rehabilitation outcomes. The primary concern is safety; other highlighted areas include the robot's ease of use, control, effectiveness, and the learning curve required to operate it. Human-machine interaction can be improved by providing a visually appealing interface and allowing for natural movement between humans and robots. In addition, robot performance, reliability, failure rate, attributes, proximity, and anthropomorphism contribute significantly to the development of robot-human trust. Humans express themselves through personalisation, whereas robots typically have a one-dimensional personality, so patients do not typically interact with robots. Customizable features would enhance patient comfort and familiarity with the assistive device in the context of this project.

The safety of any assistive technology is a significant concern, primarily due to the associated fear and lack of knowledge regarding robot operation. There is a

psychological fear that the physical interaction between a human and an ankle rehabilitation robot will result in the misalignment of joint axes. Safety controls range from high-level software control strategies to low-level control strategies that limit the forces transferred to the patient during range-of-motion (ROM). High-level controls are algorithms that provide patients with varying degrees of movement assistance. Safety measures are the application of control strategies to mitigate foreseeable risks to software, electrical, and mechanical safety components. This includes software oversight of the platform's torque, velocity, and displacement, as well as real-time monitoring. Safety measures, controls, and risk mitigation are crucial to maintain human-robot trust and prevent further injury.

For effective rehabilitation, treatment must be individualised such that it suits the needs and capabilities of the patient and is adaptable between patients. The platform, seating position, and leg support should also be adjustable to meet each patient's unique rehabilitation needs and prevent further injury. Regarding robot operation, the robot's trajectory parameter for velocity, amplitudes, and the number of ROM repetitions should be simple to assign and adjust. There must be a balance between adjustability and comfort in order to meet individual requirements. The comfort factor refers to a suitable seating position, lumbar support, individual adjustability, and cushioned leg and foot pads to support the human-robot interface. Minimising distractions, enhancing concentration and relaxation, and reducing stress aid in better patient rehabilitation.

3.4 CORRELATION BETWEEN PATIENT RECOVERY AND PATIENT POSITION

The patient's optimal recovery is the fundamental requirement of robotic rehabilitation; therefore, patient positioning is crucial. For instance, patients suffering from a stroke have altered body alignment due to certain muscle conditions. Due to this altercation, a patient's position with respect to the robot will play a crucial role in maximising recovery. The optimal seating position can prevent complications, modulate muscle tone, reduce swelling, and prevent muscle atrophy and numbness-related complications [112]. Improving the robot's design to encourage patients to participate in periodic rehabilitation results in a quicker recovery [113]. As patients suffering from a stroke can only maintain a position for a limited amount of time, a patient's comfort is essential for successfully

completing a task. In addition, the patient's seating position should be straightforward and not cause any strain as the patient enters and exits the robot, and it should be able to mimic positions that people typically assume in daily life. The thighs and buttocks support 75% of the seated weight, 19% by the feet resting on the floor, and 4% and 2 %, respectively, by the back and arms [112].

One way to facilitate a speedier recovery and reduce risk is by providing seating options that are comfortable and supportive. The ideal robot design should be easy to clean, require minimal space, and be ergonomic. This chapter proposes a design for rehabilitation while a patient is seated. At times, securing a chair in place can be overwhelming for both the staff and the patient. Pressure from sores, which are common in immobile patients, is an additional factor to consider. This discomfort can be remedied by covering the chair with removable cushions.

3.5 ELECTROMECHANICAL PARAMETERS OF THE ROBOT

The robot uses a Maxon brushless DC motor accompanied by a torque sensor, encoder, and gearhead. The motor was custom assembled as per the requirement by Maxon Group for the purpose of rehabilitation. A basic housing has been built, as shown in Figure 3.6, to keep the motor in a steady place.

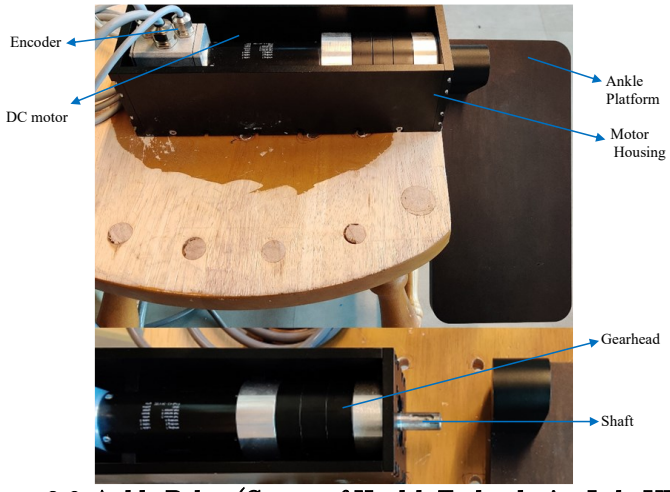


Figure 3.6. Ankle Robot (Center of Health Technologies Lab, UTS)

3.5.1 Motor

The motor is responsible for producing the majority of the shaft's rotational torque. The motor is coupled to a gearhead for increased torque and stability. A brushless DC motor with a 45mm diameter, 250 Watts of rated power, 24 volts of operating voltage, a maximum current draw of 7.15 Amps, and a weight of 1.15 kilograms has been used. It can supply a constant torque of 45.5 mNm/A and has a nominal torque of 331mNm. This project employs the EPOS2 70/10 motor controller, and the MAXON EPOS software and MATLAB are used to adjust motor parameters such as velocity, initial position, current, and motor tuning. The motor can maintain a specific ankle dorsiflexion and plantarflexion position for a predetermined period.

3.5.2 Encoder

The motor is equipped with a Maxon HDEL 9140 encoder, which is a high-precision encoder. It consists of a DC tachometer that provides feedback on the robot's current rotational speed and position. The encoder contains an electromechanical element that converts the angular position of the shaft into a digital output, which aids in measuring the rotation angle. The encoder chosen for this project has a position accuracy that is two to four times greater than required. Two signals determine the direction of rotation and the position with phase shifts of 90 degrees. Based on counting plus edges (CPT), the encoder can achieve a four times higher resolution. The encoder reads the position data as quadcounts (qc). The encoder used in this project has a 500 CPT, resulting in 2000qc (4x500) per shaft rotation, corresponding to 0.18 degrees of encoder shaft movement. The robot can move smoothly and precisely because this property is present.

3.5.3 Gearhead

A planetary gearhead has been attached to the motor to increase the torque along the shaft while decreasing the RPM of the motor. Since the motor is utilised for rehabilitation, the reduced RPM of the motor shaft is negligible and increases the user's safety. This mechanical component increases the motor's torque by the ratio of the gearhead, thereby reducing the motor's cost and size. The gearhead has a diameter of 62 mm and can generate 50 Nm of continuous torque. The

gearhead's output shaft is made of steel, making it extremely durable and capable of supporting heavy loads. It also has a reduction ratio of 236:1 and can rotate approximately at a constant speed of 2500 RPM.

3.5.4 Torque Sensor

A torque sensor converts mechanical torque to electrical signals. In the case of rehabilitation, it permits us to receive feedback on the robot's movement path and performance. The sensor would aid in measuring the patient's torque output. In this scenario, the sensor also provides feedback regarding the patient's ankle's safe extension. DHN – 200 dynamic torque sensors with a torque capacity of 10,000 Nm are used for this purpose, as depicted in Figure 3.7. The torque will be positioned in front of the gearhead, and the sensor will be attached to the ankle platform.

Figure 3.7. Dynamic Torque Sensor DHN-200

3.6 PROPOSED MECHANICAL DESIGN

The chapter's proposed design will focus on developing a structure for a 1-DOF ankle robot. The robot's movements will concentrate on two primary movements: plantar flexion and dorsiflexion. As depicted in Figure 3.6, the design will be constructed over the existing housing of the Ankle Robot at the Centre of Health Technologies, UTS. The suggested robot's mechanical components are intended for usage while the patient is seated. This posture was chosen because it is more applicable to day-to-day activities when compared to performing ankle exercises while lying down. The seated position can allow for weight-bearing exercises that increase the ankle's movement stability.

3.6.1 Design of Leg support

While performing various rehabilitation exercises on a robotic platform, it may be difficult for a patient to support their leg weight for an extended time. Consequently, the leg support is intended to support the patient's leg during rehabilitation exercises. The support is typically placed on the calf muscles and can be altered based on the physiotherapists and patient's requirements. Figure 3.8 depicts the proposed leg support with three degrees of freedom. The 3 DOF is comprised of three separate 1 DOF moving parts in different planes. The maximum length of the stand is 365mm, and when seated, this can be altered to reduce the strain on other body parts during rehabilitation.

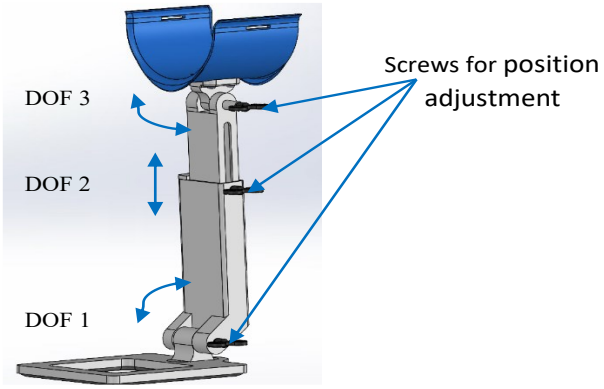


Figure 3.8. Proposed Leg Support for ankle robot.

The proposed design is easily maintainable and adaptable to the needs of the patient by the therapist. For various patients, the leg support can be adjusted for height. In addition, the leg support is not rigid and glides slightly with the leg. This allows the ankle to move freely. The screws positioned at various DOF can be simply modified by therapists to meet the demands of each patient. A 55mm screw with an end cap is used to adjust the platform. The end plate rotates 30 degrees around its axis when connected to the pin. The plate acts as a clamp, allowing the components to move freely in the desired direction of motion. A washer has been added to prevent the screws from penetrating the stand. As depicted in Figure 3.9, the bottom of the screw plate is indented to increase grip and prevent slipping.

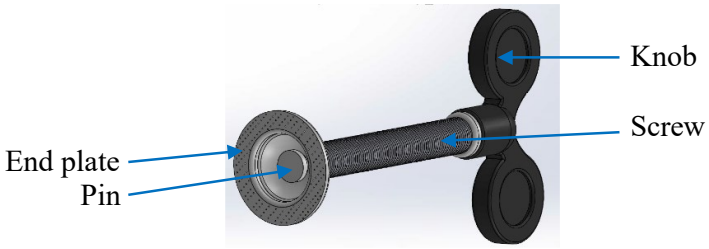


Figure 3.9. Screw

The DOF 1 shown in Figure 3.8 is connected to the casing that houses the stand's base via an adjustable screw. This section's movement is restricted to the x-plane and is limited to a 210-degree angle. Between the base and the ground, clearance has been granted for the full rotation of the screw. During the exercise, the base would be secured in a specific location. To ensure the stand's stability, it should be constructed from steel. A thin layer of rubber is placed underneath the stand to ensure that it does not move during operation and to dampen any vibrations that may occur during the robot's operation.

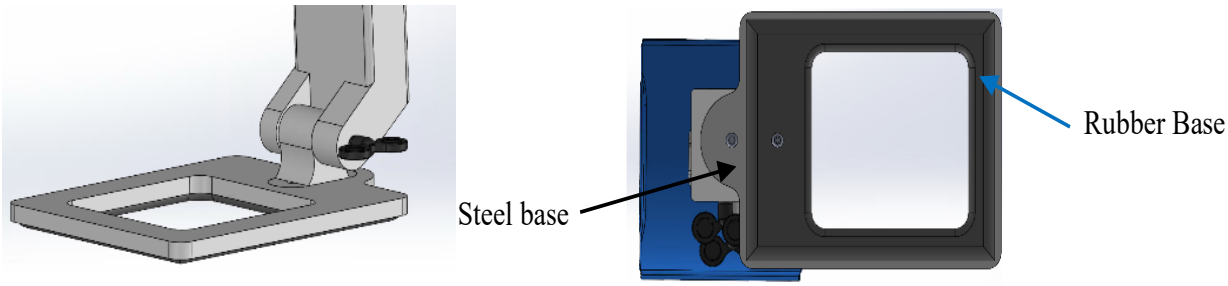


Figure 3.10. DOF 1 & Base of Leg Support Stand

The next component of the stand (DOF 2) is a height-adjusting mechanism. This component is designed only to move vertically and is primarily used for height adjustments. The vertical portion of the stand moves through a 90mm slot cut into the stand so that a screw can be used to set the desired height. Nuts and washers are employed to secure the screw to the stand. The vertical component would be constructed from aluminium to improve the stand's stability and longevity.

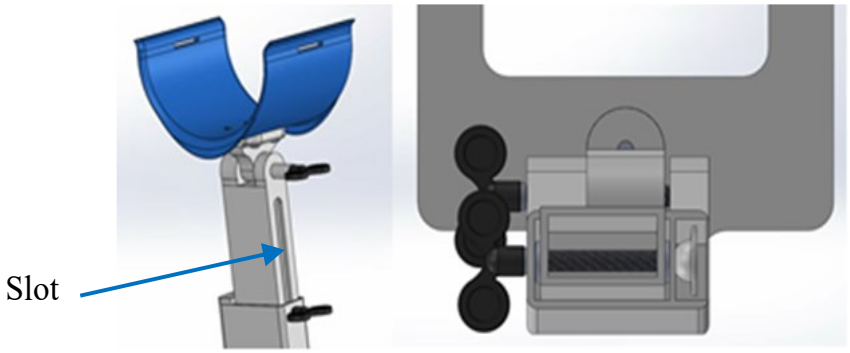


Figure 3.11. DOF 1 movement mechanism

The final portion of the stand is a leg case housing that will support the patient's leg (DOF 3). It is primarily designed to support the calf muscle but can be placed anywhere along the leg as needed. The plastic casing will have a cotton insert and internal foam padding to secure the leg in place. The platform can rotate 210 degrees in either direction, making it flexible and reducing patient strain. In addition, this facilitates the patient's entry and exit from the system. The leg case is intended to support any knee flexion caused by rehabilitation. The case is contoured to accommodate legs up to 150mm in length.

Figure 3.12. DOF 3

The likelihood of a stand deforming due to repeated use and the weight of a person's leg is deemed to be extremely low due to the high tensile strength of aluminium and the fact that the stand would only be used for a maximum of 30 minutes during rehabilitation exercises. The total weight of the stand is approximately 6 kilograms based on the chosen material and its surface area.

3.6.2 Platform Design of the ankle Robot

A platform for the ankle should be comfortable for the patient to wear and should move precisely during exercises. The chapter's proposed design discusses the

ankle platform, a connection between the platform and the motor shaft, robot housing, and a simple braking mechanism. The intended platform will be separate from the leg support. After adjusting the foot platform to the desired position, it can be secured with a swing key and twist lock mechanism. The foot plate is intended to be aligned with the ankle axis. Figure 3.12 illustrates a CAD model of the robot platform.

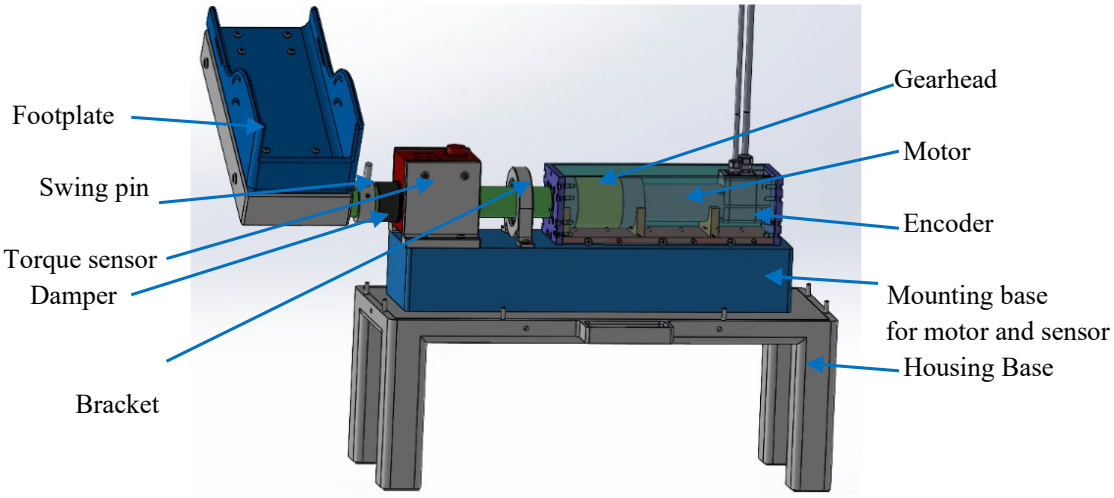


Figure 3.13. CAD model of Ankle Robot Platform

3.6.2.1 Ankle Footplate

The current ankle footplate was made of 10 kg of solid metal. Having this much weight on the footplate would make it difficult for the patient to raise their foot. The footplate was redesigned with a lightweight aluminium frame, a cross-sectional centre, and plastic for the remaining portions to address this issue. The design is based on bicycle pedals because they can withstand extremely high forces.

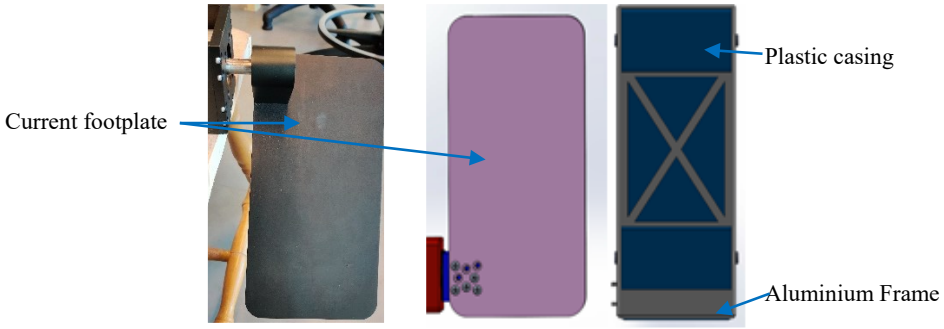


Figure 3.14. Proposed new Footplate

The casing of the footplate is intended to maintain the ankle's static position relative to the motor shaft. This footplate allows the ankle to move the joint to its properly aligned biological axis. The patient's foot can be secured by the dorsal foot and heel, preventing misalignment as the footplate rotates. A series of holes have been drilled into the footplate in order to accommodate variations in musculoskeletal properties. The plate would be comprised of a cotton fabric that can keep the foot in a stationary position, thereby increasing patient comfort and preventing involuntary movement. Figure 3.15 illustrates an example of this cotton fabric, as described in Physio. Due to the low tensile strength of the aluminium frame, there would be a risk of deformation when attaching the footplate to the torque shaft. This risk would be minimal, as both hardware and software limitations would prevent the ankle from becoming overextended. As this footplate is used for rehabilitation, the voluntary forces applied on the device by the patient would be minimal. In addition, the patient's foot would transfer its weight and force to the platform. The footplate is sufficiently light to avoid resistance during the required rotational movements. A solid bottom is accounted for in the footplate's shaft base pins to suppress additional weights while patients place their ankle on the platform. Based on the selected materials, the estimated weight of the footplate is 1.5 kilograms. Patients will not need to exert additional effort to move the foot pedal. In instances where the patient cannot move the ankle, the motor would be used to provide the necessary assistive torque to move the ankle.

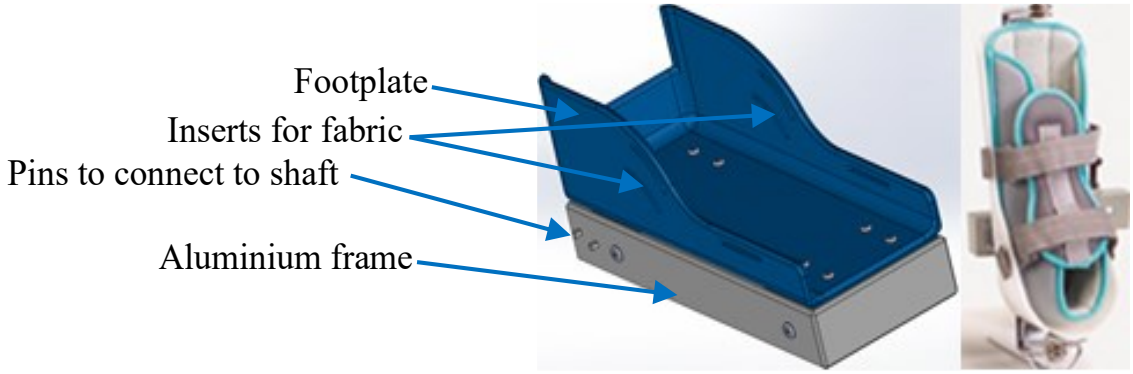


Figure 3.15. Ankle footplate (left) and reference fabric (right) [114]

3.6.2.1 Shaft Design

The connection to the torque sensor shaft must be adjusted so that the robot's platform has a balanced weight distribution and a precise centre of gravity. The connection between the two points minimises the torque transmitted to the

platform via the shaft connector. The neutral position of the footplate is secured to the platform by a simple twist-lock mechanism. The mechanism is comprised of two pins that are twisted into the footplate's aluminium frame. The torque sensor's shaft is coupled to the twist lock mechanism. Clearance between the torque sensor and the two pins on the footplate has been provided to prevent friction and minimise torque transfer error. Steel will be used to construct the connecting shaft, enabling it to perform well under load. For accurate movement measurements, the centre of rotation of an ideal footplate should be aligned with the ankle joint and torque sensor. Nevertheless, there is a 60-degree offset from the footplate with respect to the ankle joint. The precise offset of the ankle joint would vary among patients based on their unique musculoskeletal characteristics and ankle location. Based on the patient's flaccid natural ankle joint position, the footplate's neutral position can be changed.

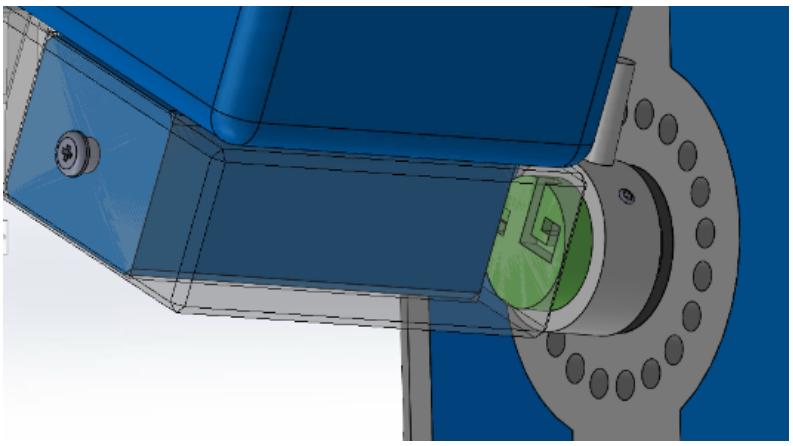


Figure 3.16. Twist Lock Mechanism

As shown in Figure 3.12, a shaft is also connected between the mounted gearhead and the torque sensor. This shaft must be capable of withstanding rotational forces and bending due to the torque sensor and motor gearhead. Due to the possibility of bending in the shaft, it would be constructed from mild steel, which has a density of 7.85g/cm3 and a long lifespan. The shaft is inserted through a bracket, fitted with a ball bearing to increase its stability. The ball bearing will allow for smooth rotation and serve as a support between the torque sensors and the motor gearhead. This would increase the system's stability without restricting shift movement and prevent the robot's shaft from bending at that location. The shaft would be lubricated to ensure longevity, prevent wear and tear, and reduce friction in order to guarantee a satisfactory operation. Dampers have been installed throughout the robot and the shaft to reduce vibrations that

may occur during the robot's operation or while the patient is undergoing rehabilitation. In addition, dampers have been installed on the mounting base where the electronic components are stored. For accurate torque measurement and to ensure clearance between the torque sensor and the braking pins, a damper has been placed at the end of the shaft between the torque sensor and the footplate. All components have been attached to an ABS mounting base using 5mm and 6mm screws. It has been ensured that there are no loose parts or components, so the system is more stable, and the robot can be moved around the workplace easily.

3.6.2.1 Braking System

The robot is fitted with a simple, low-level mechanical braking system that will stop the footplate from moving beyond the desired position. Even though there are software-based control strategies for braking, a mechanical system increases the user's safety and provides visual reassurance to both the patient and the medical professional. This project's proposed braking system is only concerned with position control. Ankle stretching can cause injury to the patient and reduce engagement with the robot. The design of the brakes resembles a gym weight machine that uses pins to select the proper weight for each individual. This project's braking design incorporates a rotating vertical pin connected to the shaft. In addition to the rotating vertical pin, two horizontal pins that can be inserted into the holes of the divider surrounding the shaft are added. Inserting the pins into the divider pin holes around the shaft can prevent excessive ankle plantarflexion and dorsiflexion extension movement.

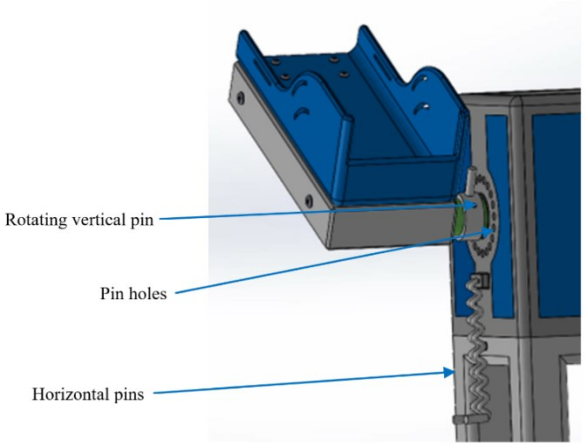


Figure 3.17. Braking Mechanism

To prevent the loss of the pins, the horizontal pins are connected to the external housing of the platforms via springs. The pins have a diameter of 8.5mm, which is suitable because the pins and pin holes are made of mild steel to ensure that they do not crack or move while engaged. If the pin diameter had to be increased, the number of pin holes on the divider would be reduced, reducing the number of available brake positions for individual patients. A damper placed behind the pin holes allows the horizontal pins to enter the torque sensor without interfering with its movement. The braking system would not interfere with the ankle's natural movement during rehabilitation exercises. Due to the visual reference, the proposed braking system provides enough time to stop the system quickly. Due to the system's small size, the amount of maintenance required is relatively low. The horizontal pins are only required to be replaced if there is significant wear and tear.

3.6.3 Housing for Ankle Robot

For patients to feel less intimidated, a hospitable environment is necessary. It also provides dust and moisture protection for components such as electronics and shaft fixtures. The housing case and divider are designed to be easily disassembled for maintenance and repair of the components. The top of the housing case features a detachable window for viewing the torque sensor data and inspecting the status of the shaft while the robot is in operation. The case is comprised of a steel frame with ABS plastic sheets surrounding the housing. Mild steel was chosen because it is strong and would not bend or deform easily under external weight and forces or shaft forces. Due to the torque and forces generated by the electronic components, the shaft must have a high tensile strength; otherwise, it may penetrate the structure. To reduce the robot's weight and increase its portability, only the cabinet’s frame is made of steel, while the panels are made of 3D-printable ABS.

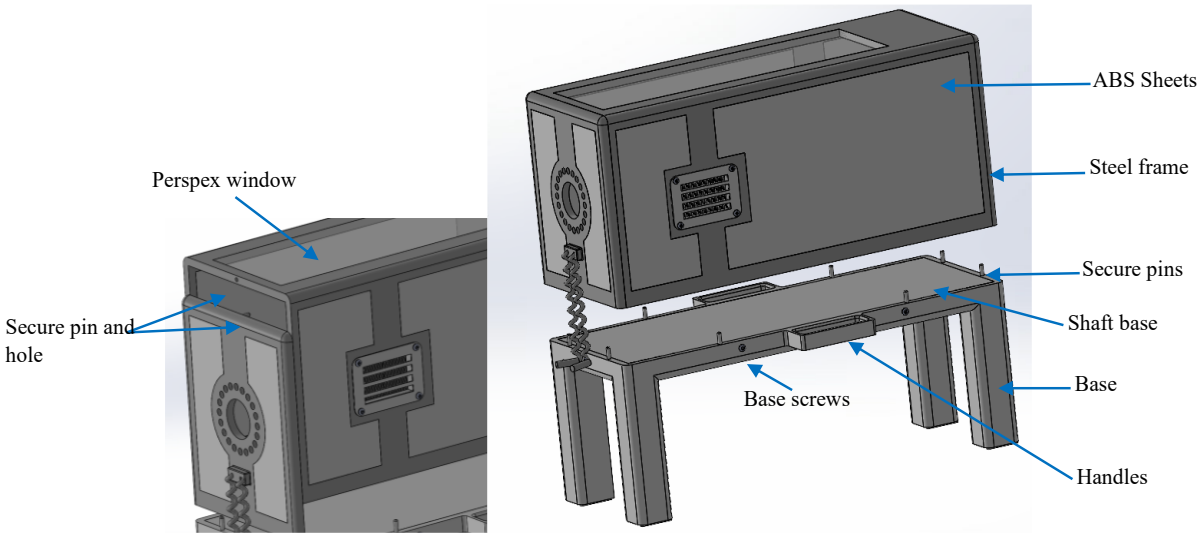


Figure 3.18. Housing case for the ankle robot

The housing base is made of lightweight aluminium and is attached to the housing using screws. Based on the material's density and volume, the housing weighs approximately 6 kilograms. To facilitate the robot's portability, handles were attached to the base rather than the housing. If handles were added to the case, there would be a chance that the robot's base would fall off while being transported. The base aids in keeping the footplate off the ground, thereby providing sufficient space for the footplate to move. Under the four legs of the base, rubber pads have been installed to act as a damper while the robot is in motion and prevents it from slipping. The housing was designed with a ventilation system to prevent electronic devices from overheating. Placed on opposite sides of the shell are two microfiber ventilation eyelets. As shown in Figure 3.19, a fan is attached to one vent to draw in cool air, while hot air must be exhausted through the other vent. A compartment for storing loose electronic cables is located on the case's rear. A spring, which is also a component of the steel frame housing, connects the brake system mounting pins to the housing.

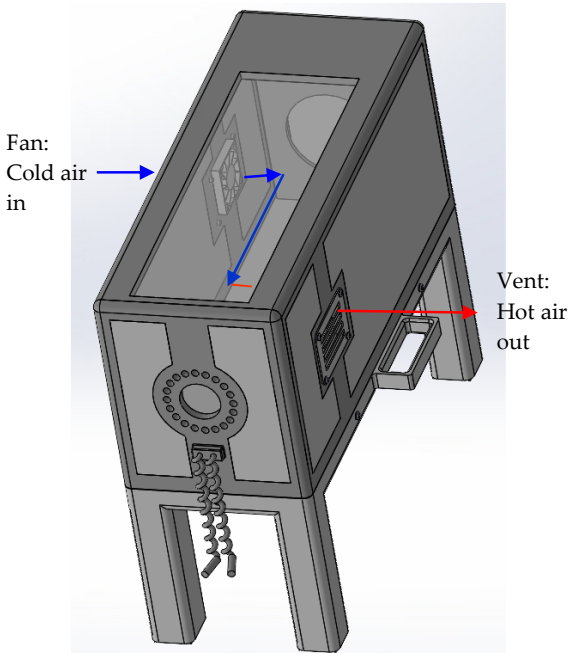


Figure 3.19. Ventilation System for Housing

3.7 CONCLUSION

Due to the unanticipated situations brought on by the COVID-19 pandemic, the proposed design was not implemented. Therefore, it is impossible to test the quality of the proposed design. The presented proof of concept of the robot's mechanical structure can aid in enhancing the relationship between humans and robots. The proposed leg support and braking system can enhance the user's comfort level, thereby enhancing the efficacy of robotic rehabilitation. The proposed design expands the scope of assistive technology applications beyond neurological injury and paves the way for the robot's successful development. The dimensions of the designed parts are mentioned in the appendix of this thesis.

4 ANKLE ROBOT INTERACTION USING EMG SIGNALS

4.1 INTRODUCTION

Compared to traditional physiotherapy, physiological signals offer several distinct benefits. These benefits consist of an improvement in the precision of the support that is offered and a lessening of the variation that takes place between different repetitions of a movement. In addition to this, it offers quantitative assessments of the patient's level of healing after an accident [6]. In addition, research has demonstrated that rehabilitation robots that operate on several physiological signals, such as a surface electromyogram signal (sEMG), have a huge capability for enhancing the outcome of diagnostic operations performed during physiotherapy. Surface electromyography, also referred to as s-EMG, has provided evidence of the advantages that robotic rehabilitation may be able to provide. If the movement intentions of the users can be precisely mapped out, robotic rehabilitation can make great advances ahead in terms of its efficacy. This will allow for considerable gains to be made during rehabilitation. Acquiring EMG data is one of the steps in predicting and understanding muscle movement patterns. Using an EMG signal for movement detection allows us to predict the patient's movement prior to the actual muscle [8, 9]. This property of the signal enables us to determine the patient's intention and assist them as necessary, thereby enhancing human-machine interaction. Additionally, it is important to remember that no single muscle controls any given ankle motion; however, certain muscles are dominant for specific movements.

The first section of the chapter examines a novel classification method for classifying five distinct ankle movements based on s-EMG signals from four different muscles. The resting condition of the ankle is one of the five classes. Several classifiers will be used to classify the data, and their accuracy will be compared with various window sizes. Ankle plantarflexion, dorsiflexion, ankle inversion, eversion, and ankle resting state were selected for classification because they are fundamental ankle movements. Though EMG is widely used for pattern classification, it could also produce noise while the signal is acquired. This noise could result in wrongful classification of the EMG signal, which could result in causing severe injuries to the user. The noise could be due to external factors such as the movement of the sensors, noise created by the sensor's electronic components and electromagnetic interference from nearby devices. One of the disadvantages of using only EMG is that the EMG signal cannot be used as a ground truth to the actual position since EMG cannot provide the actual position with respect to the user intent. To observe the ground truth in this project, we use Inertial Measurement Unit (IMU) sensor to obtain the foot orientation and the actual ankle position.

Along with monitoring foot orientation, IMU signals are collected for classification for five ankle positions. In this chapter, the IMU sensor is primarily used as a safety feature while the user interacts with the robot. The IMU sensor comes to play only if the quality of the EMG signal is below the specified threshold.

During therapy, the ankle robot cannot be moved into the desired position based only on the user's intent. The classifier helps us determine where the ankle desires to go, but it does not indicate how far the ankle desires to move within the targeted movement intention. To obtain the position from EMG signals, we need to estimate the torque and forces associated with EMG signals. To better grasp the mechanics of ankle torque and the forces associated with it, it is essential to understand how muscles produce force. Constructing a muscle model using EMG data pertaining to ankle dynamics presents several challenges. Before estimating torque, it is essential to understand how muscle contractions work. When a person intends to move their muscles, a neural signal is sent to those muscles, which causes the motor units in the muscle fibre to become excited. Because of this stimulation, the muscles around the joint will begin to move. There are several motor units in each and every muscle. Each stimulation will bring about a twitch in the muscle's contracted state, resulting in a mechanical output. The electrical

activity measured from the motor unit through the muscle is known as the EMG signal. As a result, the EMG signal's intensity will increase proportionately to the number of motor units activated at that particular point.

Several different dynamic models have been put forward over the course of time in order to predict the forces that are produced by the muscle of a moving joint. Ankle joint forces, however, are harder to measure due to the ankle joint's indeterminate nature. Since it is unable to consider variations in individual muscle activation patterns, a linear objective function cannot be used on its own to determine muscle forces. Over recent years, biomechanical models have been developing to provide a mathematical model of muscle mechanics; one such model is the Hill-based muscle model [11, 115]. There is still active usage of these models, and they are very pertinent to the current subject. Electromyography (EMG) and a suitable biomechanical model are, nonetheless, the preferred method for estimating muscle force [8, 9].

Researchers have attempted to quantify the forces created by the muscles by using the EMG signal in combination with a suitable mathematical-based biomechanical model [116]. Numerous models, like the Hill-based model, which is a mathematical model of the mechanical working of a muscle, have been created in recent years [117, 118], but due to individual differences in EMG signal, it greatly magnifies the estimation inaccuracy. The Hill Muscle model requires several inputs, such as muscle load, length of the muscle fibres, joint position etc., to estimate the torque from EMG signals. Since these variables depend on the user, the model's variables should be changed for torque estimation. In robotic assistive technology, numerical methods (including convergence, first-order method, newton-Raphson method, and bisection methods), optimisation, and even machine learning are all emerging as ideal problem-solving tools. Several experiments have shown that nonlinear optimisation problems may be solved using techniques such as the Particle Swarm Algorithm (PSO), Levenberg-Marquardt, Simulated Annealing, and genetic algorithm [23, 119]. An EMG envelop obtained after being filtered using full-wave rectification look quite similar to muscle tension curves [115]. An EMG signal-based control for an ankle rehabilitation robot is depicted in this chapter. Based on the electromyographic signal, this method uses a minimum number of constraints to make predictions about the joint forces (torque and angle).

This chapter proposes a nonlinear mathematical model for estimating joint torque. Swarm techniques are used to find the variables of the mathematical equation for each ankle movement pattern. The ankle's torque and the plantarflexion and dorsiflexion angles may be estimated using these models. During the rehabilitation process, once the intended movement has been determined, the activation functions that have been retrieved from each EMG channel will provide an approximation of both the planned and actual movement. This may be done after the intended movement has been established.

4.2 METHODS AND MATERIALS

The s-EMG signal was collected from five healthy subjects. Four distinct muscles' s-EMG signals were collected to identify five distinct ankle states. Based on the review of the relevant literature, s-EMG data was collected from the tibialis anterior, tibialis posterior, peroneus brevis and gastrocnemius medialis muscles. The EMG signal was acquired at a frequency of 2000Hz from Delsys Trigno Avanti s-EMG sensors. The first sensor was attached to the tibialis anterior, the second to the gastrocnemius medialis, the third to the tibialis posterior muscle, and the fourth to the peroneus brevis muscle. The subjects were asked to perform the ankle movement with rest intervals between each task. The subjects were given sequential instructions for completing the task. Each movement task was performed for five seconds, followed by a five-second rest period. During the subject's rest period, they were informed about the next movement. It was ensured that the participants had started the experiment when resting in a comfortable neutral position. In each session, four movements were collected once, with a resting period after each movement. The session has repeated four times, with five to ten minutes of rest between each repetition.

Figure 4.1. Protocol for s-EMG data collection

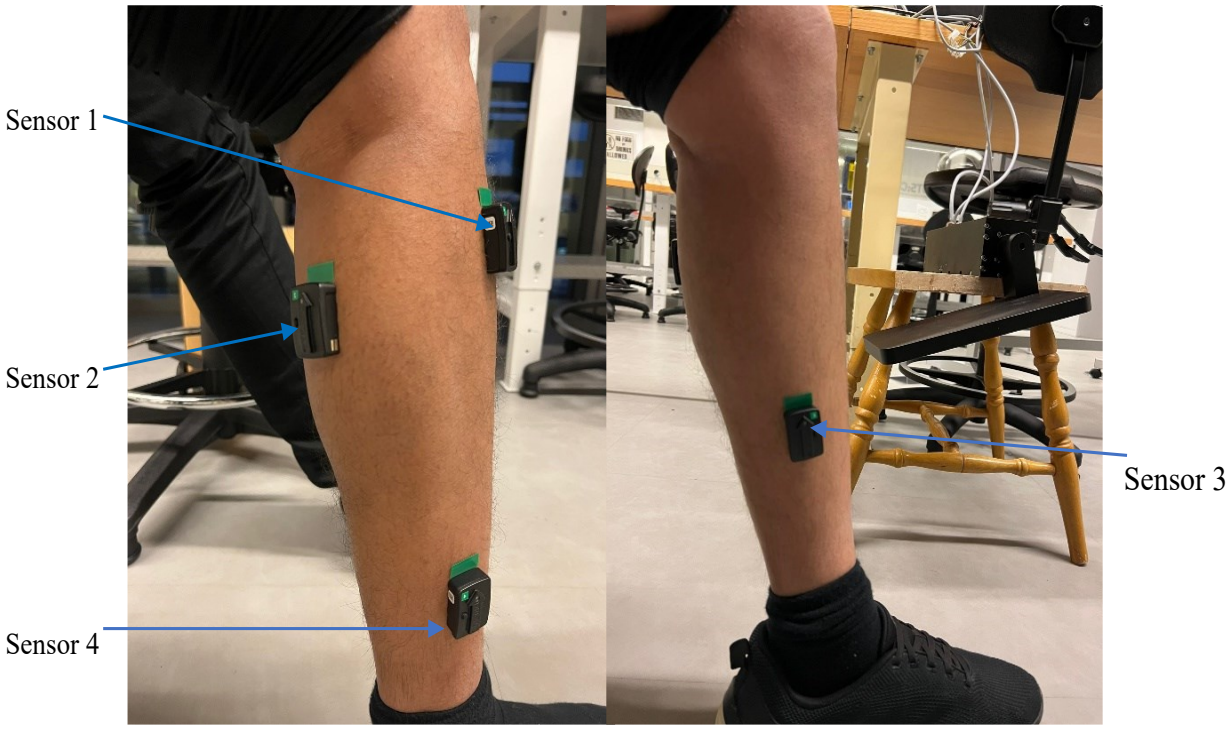


Figure 4.2. s-EMG Sensor Placement

Multiple muscle movements are observed for a single movement, as depicted in Figure 4.3. However, specific movements are dominated by particular muscles. It has been observed that the tibialis anterior muscles are primarily responsible for ankle dorsiflexion and inversion of the foot, while the peroneus brevis is primarily responsible for ankle eversion. During ankle plantarflexion, the dominant muscles were the gastrocnemius medialis and peroneus brevis. Each subject's voluntary muscle contraction during each session varied. The collected data was noisy, so a 5th-order Butterworth band pass filter from 20Hz to 300Hz was applied to reduce the noise level. A DC offset was observed while analysing the signal by taking its FFT. The mean values of the signals were subtracted from the original signal to eliminate the offset.

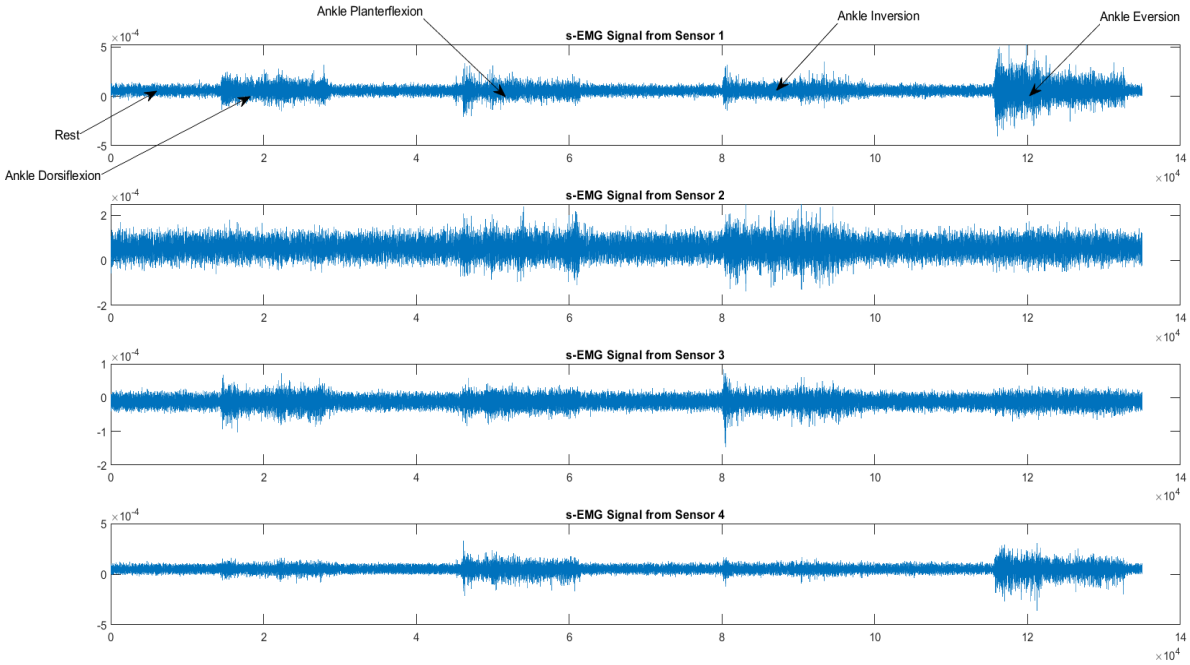


Figure 4.3. s-EMG data collected from its respective muscle

The IMU sensor was attached to the foot near the toes to classify IMU signals. The placement of the IMU sensor is shown in Figure 4.4. The IMU sensor used in this project was developed at UTS, and the signal was recorded at a sampling rate of 23Hz. Figure 4.1 depicts the data collection procedures followed when gathering IMU data. For classification purposes, only accelerometer values were gathered from the IMU. One of the issues identified with the IMU signal was the drifting of the signal. The output of an inertial measurement unit (IMU) deviates from its actual value over time. Sensor errors including bias, scale factor, and misalignment caused this drift. To reduce the errors caused by the drift, the mean value of the acceleration data was subtracted from the acquired data. The data was then filtered with an 11Hz Butterworth low filter of the first order. On the other hand, along with the accelerometer data, gyroscope data was used to observe the ankle orientation. The "imufilter" MATLAB function was used to determine the ankle orientation.

Figure 4.4. IMU sensor placement on the foot

4.2.1 Data Segmentation and Feature Extraction

As the robot will be used in real-time, the filtered data must be segmented to a suitable window length for faster and more accurate classification. In real-time applications, a small window size can produce faster classification results, but at the expense of accuracy. While a larger window can improve precision, generating an output may take a considerable amount of time. According to studies, the window should be limited to a maximum of 300 milliseconds to optimise computational efficiency and increase the response speed of the classifier and the robot [63, 120]. After identifying the user's intent, the studies indicate that the robot should respond within 300 milliseconds. Maintaining this restriction ensures that the ankle robot will move rapidly once the user's intent has been determined. Consequently, the window width is an essential parameter for real-time classification. In this chapter, we will examine the accuracy of different window sizes (50ms, 100ms, 150ms, and 200ms) and compare the accuracy of various

classifiers. Due to the fact that the contraction and release times of each movement were distinct, a 50% overlapping window has been used to segment the data. The overlap contributes to the expansion of the database used to train the classifier, thereby improving its accuracy.

Feature extraction enables us to observe more signal details and information, which is essential for pattern recognition. The majority of s-EMG signal characteristics are extracted from the time domain because they do not require signal transformation. For statistical analysis of the signal, five time-domain features are employed.

- RMS (Root mean square) measures the signal's amplitude. It is proportional to the force with fatigue-free muscle contraction and is modulated by the Gaussian random procedure. The RMS equation is as follows:

$$RMS = \frac{1}{N}\sum_{k=1}^{N} x_k^2 \tag{4.1}$$

where,
N is the length of the data
x_k is the signal.

- The next feature is Standard Deviation (SD) which is closely related to RMS. It calculates the variation in muscles contraction over a set period. It gives The equation of SD is given below:

$$SD = \sqrt{\frac{\sum |x-\mu|^2}{N}} \tag{4.2}$$

where,
x is the value in a data set
N is size of the data
μ is the mean value of the data set.

- Mean absolute value (MAV) is frequently used in statistical analysis. Since s-EMG signals contain both positive and negative values, employing MAV allows the complete reversal of negative values to positive ones. It

computes the s-EMG signal's average absolute value. It provides information about the average level of muscles contraction. The following equation defines MAV:

$$MAV = \frac{1}{N}\sum_{n=1}^{N} x_n \tag{4.3}$$

where,
x_n is the EMG signal.
N is the total sample size.

- Skewness is another characteristic that examines the symmetry of the data. If a given data is symmetrical, then its value will be zero. The skewness equation is given below:

$$Skewness = \frac{\sum_i^n (x_i - \bar{x})^3}{(N-1) * s^3} \tag{4.4}$$

where,
x_i is the random variable in the data
$\bar{x}$ is the mean of the distribution
N is the size of the data
s is the standard deviation.

- The final feature utilised is kurtosis, which measures the complexity of s-EMG data. It determines whether the mean point of the signal is flat or has a peak. Lower kurtosis values indicate a relatively flat signal, whereas higher values indicate a signal with more peaks. This feature will help to determine the amount of variability in the signal for a particular movement. Below is the equation for kurtosis:

$$Kurtosis = \frac{\sum_{i-1}^{N} (x_i - \bar{x})^4 / N}{s^4} \tag{4.5}$$

where,
x_i is the s-EMG data
$\bar{x}$ is the mean of the distribution
N is the size of the data
s is the standard deviation.

By looking at the selected 5 features, it is possible to find patterns of muscle activation that are unique to a certain task or movement. Adding more features would increase computational time and could also decrease classification accuracy. Though there are several other features in the literature for EMG classification, the data from these features is enough to classify the ankle movement accurately.

Since IMU signals are very distinct, only a few features are required for classification. Here we use skewness and kurtosis as described in equation 4.4 and 4.5 as features for IMU classification. The IMU data was segmented with a window size of 100ms with a 50ms overlapping window.

4.2.2 Classification

After signal features have been extracted, each sensor's signal features are fed through the classifier. Class number one is assigned to the resting state of the ankle, class number two to dorsiflexion, class number three to plantarflexion, class number four to inversion, and class number five to ankle eversion for both IMU and EMG classification. The classifier must accurately classify the intended movement to actuate the ankle robot in the correct direction. We propose using a cubic SVM (Support Vector Machine) classifier in this chapter. SVM is a classifier that employs a hyperplane between various data points. In three-dimensional space, SVM uses kernels to create a hyperplane and decision boundary. A multiclass Cubic SVM with a degree of the third-order polynomial kernel is utilised. The proposed kernel function equation is shown below:

$$K(s_i, s_j) = (s_i \cdot s_j + c)^3 \tag{4.6}$$

where,

s_i, s_j is two different points.

c is a constant.

Due to the use of four sensors and the extraction of five features from each sensor, a total of twenty features are available for classification. The extracted features are combined with the classes to form a vector, which is then fed into the classifier. The features are trained using the proposed multiclass SVM, which uses a One-Versus-One classification strategy with automatic kernel scales. The performance and precision of the Cubic SVM will be compared to those of other widely used classifiers, including linear SVM, quadratic SVM, LDA, k-NN, and

Naive Bayes for EMG classification. Additionally, for EMG classification, the data was classified with five different window lengths (50ms, 100ms, 150ms, 200ms, and 250ms) for each classifier to evaluate each classifier's performance and accuracy. Following offline testing of classifier accuracy, the classifier with the highest test accuracy was implemented into the real-time system. Changes were made to the window size in order to improve accuracy and response time.

4.2.3 IMU switching algorithm

In comparison to EMG signal, IMU data can provide the foot's orientation. EMG is used to predict the intention prior to the actual movement, whereas IMU data can only predict during the movement. However, IMU can serve as a reference point by providing the precise position and orientation of the ankle. The orientation data can help us determine whether the ankle is held securely on the ankle robot before or during rehabilitation exercises. This project develops a simple algorithm to use IMU as the control signal if the EMG signal is of poor quality. Figure 4.6 illustrates this algorithm's operation. Initially, IMU signals are collected to determine the orientation of the user's ankle. The robot will only be activated if the ankle remains in the proper position. Due to the movement of the foot during rehabilitation, the position of the ankle could vary. To avoid disabling the robot, an error of 5 degrees from the required position is permitted in all axes. Once the robot is enabled, we start acquiring both the EMG and IMU signal simultaneously. While acquiring the EMG signal, the signal-to-noise ratio is measured to determine the signal's condition (SNR). This ratio is utilised to determine the signal strength of the EMG signal relative to any noise present during data acquisition. A higher ratio indicates a signal with high EMG strength, while a lower ratio indicates a signal with high noise. This ratio is crucial for accurate EMG measurements, as poor signal quality can lead to an inaccurate classification of the user's intent. If the ratio is not able to reach the set threshold, IMU data would be used to control the robot. The signal-to-noise ratio is calculated as:

$$20\log\frac{RMS(contraction)}{RMS(Base\ line\ noise)} \tag{4.7}$$

The DELSYS EMG sensor's baseline noise is 5uV. However, testing revealed 12uV of baseline noise on the EMG sensors. If the signal-to-noise ratio

were less than 1.8db, the EMG signal would be deemed an error signal. This threshold indicated whether the EMG signal generated by the muscle was adequate for classification and the robot's control.

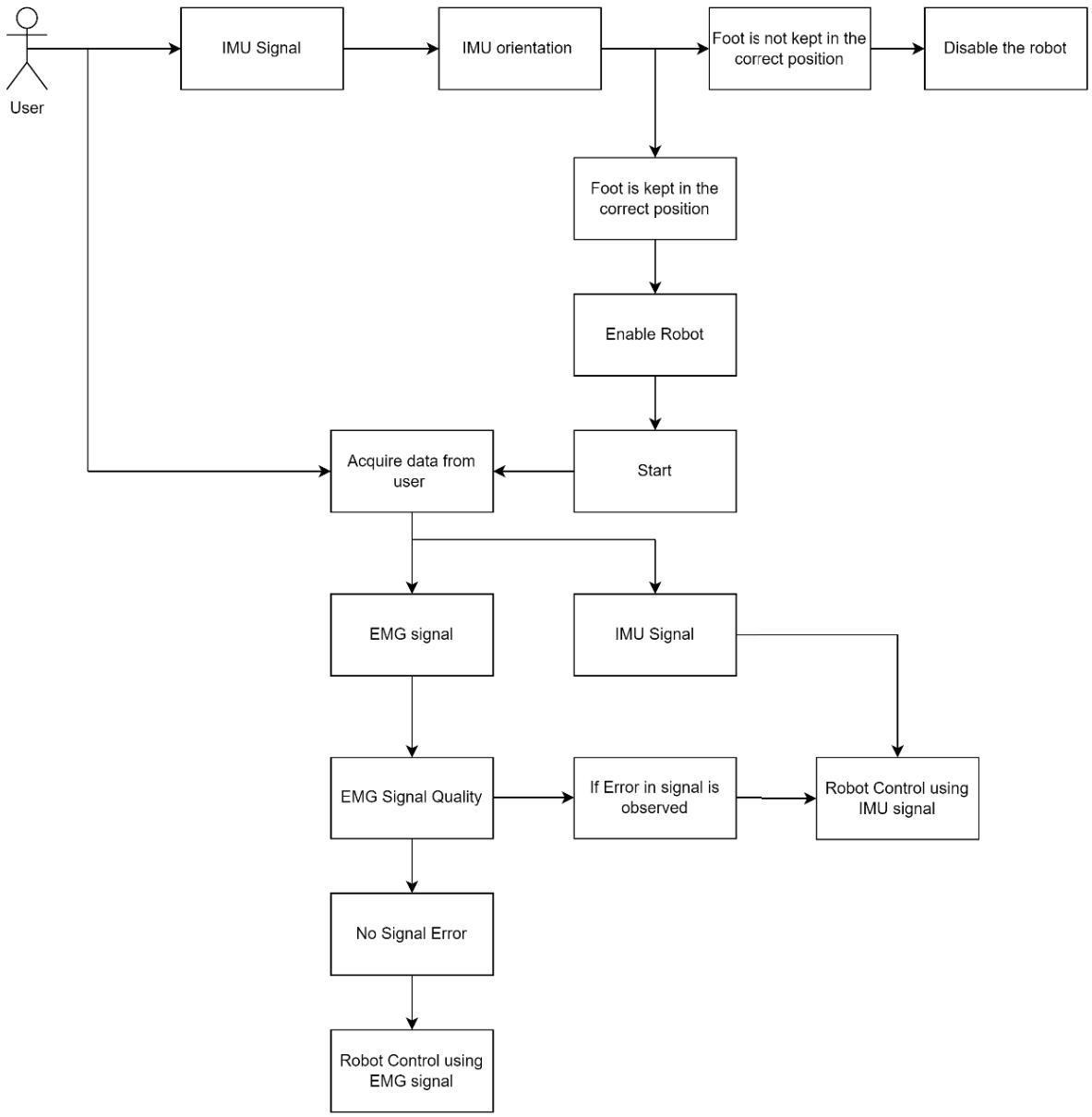


Figure 4.5. Algorithm to control ankle robot using IMU when an error is present in the EMG signal

4.2.4 Ankle Torque and Ankle angle estimation using EMG

Due to this project's hardware limitation, we estimate the ankle joint torque and angle only for ankle plantarflexion and dorsiflexion. Ankle plantarflexion and dorsiflexion are two movements that the developed model explores. Three muscles, the anterior tibialis, the lateral and medial gastrocnemius, provide the model's input of s-EMG signals.

In order to accommodate users of varying ages, sizes, and joint conditions, the model was easily modified by simply entering the relevant anatomical and physiological data. This is obtained via machine learning and optimisation techniques all of which do away with the hassles of conventional methods like calibration and complicated sensor placement. The classification of intent is the initial stage of the system. This stage generates empirical data that is used to select the model that is appropriate for the problem at hand. As a result, the system's execution and processing times are reduced, and the system's resilience and adaptability time can be increased. When classifying data, SVM is used to classify the movement intent. After being classified, muscle activation is determined from the obtained s-EMG data. This data is then used as input into a number of nonlinear mathematical models, with each model's output representing a unique torque value at a given instant in time. The system evaluates the user's intentions before selecting from one of the distinct sets of mathematical equations. In the process of optimising mathematical models, a technique known as particle swarm optimisation is utilised to iteratively improve solutions while working within the boundaries of a constrained search space.

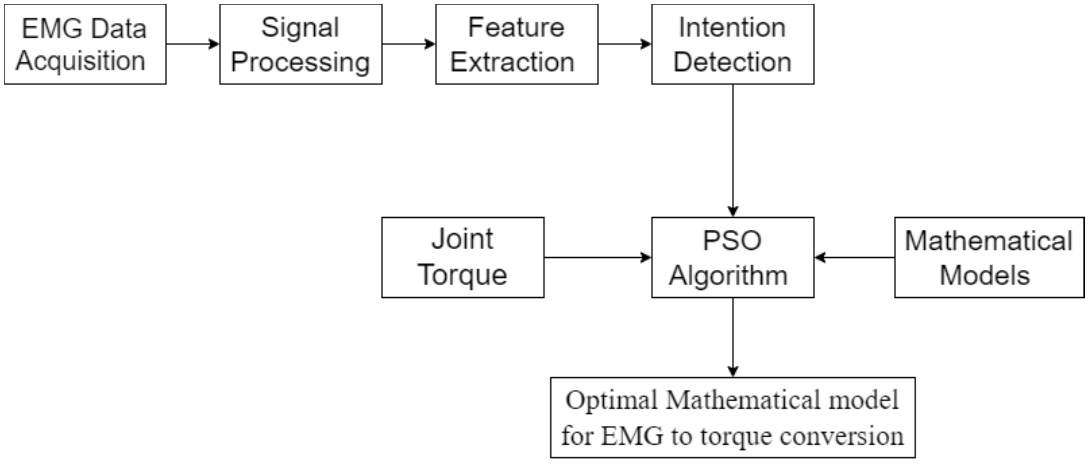


Figure 4.6. System Overview

4.2.4.1 Ankle Torque and Ankle angle estimation using EMG

The first step for determining the torque from the EMG signal is by obtaining the muscle's activation dynamics. This is achieved by utilizing a low pass filter of 2Hz to a rectified EMG signal to smooth output the envelope, yielding to the corresponding activation dynamics. The activation function may be thought of as a representation of the fundamental dynamics due to muscle excitation. Activation dynamics of the muscle is the process by which neural excitation generated in the brain is converted into the stimulation of the contractile apparatus, which is composed of a set of major proteins such as actin and myosin. This structure is specifically designed in the human body for the process of contraction and is made up of sarcomeres in the muscle [121]. Whether the relationship between EMG and its respective force is linear or nonlinear, it is described by the activation function; the activation equation provided in this study is based on the research by [9, 116]. The activation function can define any relationship between EMG and force.

$$a_j(t) = \frac{e^{Su_j(t)} - 1}{e^S - 1} \tag{4.8}$$

where,
$a_j(t)$ is the muscle activation during muscle contraction
$u_j(t)$ is the EMG signal
S is a nonlinear shape that has a value between $-3<S_j<0$

Rectified EMG signal cannot provide an accurate prediction on its own and has a shorter time span. However, studies have shown that a more accurate prediction of muscular forces may be achieved by adding muscle twitch in the activation function [122, 123]. Muscle twitches have been modelled using a linear, critical damping second-order discrete model. In addition to modelling muscular twitches, this model may be stated in a discrete form using backwards difference [124, 125]. This model was used to represent muscle excitement. In this case, the activation function was provided as a recursive filter, and its formula was as follows:

$$u_j(t) = \alpha e_j(t-d) - \delta_1 u_j(t-1) - \delta_2 u_j(t-2) \tag{4.9}$$

where,

$e_j(t)$ is the EMG signal before being processed
$u_j(t)$ is the EMG signal after being processed
α is the coefficient for muscle
δ coefficient of the recursive filter
d is 80ms is the delay between muscle excitation and the EMG signal

where,
$\delta\ 1 = e1 + e2$,
$\delta\ 2 = e1 \cdot e2$

Such that $|e1|<1$ and $|e2|<1$
$\alpha - \delta_1 - \delta_2 = 1$; since the recursive filter maintains a unity gain

Once the EMG's activation function has been established, the data is input into three distinct mathematical models that individually classify the action performed. To determine which mathematical model produces the most reliable estimate of torque, we use a technique called partial swarm optimisation (PSO).

4.2.4.2 Torque conversion from EMG signals

This chapter's mathematical analysis is based on the notion that the relationship between the torque produced by the joint and its respective EMG signal is nonlinear in nature [126]. Under this assumption, nonlinear mathematical models are provided for estimating ankle joint torque described in equations 4.10, 4.11 and 4.12. The models were developed using data from earlier research that was very similar [127].

$$MM_1 = x_1 + x_2 \cdot \sqrt{u_i} \tag{4.10}$$

$$MM_2 = x_1 + x_2 \cdot sine(u_i) \tag{4.11}$$

$$MM_3 = x_1 \cdot u_i^{x_2} + x_3 \cdot u_i^{x_4} \tag{4.12}$$

where,

u_i = EMG data.
x_i is initiated with a random variable

MM = Mathematical model

The project's nonlinear equations represent basic, complex, and equations with additional unknown variables. This would allow us to understand how various equations with identical properties can be used to find the optimal solution. Using these three distinct types of nonlinear equation would allow us to observe the behaviour of each equation in terms of its execution time, or the number of iterations required to obtain a solution for a given type of problem. Though different equations would yield different solutions, the execution time and number of iterations for similar types of nonlinear equations would be comparable.

4.2.4.3 Fitness Function

A fitness function with a sum of all squared errors will be used for an objective function, shown below (SSE). By using the function defined below, it is possible to get an overall sense of how accurately the mathematical equations can estimate torque from the EMG signal. It also helps the algorithm to find the best solution. The predicted torque is obtained from a selected mathematical model.

$$SSE = \sum_{k=0}^{n} (Actual\ Torque - Predicted\ Torque)^2 \qquad (4.13)$$

The continuous ankle torque was measured using a Biodex System 4 isokinetic dynamometer. Figure 4.7 shows the setup for ankle torque data collection. At full dorsiflexion of the ankle joint, the highest torque value is recorded by manually forcing the ankle joint upward. The same activity was also performed for ankle plantar flexion. This was done to see whether our mathematical model could maximise the s-EMG signal's torque output.

Figure 4.7. Ankle torque data collection using an isokinetic dynamometer

4.2.4.4 Particle Swarm Optimisation

PSO is a computational process in which particles in the search space are shifted according to a mathematical equation that describes their current position and velocity. This is an iterative process that leads to better solutions. This improves the candidate solutions and moves the particles by iteratively improving the candidate solutions. PSO has several advantages over other optimisation strategies, one of which is that it is more amenable to real-time implementation due to its requirement of primitive mathematical operators, which causes it to be computationally less expensive with respect to computational memory and storage [128]. Furthermore, PSO does not make assumptions about the problem it is trying to optimise [129].

Figure 4.8 depicts a flowchart of partial swarm optimisation. The swarm's particles use their exploration and exploitation algorithm to navigate the specified hyperspace, and they are equipped with two crucial forms of reasoning: the ability for each particle to remember its best position, which can be called as local best (lb), and the ability to learn about the best position in their neighbourhood known as global best (gb). The particle's velocity has an effect on where it is located in space. A particle "i" 's position in a defined space is represented by the notation $x_i(t)$ where "t" represents all the time steps during which the particle explores its

space. The location of particle shifts when its current position is multiplied by a velocity, represented by the notation vi(t).

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{4.13}$$

where,

$$v_i(t) = v_i(t-1) + c_1 r_1(lb(t) - x_i(t-1)) + c_2 r_2(gb(t) - x_i(t-1))$$

c_1 and c_2 = coefficient, which occurs due to acceleration.
r_1 and r_2 = random vectors

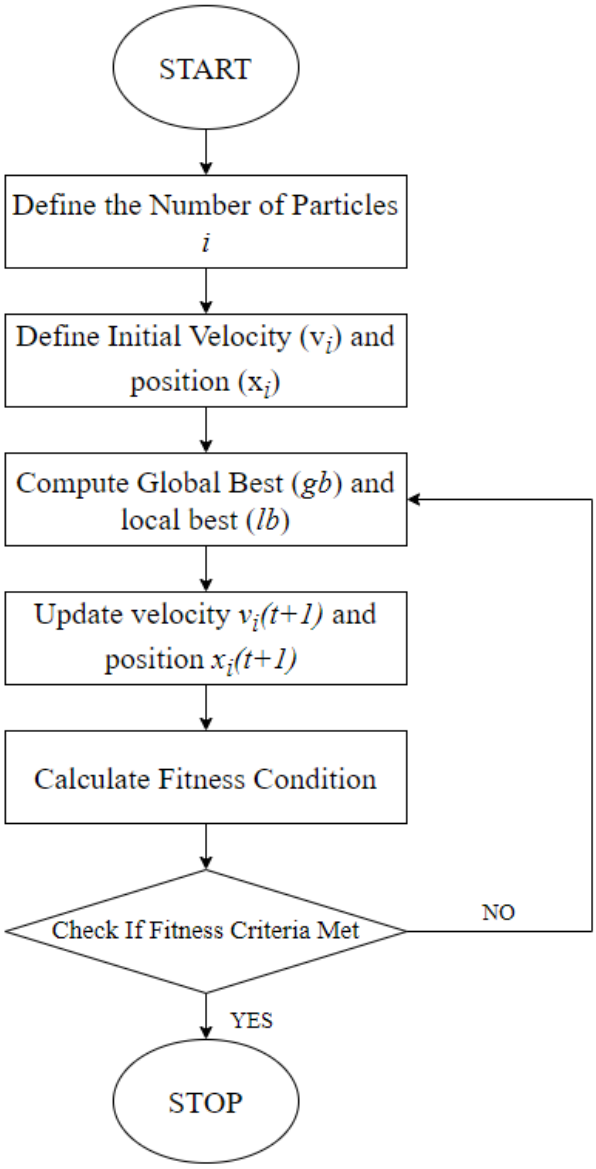


Figure 4.8. PSO flow chart

The aforementioned mathematical models are divided into dorsiflexion and plantarflexion, two of the ankle's motions. All mathematical models are switched on or off depending on the SVM classifier's output. Therefore, if dorsiflexion is identified as the movement, the suitable calculated mathematical model will be selected based on the model's performance, and if plantarflexion is the desired movement, then its respective mathematical model will be selected. Figure 4.9 depicts the methodology of the proposed method and illustrates what happens to the output torque when a patient is relaxed—this state results in zero torque being produced.

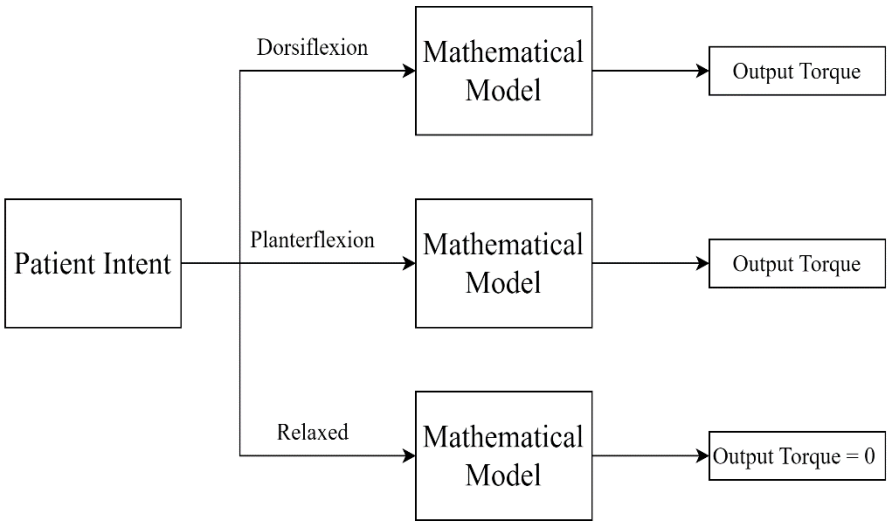


Figure 4.9. Torque estimation overview based on patient intent

4.2.4.5 Ankle Joint Position Estimation

Ankle joint estimation is required to provide smooth motion at the ankle. When the ankle is in a dorsiflexed position, there is a larger increase in muscular activity, which in turn causes increased joint torque and ankle movement. Plantar flexion reveals the same behaviour. The torque exerted by a joint and the joint ankle are highly proportional to one another when both are manipulated [130], and it is expressed as:

$$\theta_j = k\tau_i \tag{4.14}$$

where,

$\theta_j = Estimated\ Joint\ Ankle\ Angle$
$\tau_i = Estimated\ Torue\ of\ Joint\ Ankle$
$k = Adjustable\ constant$

4.3. RESULTS AND DISCUSSION

A. Intent Classification Using EMG and IMU

Six distinct classifiers were utilised to assess the implication of the selected features, and their response times were evaluated in real-time. Initially, an SVM with a third-order polynomial function was implemented. The classifier was trained with 50ms, 100ms, 150ms, 200ms, and 250ms window sizes. The remaining classifiers were subsequently trained to execute the same tasks. In previous research [22, 131, 132], RMS, MAV, SD, and Wavelength were frequently used for classification. In this project, we improve classification accuracy by combining RMS, MAV, SD, Skewness, and kurtosis. This combination of features is regarded as a revised method for using features to improve classification accuracy. Using kurtosis, we were able to observe the number of signal peaks generated by each muscle contraction. This distinction in muscle contraction provides additional insight into the nature of the contraction during each movement. The k-NN and LDA classifiers demonstrated the lowest accuracy with varying window sizes. Cubic and quadratic SVMs were the most accurate. When the window length was increased beyond 150ms, it was observed that the accuracy rates of all classifiers decreased slightly. In some cases, the accuracy percentage was saturated, with a negligible change in accuracy after a 200ms window size.

The Cubic SVM classifier achieved a 98.30 % accuracy rate. Although the quadratic SVM provided the same level of accuracy, it required more time to train. This is due to the fact that the SVM contains a polynomial kernel function with the fourth degree of order, which necessitates additional computation time. While testing both classifiers in real-time, the quadratic classifier took approximately 20 milliseconds longer to make a prediction. Though the time difference between the two classifiers in predicting the patient's intent was very small, a faster response allows the robot to move faster and assists the patient in maintaining a natural

movement pattern. A small increment of time in each component of the rehabilitation system would also cause a delay in the robot's response to the patient's intention. Figures 4.10 to 4.12 illustrate the classification accuracy of classifiers with varying window lengths.

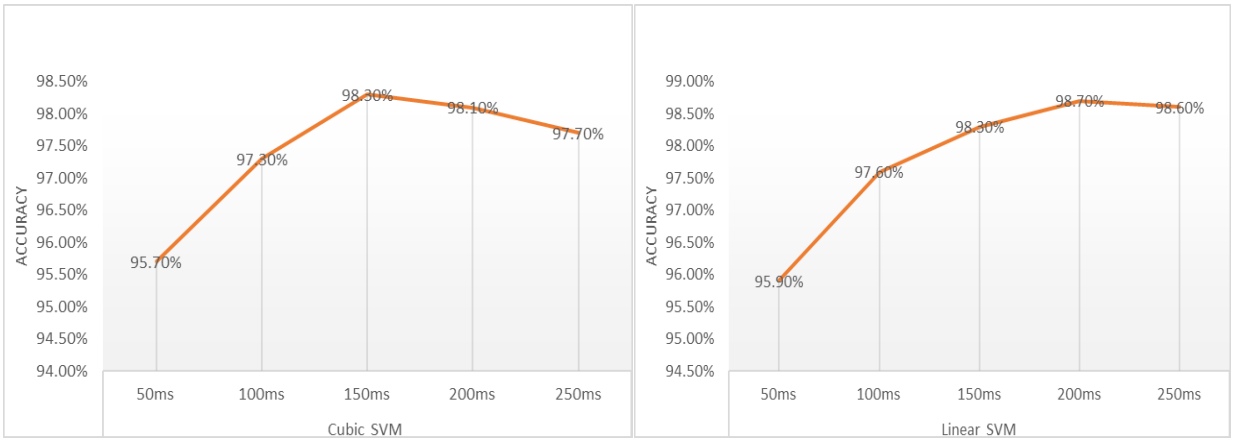


Figure 4.10. Accuracy of Cubic SVM (left) & Linear SVM (right)

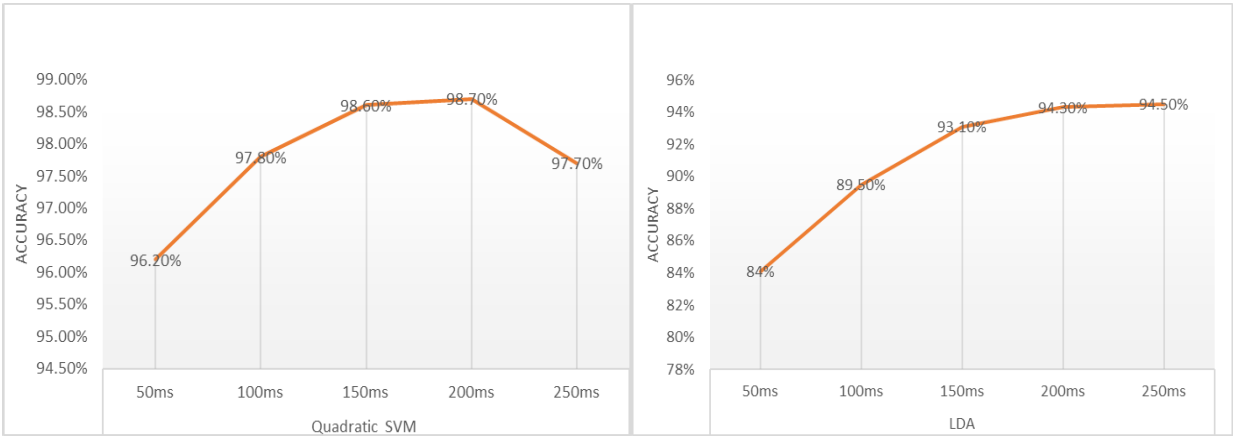


Figure 4.11. Accuracy of Quadratic SVM (left) & LDA (right)

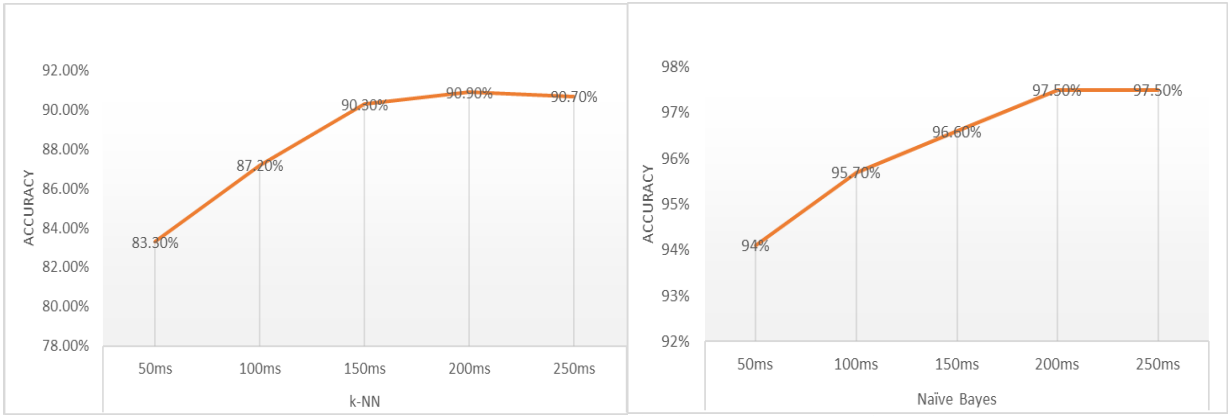


Figure 4.12. Accuracy of k-NN (left) & Naïve Bayes (right)

As previously discussed, the window size must be less than 300 milliseconds. The time between a patient's intention to move their ankle and their ankle movement should be less than 300 milliseconds. During real-time testing, it was determined that the robot's response time was close to 300 milliseconds, despite increasing accuracy after 150 milliseconds. It was also observed that the electromyogram signal of the ankle movements contained sufficient information between 100ms and 150ms. This window size would therefore allow us sufficient time to process the data, extract features from the signal, classify incoming data, and improve the robot's response time. Therefore, a window size between 100ms and 150ms was chosen. The optimal window length is determined by trial and error with 5ms increments between 100ms and 150ms. The data was categorised using the aforementioned classifiers. Using Cubic SVM and Quadratic SVM, the highest accuracy of 98.90% was observed when the window length was kept at 135ms with an overlapping window of 50ms. Most of the time, cubic SVM was used during real-time testing to reduce computation time. Implementing cubic SVM with the features mentioned above increases the robot's response time and accuracy, thereby rendering the system more stable and natural during rehabilitation exercises. The graph below depicts the accuracy of various classifiers with a window size of 135ms.

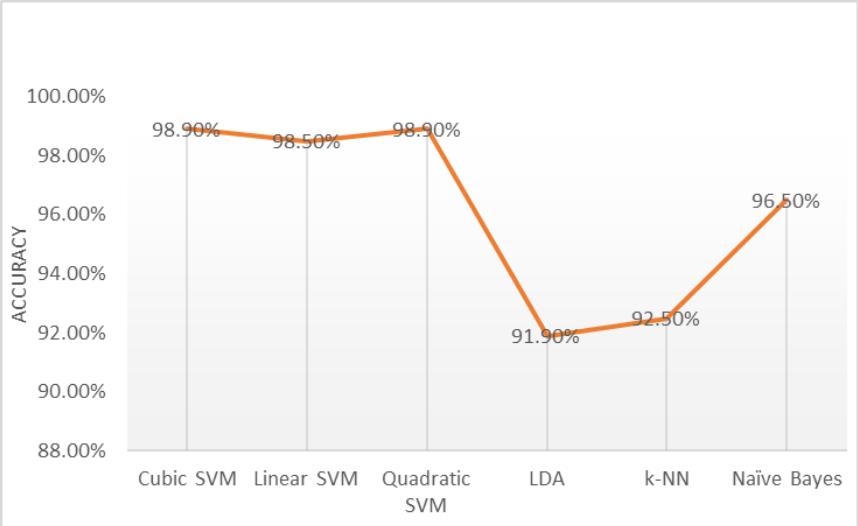


Figure 4.13. Accuracy of classifiers with a window length of 135ms.

High accuracy across all classes indicates that the classifier generated few false positives. The accuracy of dorsiflexion was the highest because it was properly separated from the other classes. The confusion matrix revealed that eversion and plantarflexion had the lowest classification accuracy of 98.5 and 98.1 percent, respectively. During the rehabilitation exercise, while performing ankle

eversion, it was observed that the subjects consistently pushed their ankles down, causing some data points to overlap in the signal. The confusion matrix was used to identify the overlapping classes, revealing that the majority of overlap occurs in ankle eversion and plantar flexion. At least one of the plantarflexion features overlapped with those of other classes, where class 1 was for the ankle in its resting state, class 2 for ankle dorsiflexion, class 3 for ankle plantarflexion, class 4 for inversion, and class 5 for ankle eversion. The error in plantarflexion could be due to the fact that bringing the ankle down requires less muscle contraction than dorsiflexion. Since dorsiflexion had the most significant muscle contractions, the features extracted from the tibilias anterior muscle (sensor 1) distinguished dorsiflexion from other movements with few misclassified classes. During offline classification, classification accuracy varied by approximately 0.75% across subjects. The obtained result demonstrates that cubic SVM along with the proposed features, can improve the consistent prediction of ankle movement activity and its patterns.

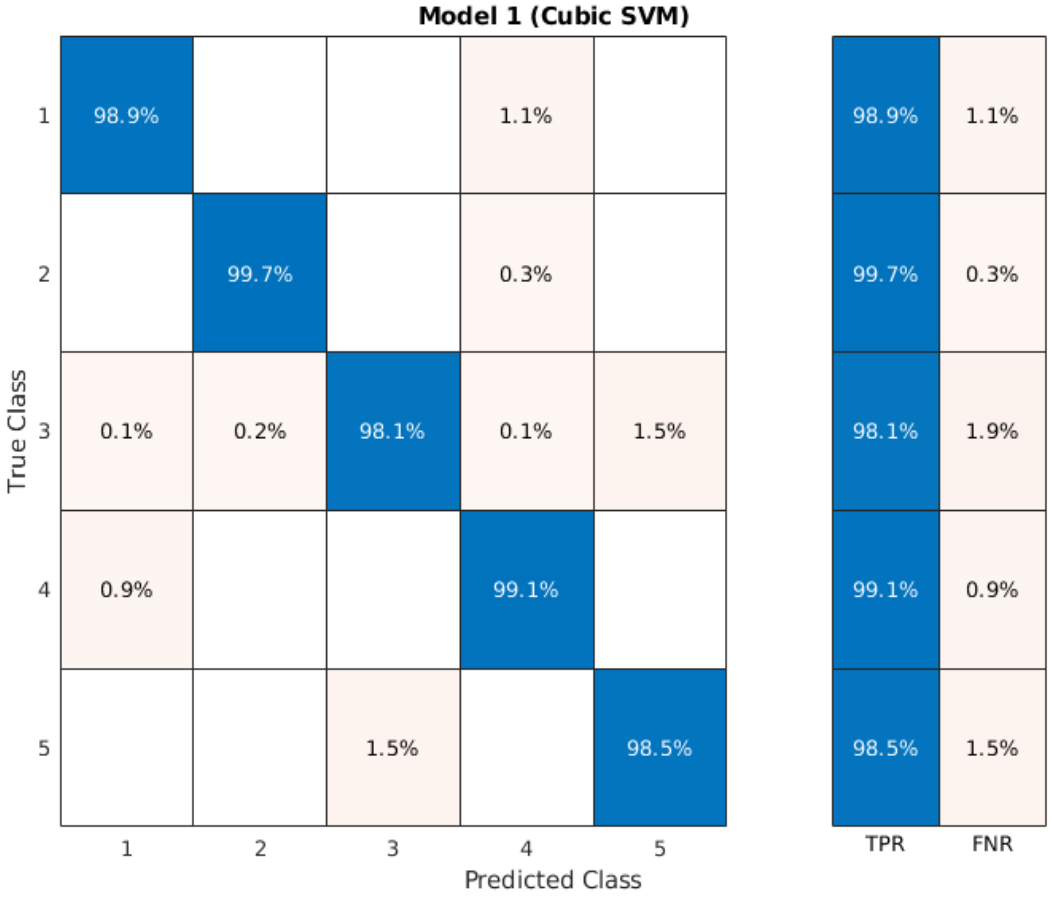


Figure 4.14. Confusion Matrix and True Positive and False Negative Rates for EMG classification

The IMU signal accelerometer data was used to classify ankle movements. However, these data are not primarily used for intent classification; rather, they are used to establish the actual position of the ankle in relation to the intended movement. The data was extracted for kurtosis and skewness features and classified using cubic SVM. During offline testing, 100% accuracy was obtained and did not generate any false positives. This is because, unlike EMG data, IMU data is distinct in each axis for each of the 5 different movements, and no two axes contain the same data for different movements. However, during real-time testing which was conducted for long a duration of 20 mins, the accuracy dropped slightly, as there had been a few misclassifications based on the ankle position. This is because the IMU sensor produces a signal drift accumulated over time. Using an IMU sensor with greater precision would resolve the signal drift-related issues. A random EMG sensor was detached from the user at a random time to test if the robot could be controlled using an IMU sensor. This resulted in an EMG sensor having a high noise ratio and within 3 seconds, the robot's control was automatically switched to use an IMU signal. If one of the four EMG sensors fails to provide a quality EMG signal, or if the EMG signal fails, the robot would be controlled by the IMU sensor.

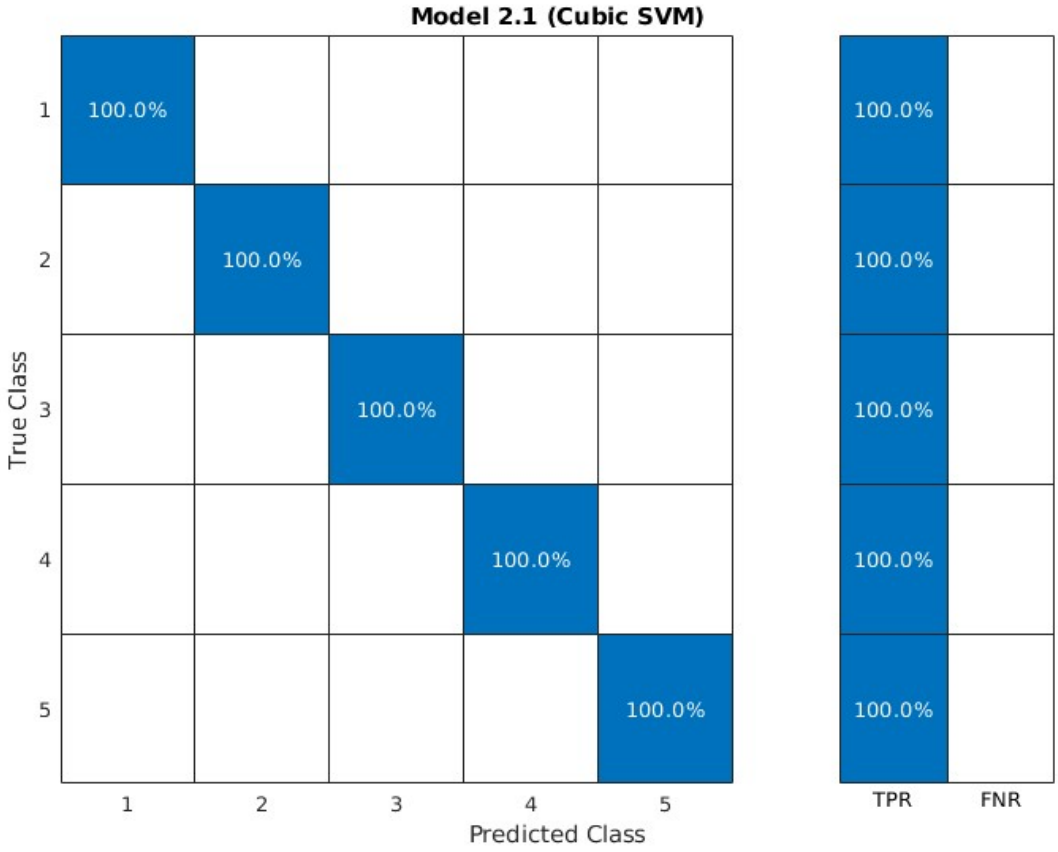


Figure 4.15. Confusion Matrix and True Positive and False Negative Rate for IMU classification

B. Ankle Joint torque and position estimation using EMG

In this project, ankle joint torque and ankle joint position was estimated only for ankle dorsiflexion and plantarflexion due to hardware limitation. The acquired EMG data was filtered and trained for three distinct ankle states using the cubic SVM classifier and with the selected features as previously discussed. Testing in real-time confirmed that the ankle robot had moved in the required direction without error. Using the EMG data, the next step of the experiment was determining the torque and ankle position. First, muscle activation data was extracted using the EMG activation function. Due to the delay in the EMG signal, the duration of the recorded force is shorter; hence, the activation function contributed to a more accurate evaluation of the muscle's forces. The muscle twitch was included in the final muscle activation function so that the estimation could be done more precisely. The acquired EMG signal, its activation function, and the change in the signal caused by the addition of muscle twitch are shown in the figure below for ankle dorsiflexion and ankle plantarflexion.

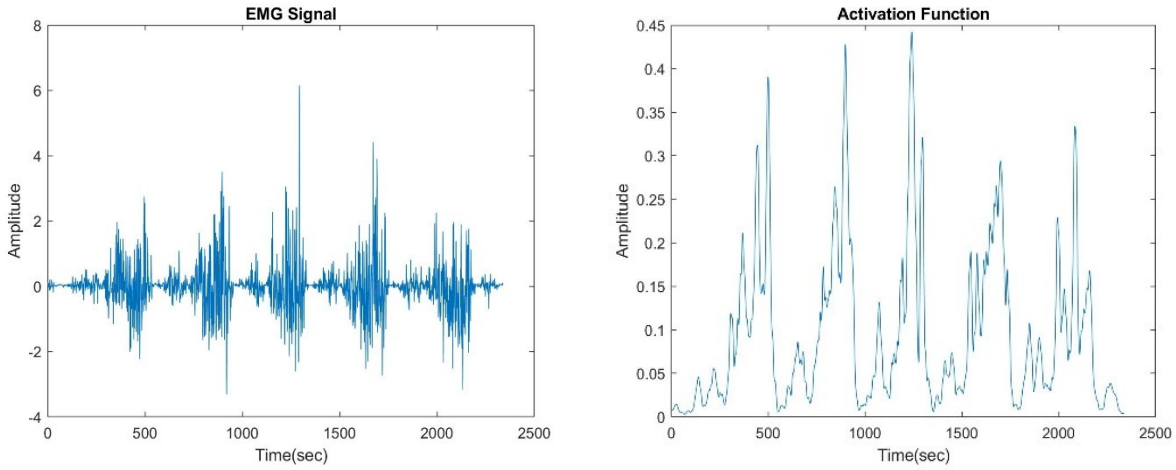


Figure 4.16. EMG signal obtained during ankle dorsiflexion and its resultant activation function

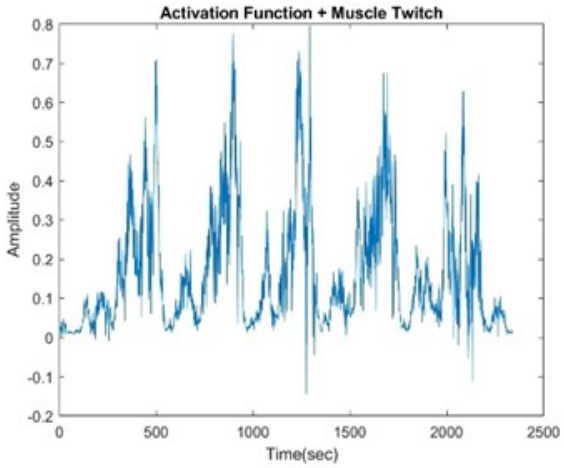


Figure 4.17. Muscle twitch added to the activation function of EMG signal during ankle dorsiflexion

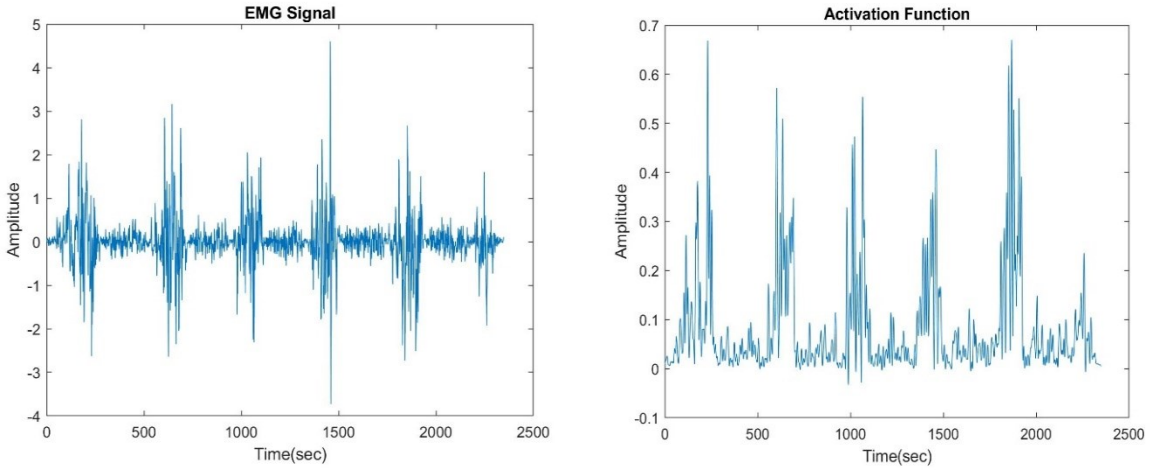


Figure 4.18. EMG signal obtained during ankle plantarflexion and its resultant activation function

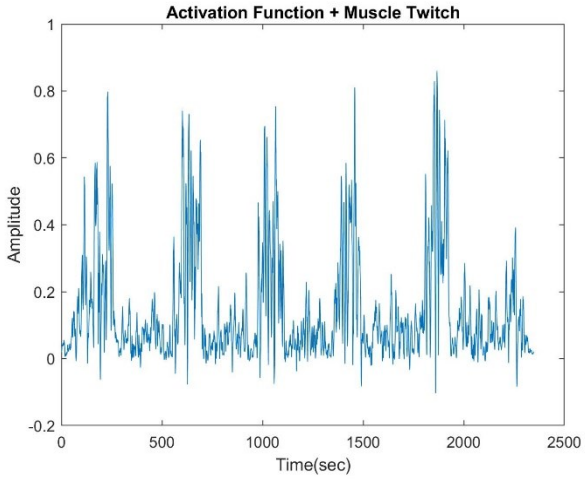


Figure 4.19. Muscle twitch added to the activation function of EMG signal during ankle plantarflexion

After that, the resultant activation function was put to use in the context of the second-order discrete linear model in order to obtain muscular excitation. This was done so that the model could predict the activity of the muscles. A recursive filter was used, and some of its characteristics included the electromechanical delay of 80 milliseconds as well as the prior EMG activation data. After the data was acquired, the muscle excitation data was entered into a sequence of three nonlinear mathematical models to build a relation between EMG and ankle joint torque. An iterative procedure using the particle swarm optimisation approach was utilised to solve the equations' unknown variables. The loop was terminated when the fitness function error between actual and anticipated torque was negligible. The predicted joint torque may be seen in the figures below, which were derived from three different mathematical models.

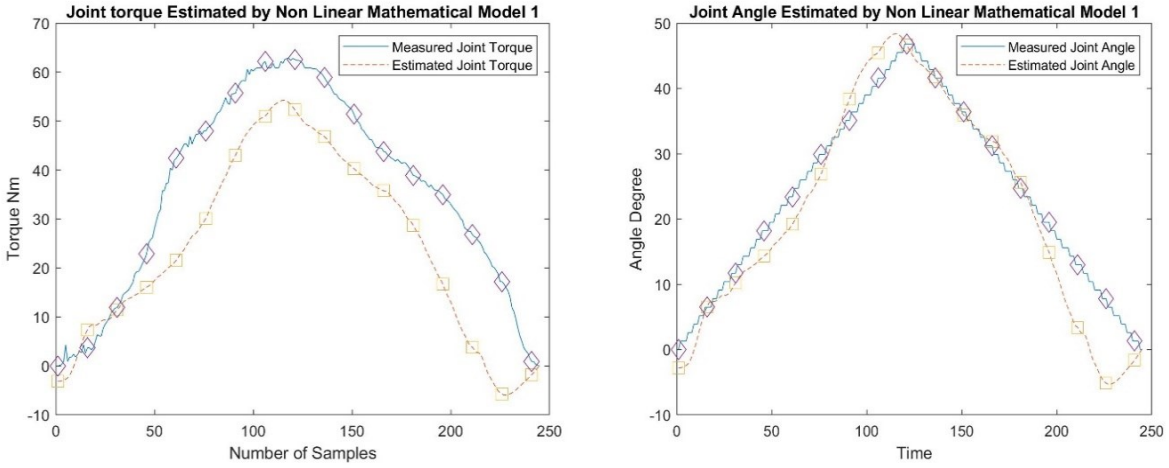


Figure 4.20. Estimated torque and angle using mathematical model 1 for ankle dorsiflexion

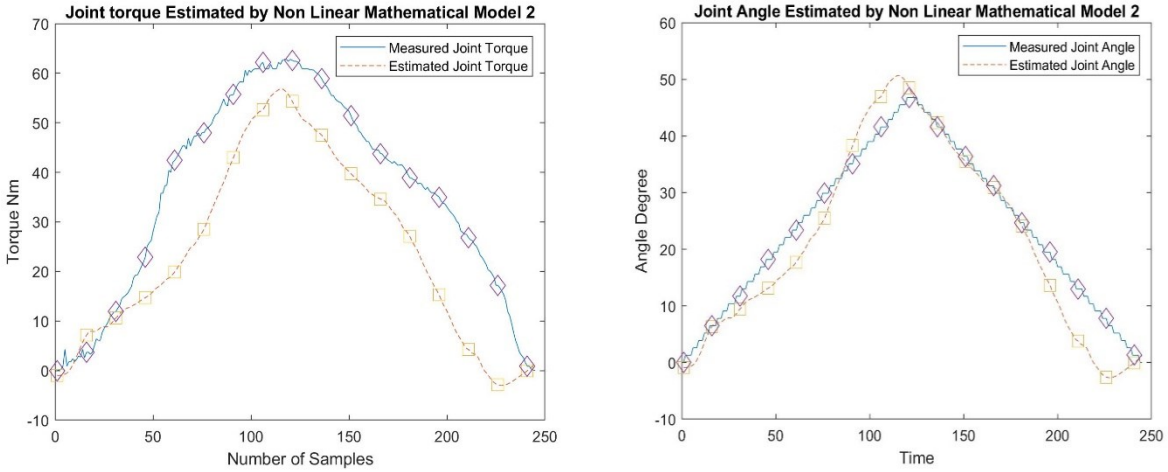


Figure 4.21. Estimated torque and angle using mathematical model 2 for ankle dorsiflexion

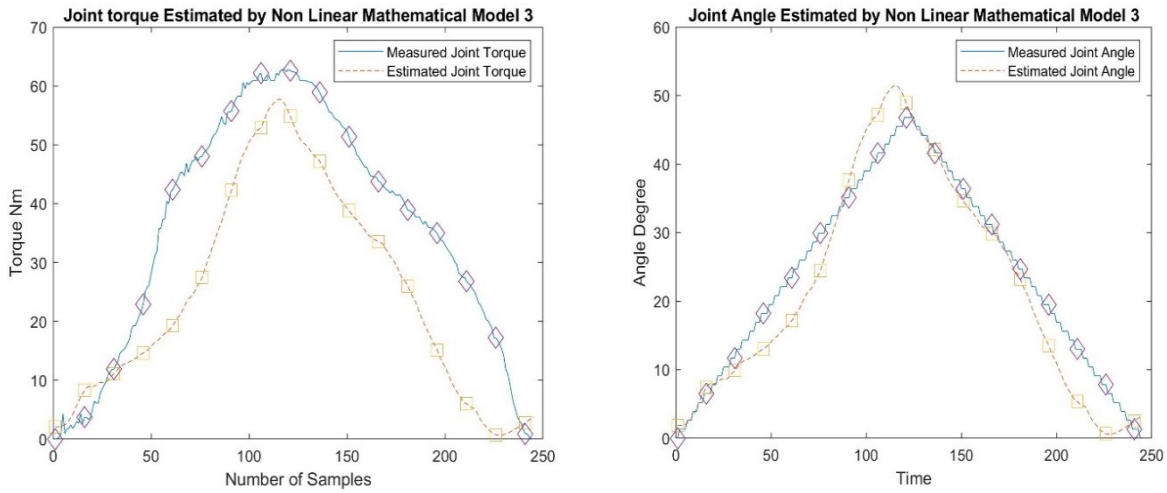


Figure 4.22. Estimated torque and angle using mathematical model 3 for ankle dorsiflexion

To get an estimate of the torque that is produced by a joint, three different mathematical models were applied. The figures show that the torque value and joint angle obtained from the mathematical model perform similarly to the data gathered using a dynamometer during both plantarflexion and dorsiflexion with high accuracy. The joint angle was found by establishing a linear connection between the torque and the angle by using the function, $\theta_j = k\tau_i$ to determine the linearity of the relationship. Compared to the projected torque, the estimated angle provides a more accurate representation of the measured angle. Inconsistent sample rate configuration between the dynamometer signal, collected at a sampling frequency of 128 Hz, and the EMG signal, taken at 2000 Hz, is also responsible for the slight shift that can be seen in the graph. The signals from the EMG were down sampled to minimise their impact on the estimated torque and angle.

Figures 4.20 to 4.22 depict the projected ankle dorsiflexion torque calculated using nonlinear mathematical models. When comparing the actual torque measured by the dynamometer, the estimated torque had an average error of 18%. A second observation was that the measured torque did not exhibit a smooth curve. Due to the dynamometer pedal's weight, the person cannot elevate it at a steady pace; consequently, slight jerks and dips have been seen in the recorded joint torque. To extract the maximum amount of torque from the subject, the subject was physically aided in pulling the pedal towards themself. This was done to ensure that the person participated to their maximum capacity. Even though the anticipated joint torque for dorsiflexion exhibited an overall smooth

curve given the indicated limitations, the error between the expected and observed joint torque was minor. The calculated ankle joint angle during dorsiflexion was found to be off by an average of 8%, which had a difference of ± 2 degrees which is negligible. Since a linear relationship was established between joint torque and joint angle, the anticipated joint angle followed the same trajectory as the joint torque with regard to their respective position values. When the ankle was returned to its original position, the most errors were detected. The variation in torque value and the joint angle between the three mathematical models was minimal.

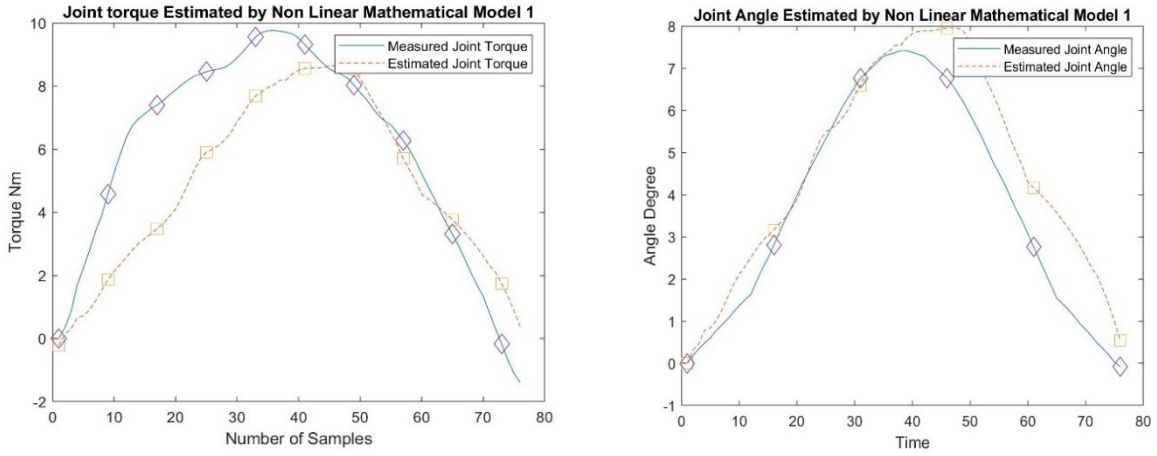


Figure 4.23. Estimated torque and angle using mathematical model 1 for ankle plantarflexion

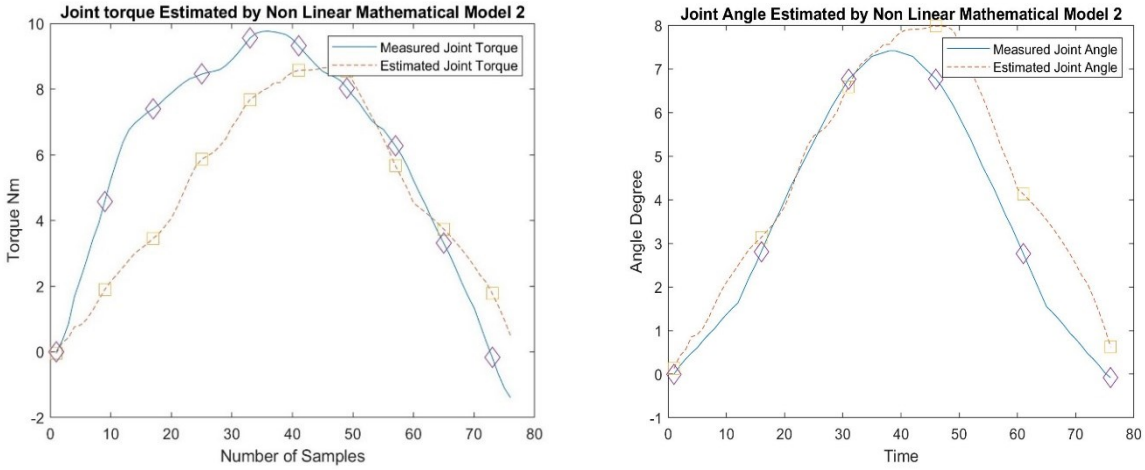


Figure 4.24. Estimated torque and angle using mathematical model 2 for ankle plantarflexion

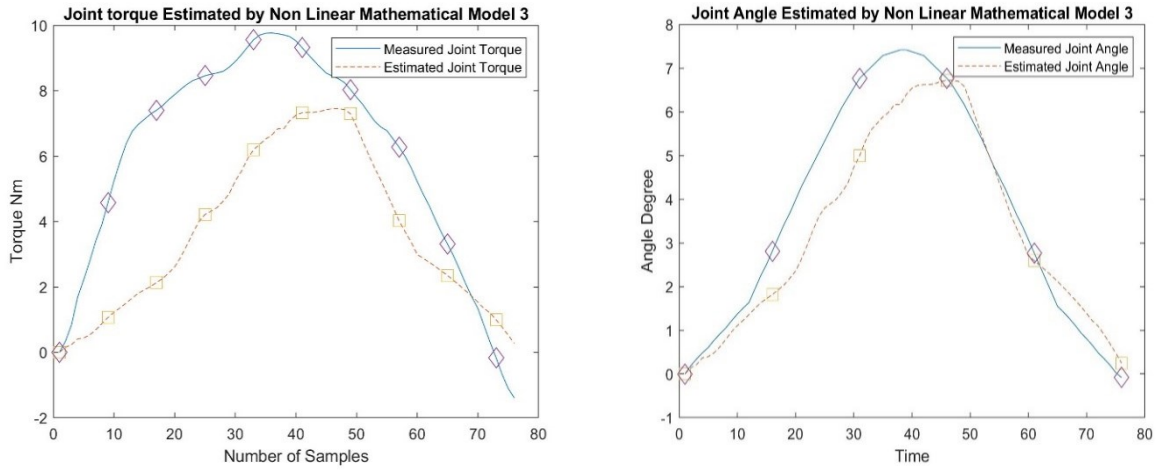


Figure 4.25. Estimated torque and angle using mathematical model 3 for ankle plantarflexion

Dorsiflexion	Variables	Values
Mathematical Model 1	x_1	-64.7
$MM_1 = x_1 + x_2 \cdot \sqrt{u_i}$	x_2	843
	Number of Iterations	205
	Execution time(sec)	0.5437
Mathematical Model 2	x_1	-15.9
$MM_2 = x_1 + x_2 \cdot sine(u_i)$	x_2	3520
	Number of Iterations	158
	Execution time(sec)	0.4741
Mathematical Model 3	x_1	994000000
$MM_3 = x_1 \cdot u_i^{x_2} + x_3 \cdot u_i^{x_4}$	x_2	1020000000
	x_3	13200
	x_4	1.41
	Number of Iterations	190
	Execution time(sec)	1.9821

Table 4.1. Values obtained for ankle dorsiflexion from mathematical models using PSO

Figures 4.23 to 4.25 display the derived nonlinear mathematical model-predicted ankle plantarflexion torque. The average deviation between the torque recorded by the dynamometer and the predicted torque was 27%. In contrast to the real torque seen during dorsiflexion, plantarflexion of the ankle exhibited a

smooth curve. This is because moving the ankle downwards towards gravity is simpler. In addition, it was shown that ankle plantarflexion resulted in a modest increase in velocity compared to ankle dorsiflexion. The participant was required to regulate the pace at which the ankle was moved, which likewise caused minor dips in the measured torque. When estimating the angle of the ankle joint during dorsiflexion, an average inaccuracy of 14% was detected. Similarly, the predicted joint angle followed the trajectory of the joint torque. The recorded torque was much smoother while the ankle was coming back to its resting position. Although the projected torque matched the trajectory of the measured torque, its curve was not as smooth as the actual torque. The outcomes of mathematical models 1 and 2 were relatively comparable; however, the third model had the most significant deviation from the measured torque joint.

The variables found for each of the mathematical models of ankle dorsiflexion that were created using PSO are listed in Table 4.1. The execution of mathematical model 2 required the least amount of time, taking just 0.471 seconds to complete the task. This made it the most efficient of the three models. Model 2 took the fewest number of iterations compared to the other two models. However, the number of iterations did not affect the amount of time required for the execution since model 1 had more iterations than model 3, even though model 1 took very little time. This is because model 1 required more iterations before it could satisfy the fitness criteria. More than twice the time was required to fulfil the fitness criteria since the number of variables that needed to be calculated for mathematical model 3 was increased. In comparison to the other mathematical models, it was observed that Model 3 had the highest values for the unknown variables on average. After analysing the torque data produced by each mathematical model, it was established that model 2 had the lowest estimate error for joint torque for dorsiflexion.

Plantarflexion	Variables	Values
Mathematical Model 1	x_1	-26.9
$MM_1 = x_1 + x_2 \cdot \sqrt{u_i}$	x_2	45.4
	Number of Iterations	387
	Execution time(sec)	0.3868
Mathematical Model 2	x_1	-12.3
$MM_2 = x_1 + x_2 \cdot sine(u_i)$	x_2	36.4
	Number of Iterations	273
	Execution time(sec)	0.2798
Mathematical Model 3	x_1	-223000000000
$MM_3 = x_1 \cdot u_i^{x_2} + x_3 \cdot u_i^{x_4}$	x_2	514000000000
	x_3	29.8
	x_4	2.57
	Number of Iterations	158
	Execution time(sec)	0.4741

Table 4.2. Values obtained for ankle plantarflexion from mathematical models using PSO

Table 4.2 outlines the variables discovered for each mathematical model of ankle plantarflexion generated using PSO. Model 2 used the least amount of execution time, requiring just 0.2798 seconds to complete the operation. Execution time, on average, was much quicker than calculating the variables for dorsiflexion for all three mathematical models. Model 2's fulfilment of the fitness function required more iterations than model 3. In addition, the number of iterations had no effect on the execution time since model 1 had more iterations than model 3 while requiring much less time. This is due to the fact that model 1 needed more iterations to meet the fitness requirements and had just two undetermined variables. The difference in timing between these two models was about 88 milliseconds. In contrast to dorsiflexion, the time needed to meet the fitness criterion for all three models was almost identical. Model 3 had the least number of iterations. Due to the increasing number of variables that must be computed for mathematical model 3, model 3 had the longest execution time of 0.471 seconds. Model 3 had high values for the unknown variables compared to the other mathematical models. After analysing the torque data generated by each mathematical model, it was determined that model 2 had the lowest estimate error for dorsiflexion joint torque, whereas model 3 had the highest estimate error. Comparing the two tables led to the conclusion that Model 2 predicted better outcomes for ankle dorsiflexion and ankle plantarflexion. The execution time

increases with an increase in variables $\boldsymbol{x}_n$, where n = 1,2,3 and 4. It can also be said that, with the increase in the number of unknown variables, the number of iterations can be reduced but can be computationally costly. Therefore, the number of iterations will also be independent of the execution time.

4.3. CONCLUSION

This chapter demonstrates the use of EMG signals to interact with an ankle robot for rehabilitation. The findings indicate that identifying the right features in conjunction with the appropriate features can enable accurate identification of the user's movement intent using EMG. This chapter examines the robot's response time in relation to the patient's intent. Reading the data every 135 milliseconds has resulted in more precise data classification and a more natural robot response. In addition to classifying EMG data, we also use IMU data to enhance the user's safety. When an EMG sensor generates excessive noise, the system switches by controlling the robot using IMU signals. In the second section, we discuss using an EMG signal to estimate ankle joint torque and angle using PSO. Once the intent is identified, the proposed novel method can enable the robot to move the ankle to the user's desired position. By doing so, the robot is able to interpret the user's intentions and direct the ankle movements to the correct location.

5 ANKLE MOVEMENT CLASSIFICATION USING EEG SIGNALS

5.1 INTRODUCTION

Electroencephalogram (EEG) signals are acquired from the brain using a non-invasive method by placing electrodes on the scalp of the subject under study. This is a type of physiological signal that can be utilised to predict ankle movement. The premotor cortex in the brain is the origin of the initial impulse in any voluntary ankle movement [133]. Using EEG signals, we are able to decode brain activity in real time. EEG signals are not affected by the contraction of muscles or the activity of nerves in areas of the body other than the brain. Due to this property of EEG, we can comprehend the patient's movement intent even when their muscles are at rest. Therefore, using EEG signals by stroke patients undergoing robotic rehabilitation is extremely beneficial. Patients who have experienced loss in the muscles or nerve activity that regulate ankle movements, such as those with strokes, damage to the spinal cord, or traumatic brain injuries, can use EEG signals to regain some degree of independence in their movements. Patients with this condition cannot exercise independent control over their bodies and, in extreme cases, are incapable of communicating with others. EEG signals can be used to examine a person's brain activity, which can also facilitate communication with the outside world and improve the life quality for patients. Motor imagery (MI) can stimulate the brain's sensorimotor cortex, which can be translated into EEG signals that reflect a person's intended movement.

In recent years, the practice of MI, in which a person imagines moving a body part without actually doing so, has gained significant popularity. EEG can

assist patients undergoing robotic therapy and aid them in completing rehabilitation activities [134]. When a person begins to visualise ankle movement, the brain's premotor cortex experiences a change in electrical potential. We could determine the patient's intended movement by analysing the changes in the signal's behavioural patterns. This would allow patients to interact with the robot for enhanced rehabilitation. MI engages the motor cortex virtually at the same level as actual movement and aids in the repair of the damaged portion [135]. Classifying these movement thoughts precisely is quite difficult. Classifying MI-based EEG data using classifiers like k-NN and LDA has been the subject of several research. Classifying MI-based EEG data using machine learning algorithms like k-NN and LDA has been the subject of several research. [42, 64, 134]. In terms of real-time application and computational efficiency, very few studies have been done on the effects of these approaches. Since the primary objective of this thesis is to improve human-robot interaction, we investigate methods for improving computational efficiency and, consequently, the robot's response time in relation to the patient's movement intent for effective rehabilitation.

One of the difficulties associated with collecting EEG signals is the number of electrodes that must be used on a patient to identify MI-based movement intent. Studies have utilised electrodes ranging from 16 to more than 120 [42, 64, 135]. This could lengthen the time required to apply the electrode and prolong the patient's discomfort. This chapter discusses the use of frequency domain and time domain features to enhance classification accuracy. To obtain the same level of accuracy as when using all EEG channels, we investigate effective channel reduction techniques based on independent component analysis (ICA). After identifying the dominant channels, we examine the effect of feature reduction on classification precision. Maximum Relevance Minimum Redundancy (MRMR) is employed to order the EEG channel's characteristics. The primary benefit of the MRMR method for feature reduction is its ability to train data quickly and produce extremely accurate results. We classify the data with Cubic SVM to evaluate the effect of ICA and MRMR on MI-based EEG data.

5.2 METHODOLOGY

5.2.1 Data Collection and Signal Processing

The EEG data was collected from three healthy subjects using an EMOTIV Epoch x EEG headset. Signals were gathered from 14 distinct EEG channels AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, and AF4. The electrode positions on the EEG headset are based on the international 10-20 system. First, the signals were gathered in order to classify between three and five distinct ankle states. As this is the primary joint movement of the ankle, the three states are ankle plantarflexion, ankle dorsiflexion, and the rest state. In addition to the three states, data on ankle inversion and eversion were also collected for classification purposes. Before collecting data, three subjects received brief training on the motor imagery tasks they would be required to complete. The procedure for collecting data consisted of a 10-second cue indicating the task to be performed, followed by a 10-second visual image of the motor imagery task. During the 10-second cue preceding the task, the participants were instructed to prepare for the task by viewing a picture of an ankle position. The 10-second-long motor imagery task was preceded and followed by a beep sound. Before the next task, there was a one-minute break. No EEG data were collected during the 10-second preparation period and the 1-minute break. Each session comprised the collection of data for all five moments, which were recorded in distinct blocks for each task. Before repeating the session, each subject took a twenty-minute break.

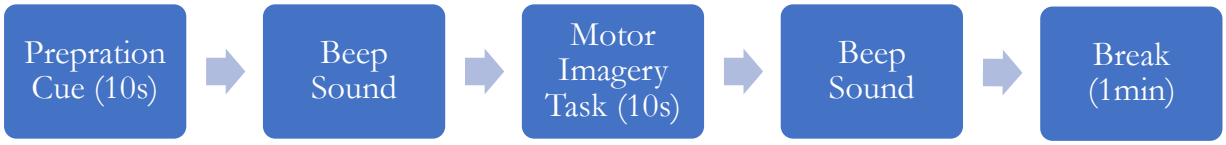


Figure 5.1. Protocol for EEG data collection

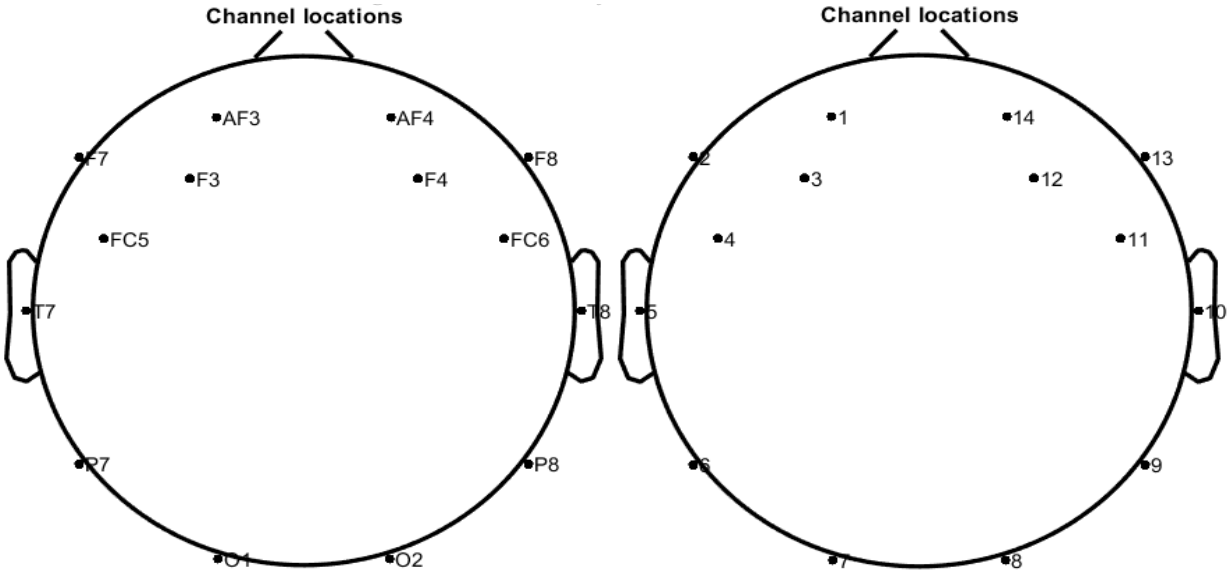


Figure 5.2. Emotiv EEG electrode position and channel number

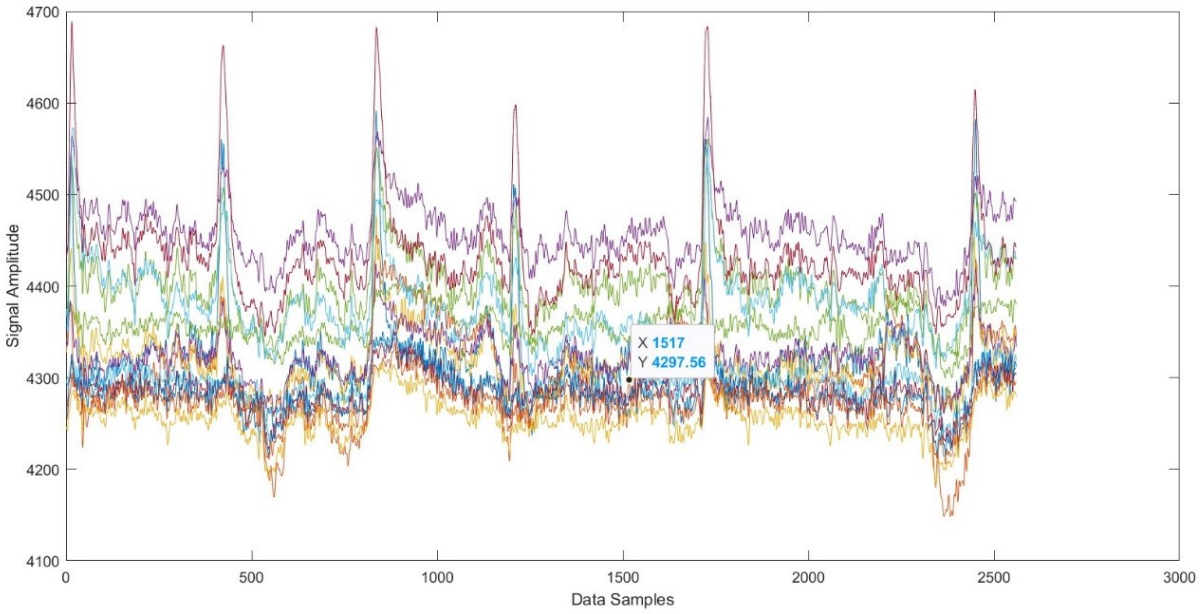


Figure 5.3. Acquired EEG data

The acquired data was filtered using a bandpass IIR butterworth filter of the 3rd order with frequencies ranging from 0.5 Hz to 30 Hz. The DC offset in the signal was eliminated by rejecting the signal's mean. There were numerous artefacts, such as eyeblinks and movement artefacts, in the EEG recording. The most evident was the eye blink artefacts present throughout the EEG signal. The

filtered EEG signal underwent independent component analysis (ICA) to remove the artifacts.

5.2.2 Channel Reduction Using ICA

ICA has been an effective tool for processing EEG signals in recent years [136, 137]. To create the most significant distinction in the EEG signal, ICA linearly unmixes the multichannel EEG data to identify the data's independent components. ICA employs a similar principle of blind source separation (BSS), in which the different components of a mixed signal can be identified without extensive knowledge of the signal's characteristics. Because EEG signals are complex as it contains a lot of movement artefacts, ICA can be used to decompose EEG signals. ICA can identify the specific activity patterns in EEG data that correspond to eye blink and muscle activity and separate them from other mixed signals [138]. In straightforward terms, an ICA projects the recorded signals from each electrode onto a coordinate frame in which the components are independent of one another. By analysing the statistical characteristics of the data in the new coordinate frame, we can eliminate the signal's artefacts more efficiently. This study utilised EEGlab for ICA analysis of EEG signals [139]. The filtered signal can be recovered by using an unmixing matrix $m \times n$ to the EEG data. The equation for the unmixing matrix is given as follows:

$$r = Wx \tag{5.1}$$

where,
$x = W^{-1}r$

x is the EEG data
r is the components of the signal
W is the unmixing matrix $m \times n$ as the special filter for independent components
W^{-1} is the $n \times n$ mixing matrix that has the weights of each EEG component

The spatial filter is the matrix, W^{-1} which contains the weights of each channel. The derived weights can map each source component's projection to its corresponding EEG channel. Therefore, this information can be used to obtain topographical information about the independent components and can be plotted on a map. Here, r represents the activity that varies with time, while W^{-1} represents the weights for each EEG channel. The function would aid in mapping the component weights of the EEG signal, allowing us to visualise the projection of

each channel. To eliminate the components that have been identified as artefacts without affecting the signal's other essential components, the column, W^{-1} of that component can be replaced with zeros.

In EEGlab, an automated infomax algorithm function known as runica is used to decompose the EEG signal. The Infomax ICA algorithm has been widely used to analyse and process EEG data. Although several other ICA algorithms, such as FastICA, JACE, and SOBI are significantly faster than the infomax algorithm, these algorithms tend to have stability issues during real-time implementation [140]. The infomax algorithm operates by minimising the mutual information contained in the data [141]. The algorithm attempts to find a stochastic gradient ascent and the matrix W that maximises the entropy of vectors with zero mean, where W separates signal components. While the ICA can decompose the signal into its component parts, W^{-1} can provide the relative strengths of each component [142]. This chapter uses a modified Infomax ICA with Hessian optimisation to derive the EEG channels' weight component matrix. The Hessian matrix consists of partial derivatives of the second order and is obtained by partially differentiating the gradient with respect to its corresponding vector [143]. Therefore, the Hessian-optimization ICA can optimise a given function and provide essential information regarding the functions being optimised. Although the algorithm is computationally expensive, it is faster and more accurate in deriving the signal components and has a higher spatial resolution for each channel projection [144].

$$\underset{\text{EEG Channels}}{x} \begin{bmatrix} x_{11} & x_{12} & x_{...} & x_{1m} \\ x_{21} & x_{22} & x_{...} & x_{2m} \\ x_{...} & x_{...} & x_{...} & x_{...} \\ x_{n1} & x_{n2} & x_{...} & x_{nm} \end{bmatrix} = \underset{\text{EEG Channels}}{W^{-1}} \begin{bmatrix} w_{11} & w_{12} & w_{...} & w_{1n} \\ w_{21} & w_{22} & w_{...} & w_{2n} \\ w_{...} & w_{...} & w_{...} & w_{...} \\ w_{n1} & w_{n2} & w_{...} & w_{nn} \end{bmatrix} \cdot \underset{\text{Components}}{r} \begin{bmatrix} r_{11} & r_{12} & r_{...} & r_{1m} \\ r_{21} & r_{22} & r_{...} & r_{2m} \\ r_{...} & r_{...} & r_{...} & r_{...} \\ r_{n1} & r_{n2} & r_{...} & r_{nm} \end{bmatrix}$$

Time Components Time

Figure 5.4. ICA decomposition equation

Since the weights, W^{-1} provide all the source components of a specific EEG channel; it also allows us to identify the dominant EEG signal components. The value of these weights indicates the EEG signal's most prominent characteristics. With this information, we can also eliminate unnecessary electrodes for classification. The inverse matrix consists of several components with varying

values that are present in the corresponding EEG channel. In a prior study by Rifai et al. [145], the minimum-maximum value of weights was used to identify the most important characteristics. If the weight value of an EEG channel exceeded a certain threshold, the channel would be selected. This method only considers the single weight value for each channel, which is one of its disadvantages. Since a single EEG is the sum of all of its components, in this chapter, we calculate the absolute mean of all the component weights for the EEG channel in question. We can observe the channel's average weight value for various tasks. Channels are selected with an absolute mean value higher than a selected mean threshold value. The selected channel is then classified for analysis using different mean threshold values.

5.2.3 EEG feature Ranking using the MRMR method

The subsequent phase of this project involves applying feature reduction techniques to minimize the dimension of features used for classification, thereby enhancing computational efficiency and helping to stabilise the classifier. Maximum Relevance Minimum Redundancy (MRMR), a type of filter method algorithm is used in this project to rank the features of the selected EEG channels. The algorithm identifies features that, when combined, must have the highest possible predictive power. This concept was initially developed to identify the optimal genes for specific diseases [46, 47]. If multiple features with essentially identical information are present in the data, MRMR will only select one of those features and discard the others. Therefore, enabling the removal of unnecessary features. The MRMR framework facilitates the selection of relevant features while minimising redundancy among the selected features. In the study by Direito et al. [146], this method was applied to reduce the EEG channel's features for predicting epileptic seizures. Experiment results demonstrated that MRMR could be used to determine which features are more significant than others. The SVM classifier achieved high accuracy due to the algorithm's assistance in reducing 147 features to the top 20 ranked features. Despite the fact that the MRMR algorithm is computationally efficient, it may have a negligible reduction in classification accuracy. Another application of MRMR is the selection of EEG features for stress classification, where the classifier output achieved a maximum accuracy of 93.75% after feature reduction [147]. In an experiment conducted by Xu et al. [148] EEG feature selection using the MRMR method was used to classify emotions which helped in minimising the number of EEG channels from 22 to 10 with a negligible

loss in accuracy. The literature demonstrates the effectiveness, dependability, and consistency of the MRMR method's outcomes.

At every iteration of the algorithm, features with the highest relevance to its class and the lowest redundancy between previously selected features are selected. Therefore, once a feature has been selected, it remains in the pool of selected features. At every iteration, the feature with the highest rank is selected. The following equation expresses the maximisation of total relevance based on MRMR:

$$M_r = \frac{1}{|r|}\sum_{x_r \in r}(x_r, x_i) \tag{5.2}$$

where,
r is the selected features
$|r|$ is the size of the selected feature set
x_r is one feature in the feature set r
x_i represents all available features
For n number of features, the equation to minimise the redundancy of the features based on MRMR is expressed as:

$$W_r = \frac{1}{|r|^2}\sum_{x_{r,y} \in r}(x_r, y) \tag{5.3}$$

where,
y is the target class label

The mutual information between M_r and W_r is defined by $mi(.,.)$ where the algorithm can determine how a variable's uncertainty can be minimised by analysing the information available in the other variables of the data [46]. The information between independent variables is defined as:

$$mi(y, x) = \sum_{i,j} p(x, y)\log\left(\frac{p(x, y)}{p(x), p(y)}\right) \tag{5.4}$$

where,
$p(x, y)$ is the probability density
$p()$ is the density function

It should also be taken into consideration that if the variable are independent of each other, then $mi(y, x)$ would be zero; however, if y and x are

equal, then “mi" would be the entropy of x [46]. The goal of MRMR is to find all relevant features such that it maximises the value of M_r. Though the algorithm was developed for pattern classification, several other MRMR algorithms were developed to improve the measure of relevance and redundancy of the features. The different variants of MRMR include, firstly, Mutual Information Difference (MID), which uses the difference to balance the redundancy and relevance and secondly, Mutual Information quotient (MIQ), which is the quotient method to achieve the same goal. One of the challenges with MRMR is that there may be a scale difference between relevance and redundancy [149]. The study by Zhao, Anand & Wang [149] shows that MIQ is resistant to the occurring scale difference. Therefore, in this study, MIQ, variant of MRMR would be used to rank the features of the EEG channels. The algorithm aims to find the maximum $MRMR_{MIQ}$ value and the equation is expressed below:

$$MRMR_{MIQ} = \frac{I(y,x)}{\frac{1}{|r|}\sum_{x_r \in r}(x_{r,}x_i)} \tag{5.5}$$

The algorithm starts by finding the most relevant features, and the selected ones are stored in r. The selected features are ranked via a heuristic algorithm using the MATLAB scores function, where a large score indicates the importance of a particular EEG channel feature with respect to its target class. In this project, scores of different rankings below a specific value would be neglected, and the results would be compared to the classification accuracy of all features.

5.2.4 Data Segmentation

A pattern in the EEG data can be observed for a given activity or mental task. Therefore, the EEG data must be divided into smaller segments. EEG signal classification requires a larger window size compared to s-EMG signal classification. During EEG-based real-time rehabilitation, the effectiveness of feedback is crucial. Therefore, if the feedback could be enhanced, the rehabilitation task would be performed more effectively. Since EEG signals are highly complex due to the signal's variability, selecting the proper window size is essential for increasing classification accuracy. Due to the uniqueness of the EEG signal's patterns, using smaller windows would provide little information, while using larger windows would delay the patient's feedback. In addition, a large window size may include data from other activities, which may be problematic if

the activity changes frequently. A larger window size would also reduce the available data sample size training. Even though smaller window sizes would contain less information, overlapping windows have been demonstrated to provide high classification accuracy for short window sizes when analysing EEG data [150].

Several studies have been performed to determine the optimal window size for EEG signals [150-152]. Studies have demonstrated that the accuracy of classifiers varies with window size. With multiple classifiers, window sizes less than 1 second and greater than 12 seconds have demonstrated lower accuracy. It has been observed that the highest degree of precision is attained for window sizes between 3 and 10 seconds. When compared to other classifiers, SVM has the highest accuracy with a 3-second window size. Candra et al. [151] and Tzimourta et al. [150] employed an overlapping window to improve accuracy and increase the number of data segments for training. Increasing the classifier's accuracy through smaller window sizes would allow the ankle robot to respond more rapidly to the patient's intent. Based on the literature, a 3-second window would be used to segment the EEG data in this study. A 1.5-second overlapping window comprising 50 percent of the actual window will be used for segmentation. Using an overlapping window resulted in a doubling of available training data and an improvement in accuracy.

5.2.5 Feature Extraction

EEG signal features are commonly extracted from its frequency domain, which is transformed from its respective time domain. The simplest time-domain characteristics, including RMS, Mean, Standard Deviation, Skewness, and kurtosis, have a classification accuracy of 87% for different sleep stages [153]. In classifying EEG signals, frequency features such as wavelets and fast Fourier transform are extremely popular [154]. In this session, along with time domain features shown in equations 4.1 to 4.5 in chapter 4 of this thesis, three frequency domain features are used to classify the EEG signal. A different combination of frequency domain features will be analysed to improve the classifier's accuracy. The equation and frequency characteristics are shown below:

- EEG waves consist of multiple frequency bands, the strength of which depends on the state of the brain. Delta (0.4 – 4 Hz) is the slowest band

with the highest amplitude, theta (5 – 8 Hz) represents brain activity in a relaxed state, alpha (8 – 12 Hz) is related to awareness, and beta (12 – 30 Hz) represents faster brain waves typically associated with intellectual activity and focus. The alpha and beta frequencies are dominant during cognitive activity [155]. The band power ratio between alpha and theta is taken as one of the features in this project, described below.

$$BPR = \frac{band\ power(\alpha)}{band\ power\ (\beta)} \tag{5.6}$$

where,
BPR is the band power ration
α is the alpha band
β is the beta band

- The next frequency feature is Log energy entropy (LEe). This feature can be used to determine the degree of complexity of the signal. It uses the probability of the summation of logarithms. The equation is described below:

$$LEe(x)_i = \sum_{i=0}^{N-1} (\log(p_i(x)^2)) \tag{5.7}$$

where,
$(x)_i$ is the signal
N is the length of the signal
p_i is the probability distribution function

- Mean Energy (ME) is the frequency feature where the energy values of a signal increase with the increase in activity [156]. It is a feature that could help determine different EEG activity stages. It calculates the average value of the squares present in all the samples of the signals. It is defined as:

$$ME = \frac{1}{N} \sum_{m=k-N+1}^{k} x(m)^2 \tag{5.8}$$

where,
$x(m)$ is the signal

N is the window length
k is the last sample of the signal

5.2.1 Classification

Various techniques and algorithms have been utilised over the years such that EEG signals may be classified more accurately. The performance of the robot is highly dependent on the selected features and classification accuracy. Using EEG for control applications requires large feature vector dimensions. Studies have demonstrated that SVM classifiers accurately classify various mental tasks [58, 59]. SVM can map data from one feature space to another, aiding in the classification of large feature vectors. Even though artificial neural networks have shown to improve the accuracy of detecting mental tasks, training requires enormous amounts of data. Furthermore, powerful systems are necessary to execute the algorithms. This session uses a multiclass cubic SVM to classify the three and five distinct ankle movement intentions, as previously discussed. This is done because the primary movement of the ankle joint involves plantarflexion and dorsiflexion. 14 EEG sensors are used to collect data in total. Classification of EEG data makes use of time and frequency domain features. Consequently, a maximum of eight features are obtained from each channel. The accuracy of each frequency feature will be evaluated separately, as well as the accuracy of the features when three frequency features are combined with the all-time domain features. Signal processing efficiency can be improved by minimising the number of features being processed. The results demonstrate that only using two frequency features along with all the time domain features can improve classification accuracy, whereas the classifier's accuracy decreases when all features are combined.

The next step in EEG classification would be using the weights generated by ICA to minimize the number of EEG channels. The frequency features that will be used for channel reduction will be decided based on the accuracy of the classifier after using all 14 EEG channels. Since all 14 EEG channels will be used initially for classification along with all eight features, a total of 112 features will be used to classify the EEG signal. A One Vs One approach will be taken to classify the EEG data. The obtained results illustrated reducing the irrelevant channels will not affect the accuracy heavily.

5.3 RESULTS AND DISCUSSION

The untrained EEG signal was initially filtered using a bandpass IIR filter of the 3rd order for frequencies between 0.5 Hz to 30 Hz. Even though the signal was filtered, artefacts such as eye blinks were still present in the signal. All 14 EEG channels were treated with the Newton-optimized Infomax ICA algorithm to eliminate artefacts. EEGlab was utilised to visualise the scale map of the signal's independent components. Visual examination of the components and signal revealed that five components contained the eyeblink artefact. In general, the artefacts were observed in the frontal lobe EEG channels. To reduce the prevalence of eyeblink artefacts in the EEG data, the identified components which had the artefacts were removed.

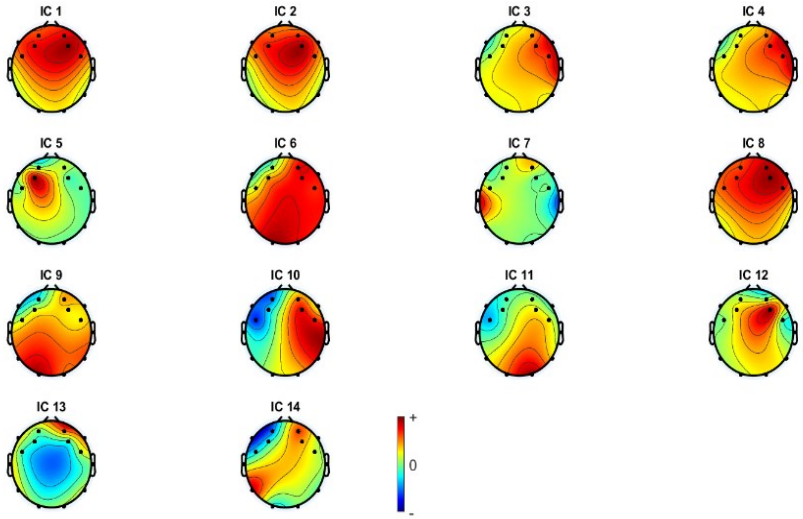


Figure 5.5. Scale map of Projected EEG component

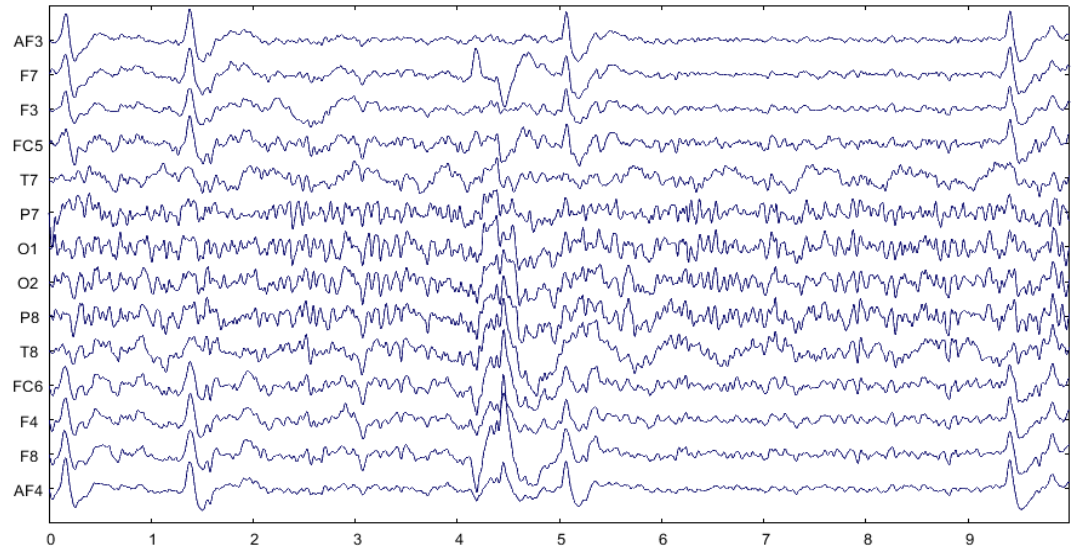


Figure 5.6. EEG signal with eye artifacts

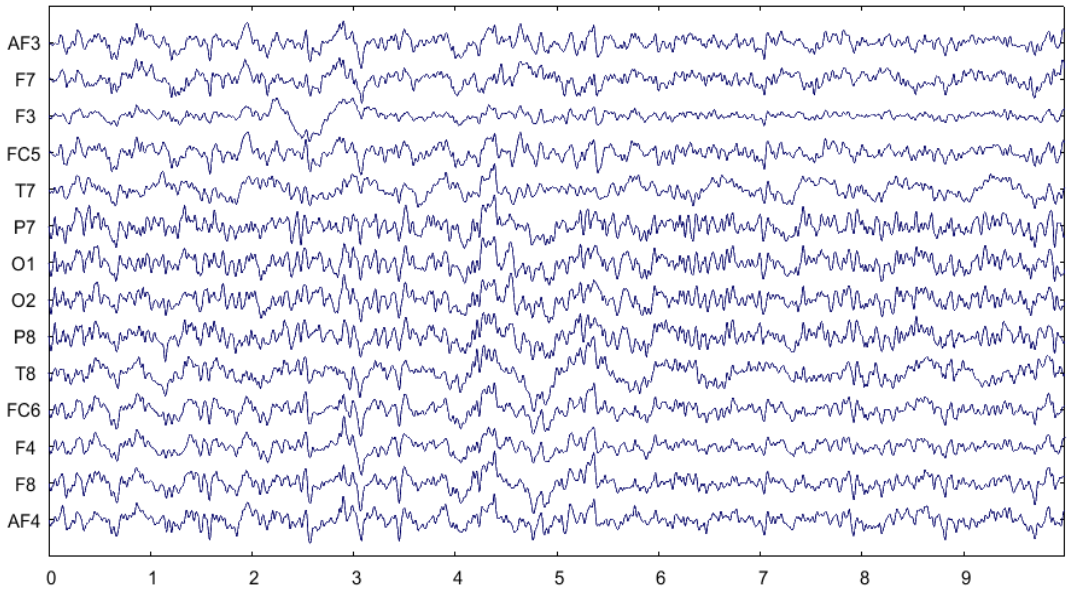


Figure 5.7. EEG signal after removal of eye artifacts

The classification of ankle movements based on EEG signal extracted features is investigated. With a combination of five time-domain features and three frequency domain features, a cubic SVM was utilised. The classifier was used to classify three classes, namely ankle dorsiflexion, ankle plantarflexion, and the resting state, and five classes, including all of the above ankle movements along with ankle inversion and ankle eversion. All EEG signals were segmented with a three-second window that overlapped by 50%. The initial step consisted of classifying using only the time domain features, which included RMS, Mean,

Standard deviation, kurtosis, and Skewness. In the second step, all frequency features, the Power band ratio of Alpha and Beta, Mean Energy, and Log Energy Entropy were utilised. All combinations of frequency features and time domain features were analysed in order to select the most accurate features. The use of multiple-frequency features decreased the accuracy as opposed to a single-frequency feature alone. Using all frequency features decreased the classifier's accuracy. The result demonstrated that combining time and frequency features increased the accuracy of the classifier.

Classifier	Time Domain Features	Accuracy
Cubic SVM (Classifying 5 classes)	RMS, Mean, Standard deviation, Kurtosis and Skewness.	67.90%
Cubic SVM (Classifying 3 classes)		72.60%

Table 5.1. Accuracy of the classifier using only time domain features

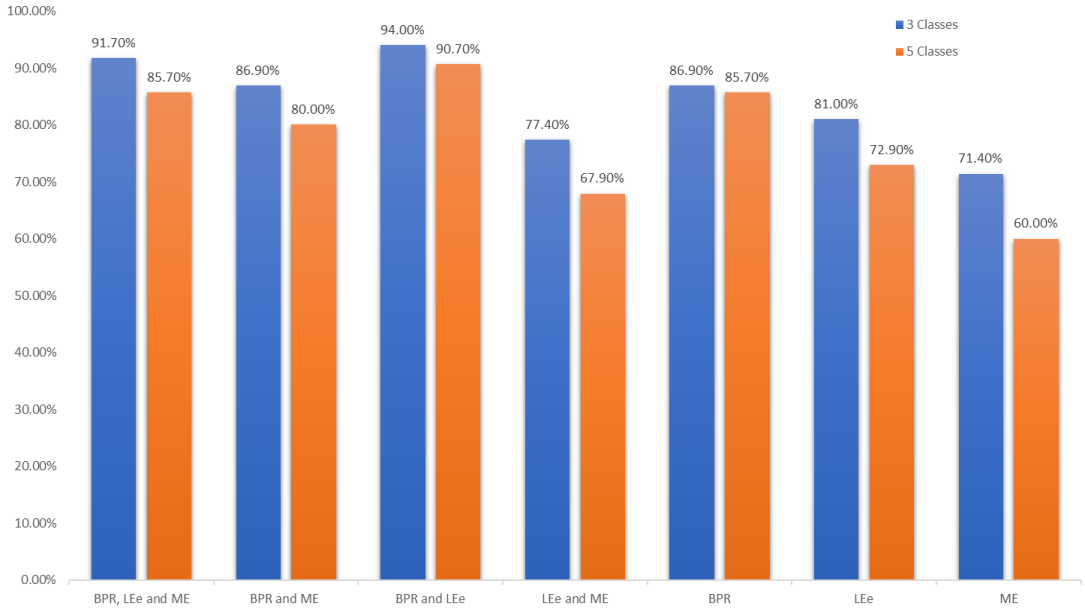


Figure 5.8. Accuracy obtained for different combinations of frequency features with all-time domain features

Table 5.1 illustrates that the SVM classifier performed poorly when only time domain features were used. Classifying 5 classes had an accuracy of 67.90% while classifying 3 classes had an accuracy of 72.70% of the time. The obtained results indicate that ankle dorsiflexion and plantarflexion are more accurately classified than ankle inversion and eversion. Since EEG signals are typically analysed in the frequency domain, it can be stated that time-domain features alone cannot accurately classify the signal. The addition of frequency features can enhance this precision. Figure 5.8 depicts the accuracy when frequency features are added to all-time domain features. Although the accuracy of individual features is low, the accuracy of the combined features is higher. The results also demonstrated that adding more features could reduce the classifier's accuracy. Mean Energy (ME) had the lowest accuracy by itself. ME aids in identifying the various states of brain activity. The fact that the state of the brain does not change during rehabilitation exercises is supported by the literature and scientific research. Therefore, feature values remained constant across all EEG tasks performed. When classifying 3 and 5 distinct classes, the accuracy was lower when using only ME along with all time domain features.

The next feature evaluated for classification was Log Energy Entropy (LEe). LEe outperformed ME by at least 14%. Even though the performance of classifying five classes is relatively poor, it was still superior to that of ME. During each motor imagery task, the signal's complexity changes, resulted in this improvement in precision. By analysing the amount of randomness in the signal with LEe, it was possible to identify the unique characteristics of each task. Although the combination of LEe and ME's use decreased the overall accuracy of both classes, it did not negatively affect the accuracy of either class individually. Band power ratio (BPR) alone had the highest accuracy of 86.90% for classifying three classes and 85.70% for classifying five classes. It was observed that the ankle rest state was very distinct because the subject was not particularly focused on performing any action; therefore, BPR enhanced accuracy. The alpha values were significant when the subject was in a state of rest, whereas the beta values were more significant when participants were engaged in the imagery task. The advantage of utilising BPR was the ability to determine the power amplitude between the two frequency bands, which played a crucial role in developing distinguishable characteristics for distinct brain activity states.

Similar to the combination of LEe and ME, the combination of BPR and ME produced lower accuracy, but at least 13% higher accuracy than the combination of LEe and ME. Among the different frequency feature combinations, BPR with

LEe was the most accurate. This combination achieved a classification accuracy of 93% when classifying three classes and 89% when classifying five classes. This result demonstrates that the combination of BPR and LEe can produce high classification accuracy for motor imagery tasks. The finding demonstrates that these characteristics can elicit distinct EEG signal characteristics. In contrast, the combination of all frequency features and time domain features reduced the accuracy during classification. This may occur for one of two reasons. First, ME as a feature was not distinguishable for any of the MI tasks performed, so ME did not contribute significantly to classification. The results indicate that ME alone and in various combinations significantly decreased the overall accuracy. ME had the same feature values for the majority of the tasks performed, and the feature values for each motor imagery task overlapped. Second, as more features are added, the risk of the classifier overfitting the training data increases. Increasing the number of features can prevent the classifier from learning the data's pattern, resulting in less accurate predictions and a reduction in accuracy. Despite the higher accuracy of EMG classification, having an accuracy of at least 85% makes the robot more stable during real-time operation when using EEG for patients with limited mobility. This feature combination along with time domain features, would be used to classify and test the data for EEG channel reduction using cubic SVM, as BPR and LEe produced the highest accuracy.

For channel reduction using EEGLab, a newton-optimized infomax algorithm was used. The obtained inverse matrix, W^{-1} displayed the source components for each channel. Using ICA to remove EEG artefacts, it was observed that channel 1 (AF3), channel 2 (F7), channel 13 (F8), and channel 14 (AF4) produced the majority of artefacts due to eye blinks and muscle activity. This is because the electrode is directly placed on the forehead. Despite the fact that ICA removed a portion of these artefacts, they contributed to the significance of these channel attributes and characteristics. Channels were chosen based on the absolute mean value derived from the weight matrix of each channel's components. Three different threshold mean values of 0.5, 0.6, and 0.7 were applied to select the channels. The corresponding channel was chosen if the mean value was greater or equal to the selected threshold values. Ten channels had a mean value greater than 0.5, seven channels had a value greater than 0.6, and four channels had a mean value of 0.7. As only two channels exceeded the classification threshold of 0.8, this scenario was not considered for classification. Figure 5.8 depicts the absolute mean values of all 14 EEG channels.

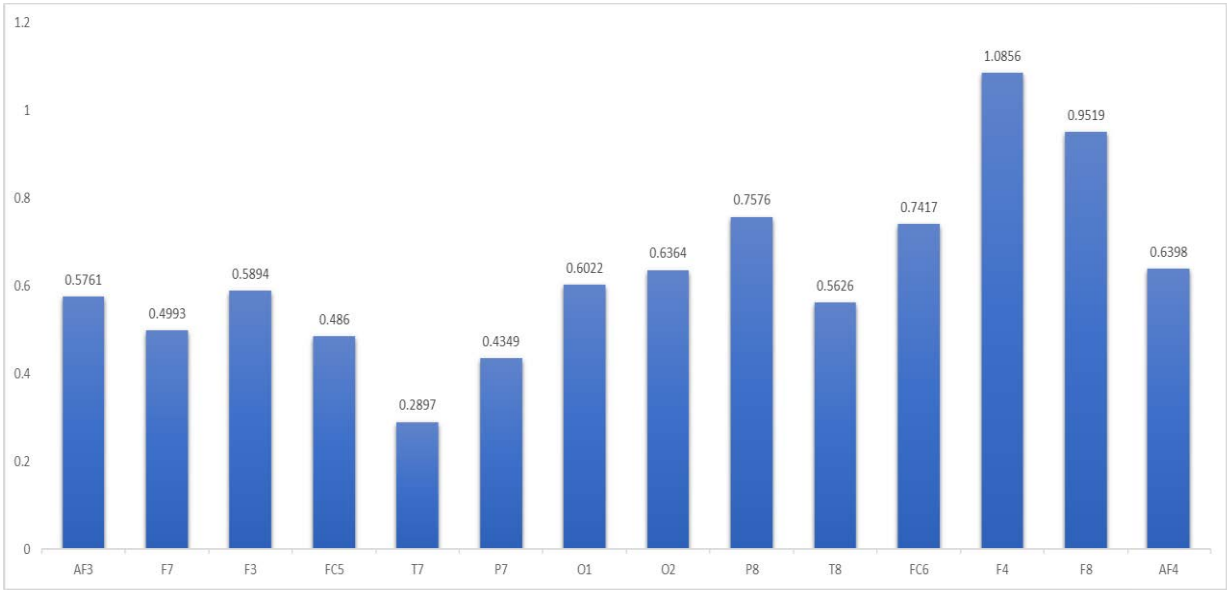


Figure 5.9. Absolute mean values obtained from ICA for each EEG channel

Using the selected time domain and frequency domain data, the selected channels were categorised with cubic SVM for the three distinct threshold values. All ten, seven, six, and four data channels were classified for all three subjects. The classifier's accuracy varied across three distinct subjects. The classification accuracy for three classes using ten channels was 92.22% on an average for all three classes, while it was 87.37% on an average for the five-class classification. When the number of channels was set to seven that had a threshold above 0.6, the average accuracy for three classes was 91.27% and for five classes, it was 86.93%. The least accuracy was observed when using the 4 EEG channels, which had a mean value above 0.7 with an average accuracy of 82.73% for three classes and an accuracy of 70.80% for five classes. Subject 2's results were comparable or improved when the number of channels was set to seven compared to using all channels. Subject 1 exhibited a negligible decrease in accuracy when using seven channels when compared to using all channels.

When the number of channels were set to 4, a substantial reduction was seen in the classification accuracy. Subject 3's accuracy was significantly lower than the accuracy of the other two individuals, however, even with only seven channels, the accuracy was comparable to that of utilising all 14. The classification accuracy for each subject for various channels is shown in Tables 5.2 and 5.3 for 3 and 5 classes, respectively. The findings indicate that seven channels are sufficient to

categorise the EEG data for ankle motion. Although there was a minor loss in accuracy across seven channels in a few instances, the decline was minimal.

3 classes				
Subject No.	*14 Channels*	*10 Channels*	*7 Channels*	*4 Channels*
1	*93.90%*	*93.00%*	*92.90%*	*75.00%*
2	*92.00%*	*91.20%*	*92.50%*	*80.00%*
3	*91.10%*	*90.00%*	*88.00%*	*82.70%*

Table 5.2. Classification accuracy of 3 subjects for 3 classes

5 classes				
Subject No.	*14 Channels*	*10 Channels*	*7 Channels*	*4 Channels*
1	*89.60%*	*87.90%*	*87.10%*	*65.00%*
2	*88.70%*	*87.50%*	*88.10%*	*74.20%*
3	*87.30%*	*86.70%*	*85.60%*	*70.80%*

Table 5.3. Classification accuracy of 3 subjects for 5 classes

To determine the effectiveness of the proposed strategy for reducing the number of channels, seven channels were chosen at random for classification. The accuracy was lower than anticipated, averaging 84.60% for three classes and 79.70% for five classes on average. The findings indicate that the mean value of an ICA weight matrix can be used to eliminate EEG channels that are not particularly significant, thereby enhancing computational efficiency and classification accuracy. Considering that the overall accuracy is only improved by 2 to 3% regardless of whether 14 or 10 channels are used, selecting 7 channels would be the better option. In addition, using fewer channels improves computation efficiency, which in turn accelerates the robot's response time. The accuracy of the 7 channels, which were obtained based on the mean threshold value of the inverse matrix of the EEG components which consists of 50% fewer channels, is still accurate at less than 2% difference. This also suggests that

utilising all available channels offers very little advantage over utilising the dominant channels that have been chosen. This novel method can minimise the number of channels while maintaining a slight decline in classification accuracy.

The next method proposed in this chapter involved using the MRMR method to reduce the size of the feature set to improve computation efficiency and reduce the feature dimension using a feature ranking process. According to the ICA channel reduction approach's findings, only seven channels are required for accurate classification. Considering this, we apply the method of reducing features to the features of the 7 EEG channels that we have chosen. According to what was found earlier, the total number of features available for classification is 49 since each channel possesses seven features. The mutual quotient approach was utilised to rank the EEG channel features. MRMR gave a score ranging from 0.2867 to 0 for each of the 49 features for the 3-class classification, with 0.2867 being the highest score and 0 being the score for the features with the lowest overall value. When performing the 3-class classification, features with scores lower than $1x10^{-5}$ were disregarded. When choosing the best features for a five-class classification, each of the 49 features received a score ranging from 0.173 to 0, with 0.173 receiving the highest score and 0 receiving the lowest score of the features. The features were chosen in such a way that any feature with a score lower than $1x10^{-5}$ were disregarded during the selection process.

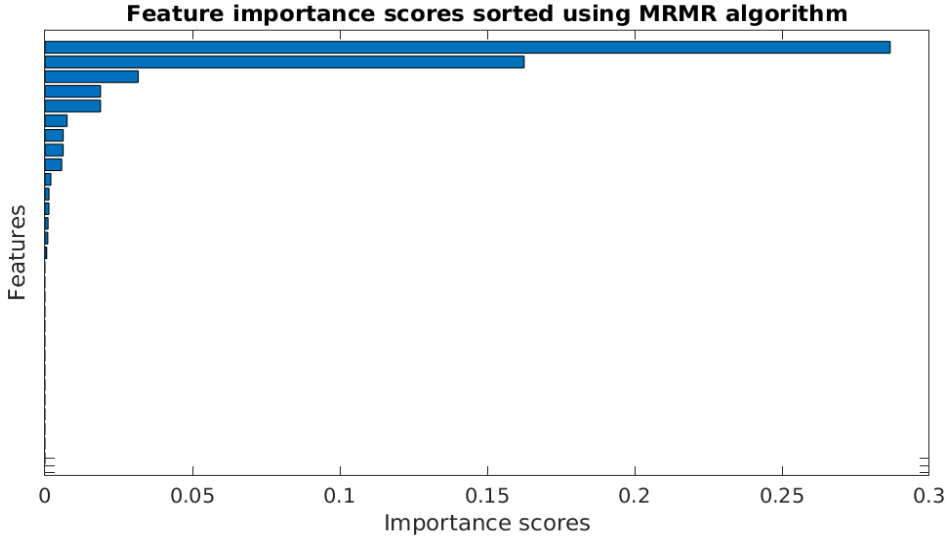


Figure 5.10. Feature scores for 49 features of 7 EEG channels for 3 classes

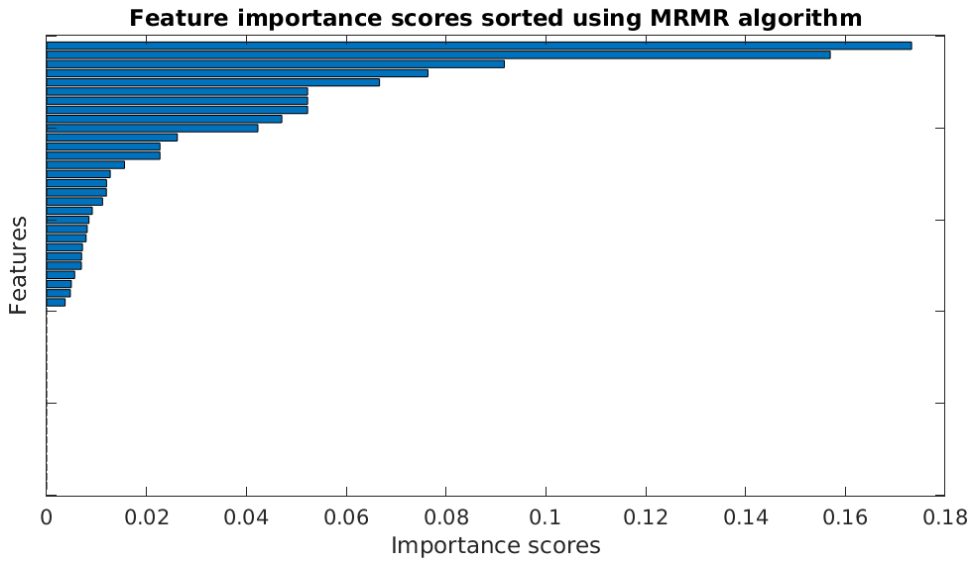


Figure 5.11. Feature scores for 49 features of 7 EEG channels for 5 classes

Before implementing the feature reduction method, the EEG feature data for all three participants were put together with their respective labels. This would be helpful in determining which features are more prevalent across all three subjects. According to the findings, 29 features had a score value that was greater than $1x10^{-5}$ for the 5-class classification, whereas only 15 of the features had a score value that was greater than $1x10^{-5}$ for the 3-class classification. For the classification of dorsiflexion, plantarflexion and rest state, standard deviation along with mean value received the lowest score, whereas log energy entropy and Skewness received the highest scores. It was shown that frequency domain features were significantly more prominent than time domain characteristics for the 3-class classification. The 15 features that were chosen for the classification did not include time domain features, such as the mean value or the kurtosis. Even though Skewness earned the third-highest score, only one skewness feature was present in the selected features. RMS was prominent in the time domain, whereas band power ratio was a prominent feature in the frequency domain, and both features were present in the selected list of features at multiple times.

While performing feature reduction for a 5-class classification that included ankle inversion and eversion, at least one feature from each of the 7 EEG features was present in the pool of selected features. In the same way as classifying 3 different ankle states that relied heavily on log energy entropy, classifying ankle states using five different categories did the same. The two frequency features, log

energy entropy and band power ratio, were the most prominent, and their distribution was fairly even across the list of selected features. The most notable time domain feature that the list contained was the standard deviation. Even though one feature of each kurtosis and mean had high scores, with the third and the fourth rank, respectively, there were only two of each present in the list, and the other feature of each mean and kurtosis had low scores. Each skewness feature generally received a high ranking; however, the selection only contained three of those features. RMS was the second dominant time domain feature in the selected list of 29 features. To test the accuracy of the selected features, cubic SVM was used as a classifier.

In this study, we evaluated the accuracy using a minimised feature set consisting of the 15 and 29 highest-ranked features for three and five class classification, respectively. This feature set consisted of 49 features extracted from the seven selected EEG channels. Following classification with this number of features, the obtained results indicate that the accuracy is relatively the same as that of using all features. When classifying three distinct classes, the accuracy dropped from the previous level of using all the features to just 2.4%. In contrast, the accuracy increased by 2.9% when all five classes were classified. Identifying optimal features is a necessary step for the prediction of EEG ankle movement, as suggested by this method. In addition, it reduces the computational cost, which in turn improves the implementation's real-time performance. Improving the human-machine interaction requires the robot to respond as quickly as possible, which is one of the most crucial factors.

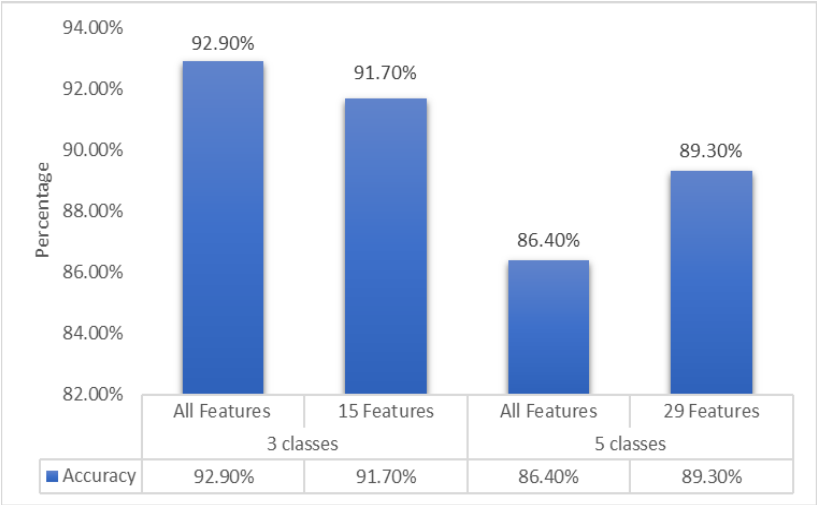


Figure 5.12. Accuracy after feature reduction for 3-class and 5-class classification

Given that the highest observed accuracy for three classes is 90.50%, and the highest observed accuracy for five classes is 89.30%, respectively, this tells us that several observations were incorrectly classified. The confusion matrix was plotted in order to observe the classes that were incorrectly classified. The classes were labelled as follows: class 1, for the resting state of the ankle; class 2, for the dorsiflexion of the ankle; and class 3, for the plantarflexion of the ankle, for a total of three classes. Ankle plantarflexion and the rest state of the ankle both had the highest error percentages, with the rest of the ankle being incorrectly classified as dorsiflexion and plantarflexion. The classification accuracy was highest for dorsiflexion, but a few of its components were classified as belonging to the plantarflexion category. All incorrectly classified classes of plantarflexion were all categorised as being in the ankle rest state. The confusion matrix for three class classification is illustrated in figure 5.13.

In contrast, for the five-class classification, the inversion was given the class 4 label and the ankle eversion was given the class 5 label. Ankle eversion had the fewest number of classified errors, while ankle inversion had the most. The remaining ankle states each had the same percentage of error, and the error was misclassified equally across all of the different categories. Every one of the classes shares relatively few of its components with the other classes. The most overlapping classes were seen with the ankle inversion. Figure 5.14 depicts the confusion matrix for the five-class classification. In motor imagery tasks, it is very difficult to imagine a particular movement for a particular time, so it is quite common for classes to overlap. It was observed that during certain trails, the subject did move their ankle, which helped them to imagine the task better. This was the case even though the subject was imagining the movement, which is why it was observed. Although each participant received 15 minutes of training, thinking about the inversion and eversion states of the ankle caused confusion and was much more difficult than thinking about the other states of the ankle. It's possible that this is one of the factors that led to most of the class being misclassified in the classes on ankle inversion and ankle eversion. The obtained results indicate that the proposed signal features along with ICA channel reduction technique and MRMR method combined with cubic SVM can improve the ankle pattern classification using EEG signals.

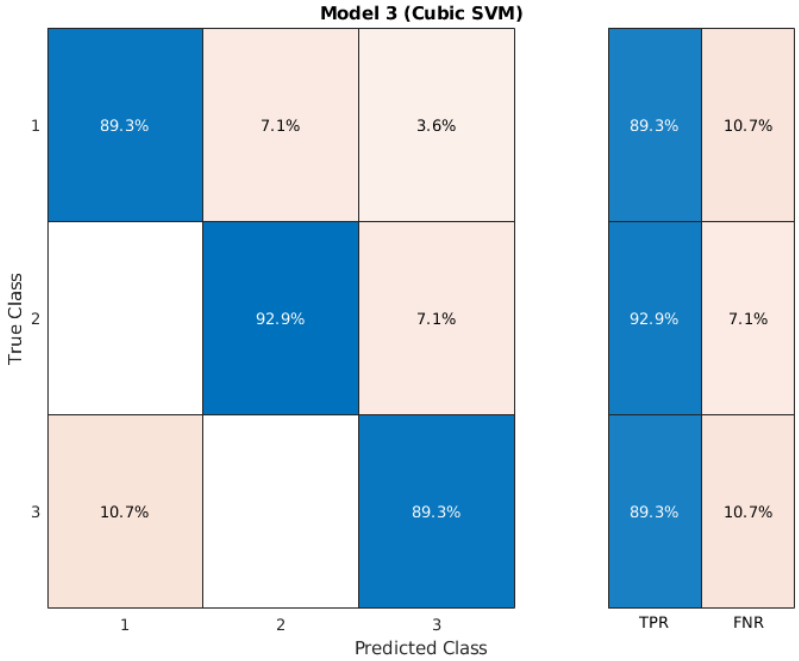


Figure 5.13. Confusion Matrix True Positive and False Negative Rates for 3 class classification

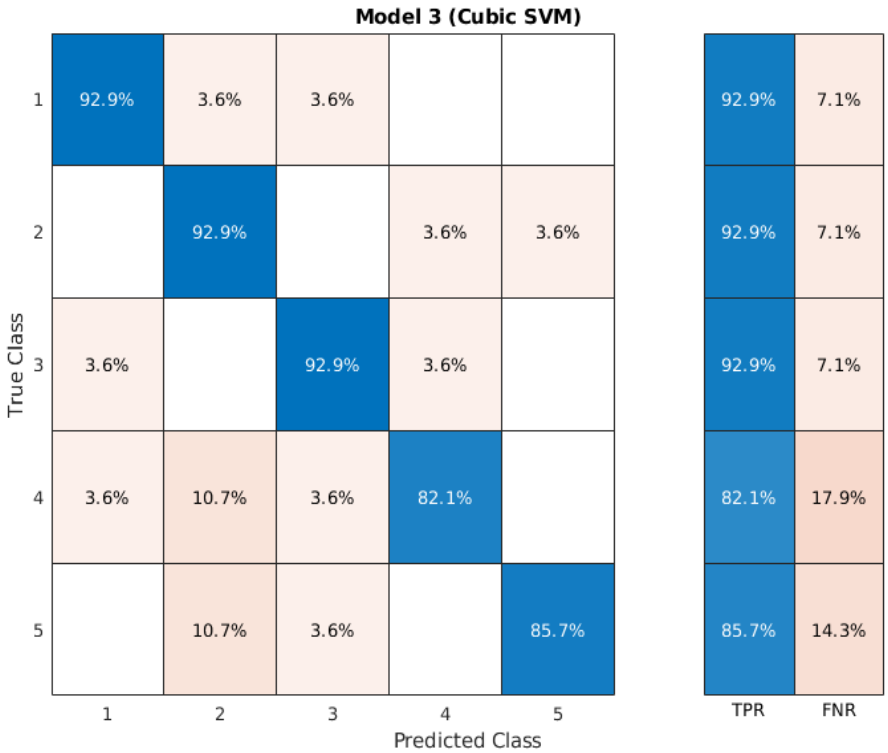


Figure 5.14. Confusion Matrix True Positive and False Negative Rates for 5 class classification

6 ADAPTIVE PID CONTROLLER USING REINFORCEMENT LEARNING

6.1 INTRODUCTION

Ankle rehabilitation requires the development of an efficient control method for assistive movement. Due to the non-linear nature of ankle movements, a system that can adjust to these non-linear changes in movement is required. Due to the complexity of modelling the movement between the robot and the user's ankle, an adaptive controller has not been explicitly designed for ankle rehabilitation. As safely and efficiently as feasible, the controller should be able to maintain the target output. Traditionally, in control engineering, Proportional-Integral-Derivative (PID) controllers are employed, and only minimum knowledge about the system is necessary to modify the controller. The controller works as a closed-loop system that generates an action to minimise output error. It can be fine-tuned by trial-and-error procedures and is relatively robust for systems with low uncertainty and few environmental changes. Non-linear systems are, nevertheless, inapplicable to PID controllers. Each new change in the system necessitates a significant amount of human intervention to tweak the controller. In some situations, external noise can trigger the controller even when no action is required. Tuning the controller does not offer a perfect solution. Ziegler-Nichols and Chien-Hrones Reswich algorithms have been utilised to autotune the PID controllers in order to mitigate these problems controllers [157, 158]. The performance of the system degrades when the system's dynamics change. These strategies were only helpful for systems with simple processes. In the early 2000s, several model-based adaptive PID algorithms that needed a number of assumptions to match system dynamics were proposed [159, 160]. In recent years,

PID controller tuning has increasingly utilised machine learning techniques to tune its parameters. These techniques require neural networks to be trained on labelled data, and supervised learning methodologies for complicated systems can be time demanding. Due to the continuous change in ankle mobility, developing an adaptive controller for ankle rehabilitation is essential.

Reinforcement learning (RL) is a machine learning algorithm in which an agent accumulates the highest rewards by exploring a given environment. The RL agent interacts with its environment to alter its actions as needed [161]. RL is ideally suited for applications that employ trial-and-error methods to determine the optimal answer. Since RL can converge to the desired solution, it can be employed in control systems where the system's process evolves nonlinearly over time. Therefore, RL can be used to tune a PID controller that can adjust its parameters in response to varying movements. The fundamental advantage of utilising RL is that the controller may explore all possible motions, learn about all environmental uncertainties, and then retune itself to perform the intended action. In robotic rehabilitation, robots are programmed to accompany patients through their exercises by following a predefined movement trajectory. To properly guide patients while interacting with a robot, the robot must understand how to interact with the environment and be capable of dynamically changing its characteristics, just as the human body can change the properties of its muscles based on force and gait actions [162]. Since we focus on active rehabilitation exercises, an electromyogram signal is used to control the robot, which is non-linear in nature. Adapting to these changes with a constant PID controller is difficult; hence an adaptive controller is required. Studies have demonstrated RL's utility in developing an actor-critic-based adaptive controller for several industrial applications [163, 164]. Here, the actor represents the control signal and performs an action. In contrast, the critic evaluates the agent's activity depending on the agent's state and environment.

RL can enhance tracking abilities in numerous ways. The algorithm and parameters used to construct it vary depending on the application. RL has been utilised in a number of studies to tune PID with little error [165-167]. Q-learning and policy learning are two of the most common learning techniques used to create an adaptable PID controller. The Q-learning method only applies to discrete actions and cannot function in environments with continuous action spaces. Due to the finite action spaces, Q-learning can lead to mistakes and poor performance

of the PID [168]. Since ankle movement is continual, the Q-learning method is not applicable to this investigation.

On the other hand, actor-critic-based policy learning methods can be utilised to tune continuous controllers. Deep Deterministic Policy Gradient (DDPG), which was proposed by Silver et al. [169], is highly effective for tuning PID controllers with continuous action spaces [170]. Reduced variance versions of the DDPG algorithm, known as twin delayed DDPG, were also created [168]. Although the technique has proven to be more efficient than the standard DDPG algorithm, it performs poorly in high-dimensional action space environments. This is owing to the algorithm's limited ability to explore its environment, which results in poor PID controller tuning. In some instances, the twin delayed DDPG algorithm requires multiple critic networks for a single action which can also increase computation. The modifications to the controller would affect the robot's output performance and the user's interaction with the robots. DDPG is designed to learn from each update by collecting input from its environment, performing an action, and then learning from that updated action. Because of this, the RL agent is able to observe its environment through a series of iterations, which ultimately results in the production of the desired output. Soccer robots and assistive control are a few of the applications based on self-tuning PID controllers [171, 172].

Due to the algorithm's simplicity and the fact that the agent controls the physical system directly, a method for creating an adaptive controller using the RL DDPG algorithm is developed in this chapter. This chapter describes the formulation of the observation and reward functions for the learning agent. The proposed actor-critic network would permit the RL agent to adapt to untrained paths. A 1-degree-of-freedom ankle robot's 3D model was developed in MATLAB Simulink to train the agent. Due to the hardware limitation, the agent is only trained for ankle dorsiflexion and plantarflexion. Initially, the ankle robot was evaluated with a conventional PID controller to determine its limits. This also aids in the determination of the action ranges for RL agents. Although the controller is adaptive in nature, relying solely on the RL agent for rehabilitation is unsafe, as the robot could move in an unintended direction, causing harm to the patient. To prevent this, a smoothing filter has been utilised in order to avoid unnecessary sharp movement. By calculating the amount of error present between the desired path and the actual movement of the robot, we can evaluate how well an adaptive PID controller is performing.

6.2 METHODOLOGY

Ankle plantarflexion and dorsiflexion are two movements that the developed model explores. Three muscles, the anterior tibialis, the lateral and medial gastrocnemius, provide the model's input of s-EMG signals.

6.2.1 Ankle Robot 3D model

MATLAB Simulink was used to create a 3D model of the ankle robot to train the RL agent. This model was endowed with characteristics comparable to those of the actual ankle robot. The robot is composed of five distinct solid components designed using solid works. A solver configuration block was required for the simulation to run on the model. Before the simulation could begin, the necessary parameters could be specified using this block. As required by the simulation, the solver applies a 1-D/3-D filtering connection with a filtering time constant of 0.001 seconds while treating the model's matrices as sparse so that their values are not zero. This ensures the accuracy of the simulation. The mechanical properties of the model are set to the world frame-block in which the three-dimensional model exists. The 3D model uses the frame as a reference point. A uniform mechanical configuration regarding gravity was applied to the entire robot surface. To distinguish between the motor and the moving pedal, a revolute joint was added to the moving joint, which is the pedal of the robot. The robot's output is measured at its revolt joint, which provides position and velocity based on input torque. Signal torque acts as the input to the revolt joint.

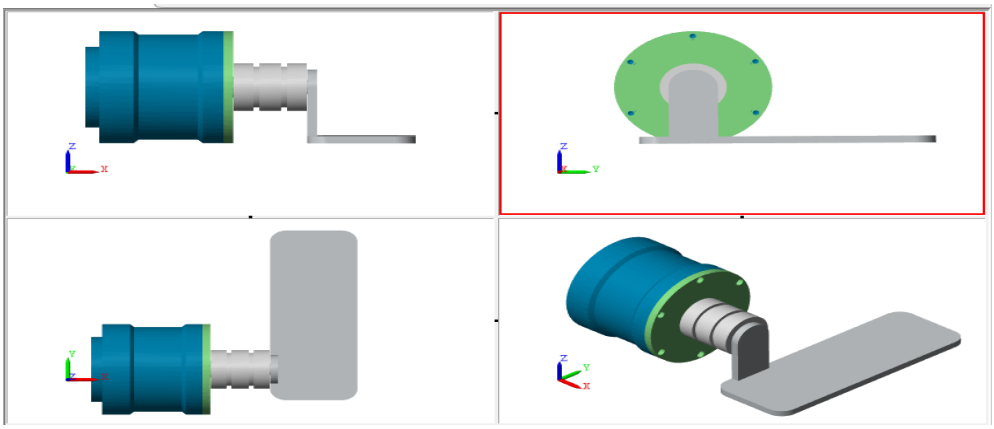


Figure 6.1. 3D model of the ankle robot

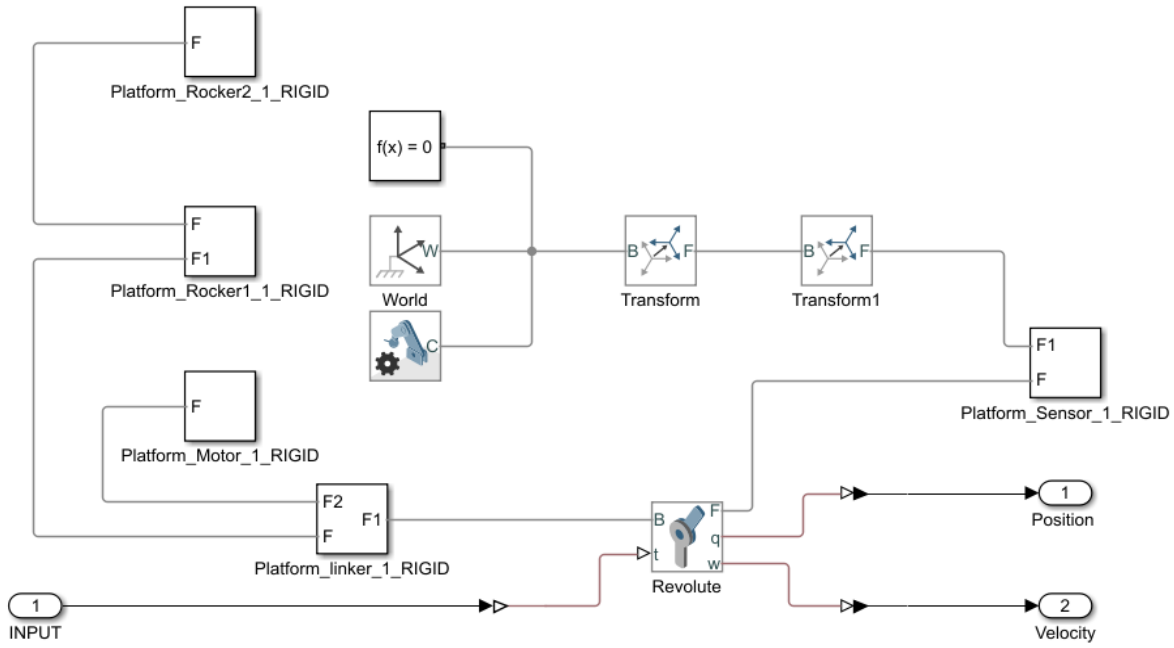


Figure 6.2. SimMechanis implementation of the Ankle robot in Simulink

6.2.2 PID-controller for the ankle robot

The classical PID controller operates as a negative feedback closed loop system which gives an error ($e(t)$) between the required and the actual measurement. The controller consists of three components, the first of which is the proportional component, g_p, in which the error is multiplied by the gain, g_p. The second component of the controller is the integral component, g_i, where the error is summed over time to determine its magnitude. The integral component is used to correct the small system error that has occurred. Consequently, a high value of, g_i for a small error can result in an overshoot. The last component of the controller is the derivative controller g_d where the rate of change of the error is measured. The controller's output is the sum of its proportional, integral and derivative components. One of the reasons the PID controller has become so widespread is because the final output is the sum of three very simple mathematical processes. Tuning the PID gain values is essential because entering inappropriate gain values can result in a dangerous overshoot during human-machine interaction, thereby injuring the user. The equation for a PID controller is described as follows:

$$g_p * e(t) + g_i * \int_0^t e(t)dt + g_d * \frac{d}{dt}e(t) \qquad (6.1)$$

Before training the agent, the PID controller's settings were manually modified to comprehend how the produced 3D model operated and to obtain the controller's maximum gain parameter values. This was completed in preparation for the agent's training. Otherwise, the RL agent would explore an infinite range of gain values to fine-tune the PID controller. The obtained gains would serve as both a reference and a limit for its actions. Without gain constraints, the agent would engage in an undesirable exploration of the gain value, which would lengthen the overall time necessary to educate the agent. To achieve the desired result, the proportional gain was set to 100, the integral gain was set to 5, and the derivative gain was set to 10. Since the robot also outputs velocity, the velocity of the reference position, which is a sinusoidal signal, is the input to the derivative gains. The robot's position feedback is fed back through the negative sum input of the feedback loop to both the proportional and integral gains. In contrast, the robot's velocity information is fed back into the negative sum input of the feedback loop for the derivative gain.

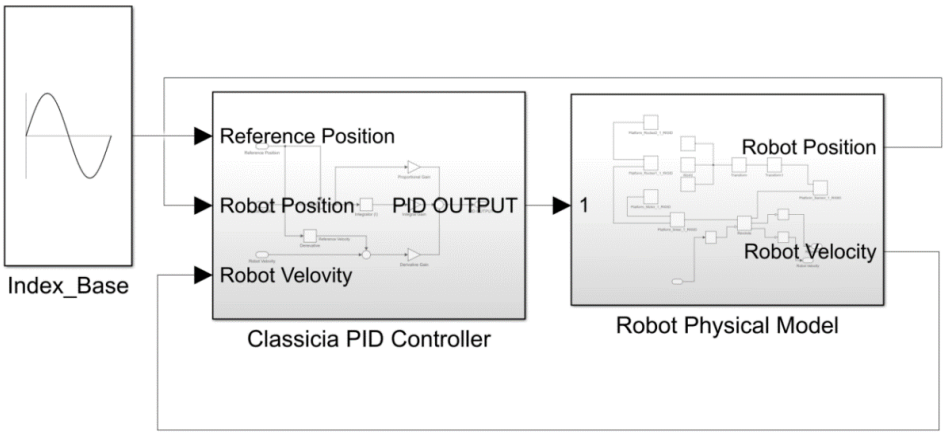


Figure 6.3. Simulink implementation of ankle robot connected to a classical PID controller

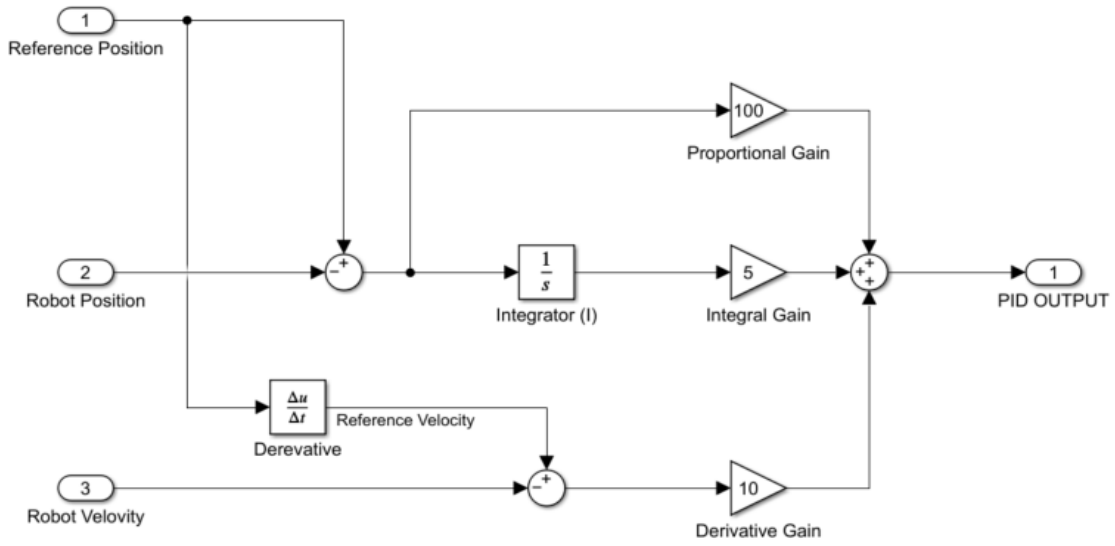


Figure 6.4. Implementation of classical PID controller in Simulink

6.2.3 Reinforcement Learning

The Markov decision process (MDP) is the origin of the concept of reinforcement learning [173]. An MDP is predicated on the notion that the likelihood of transitioning from one state to another is independent of the system's prior state. An RL agent can interact with a multi-state MDP environment. The agent enforces an action on or within the environment by observing the set of variables in a particular state. The agent receives rewards according to the activities it participates in while in the environment. In addition, it can acquire rewards for taking action and shifting between states. Depending on the agent's goal, the rewards could be positive or negative for them. The agent's aim is to maximise its own rewards per the desired outcome [173]. The components of an MDP are the states, the actions are observed within the environment, the rewards that are gained, and the transition model.

Time is a crucial component when addressing reinforcement learning problems compared to other machine learning techniques. At every time step, the agent performs an action depending on the action space available, A_s. The action, A_t is the element of the action space, A_s. The environment includes the conditions in which the activity occurs. The agent sees its state, S_t from the accessible state space S at each time step. The transition model P is the likelihood of the agent

moving from one state to the other when performing an action A_t. The agent earns a reward r_t for each activity it does. The agent strives to acquire maximum rewards, M_t with a discount factor ranging from 0 to 1. The equation for maximising rewards is as follows:

$$M_t = \sum_{t=0}^{\infty} \gamma^t r_t \tag{6.2}$$

While in a certain state, the agent will take the following action according to the specified policy (π) that connects its current state with possible future actions. A state value function $V_\pi\,(S)$ is used to assess the rewards from a particular state. Obtaining the state value function is not sufficient to make a decision; therefore, an action value function $V_\pi\,(A_t)$ is used to evaluate future rewards based on the current policy. Both of these value functions are utilised in order to maximise the agent's potential for receiving rewards. The decision-making process is improved by the state action-value function, which shows how the total rewards shift depending on the current state. Taking the average of all potential actions in a given space yields the value function. The relation between the state and action-value function is defined by:

$$V_\pi(S|\,A_t) = r + \gamma V_\pi\,(S) \forall\ S_t\ \in S,\ A_t\ \in A_s \tag{6.3}$$

An optimal policy Q_π is created by the state value function to maximise the agent's rewards which is described as the optimal value function $Q^*(S, A_t)$. The optimal policy in state S will select an action such the agent can obtain the maximum rewards. Therefore, the optimal policy can be used as a strategy to tune the PID controller.

$$Q^*(S, A_t) =\ max_\pi\, Q_\pi(S, A_t)\ \forall\ S_t\ \in S,\ A_t\ \in A_s \tag{6.4}$$

To maximise the number of rewards that can be earned, the agent needs to try out various actions and evaluate each one in relation to the rewards that were earned from the action taken before it. This requires the agent to explore the environment to accumulate additional knowledge concerning the environment. On the other hand, the agent should also make use of the best policy available at the time to maximise the rewards they receive; this strategy is referred to as agent exploitation. Therefore, the agent must find the right balance between exploring

and exploiting to choose the best course of action. This could lead to the agent achieving a negative reward in the long run because of sacrificing short-term rewards; however, in the long run, this could lead to the agent discovering the best course of action by finding the optimal policy.

Model-free and model-based reinforcement learning are the two distinct algorithms. Model-based algorithms provide detailed information about the environment, enabling them to make accurate predictions regarding rewards and transitions between states. By taking into account the present state of the system, a model-based algorithm provides the agent with the ability to foresee and prepare for the outcomes of potential actions [174]. It makes it easier for the agent to evaluate the various possible outcomes before taking action. AlphaZero is a model-based algorithm that was designed to master games with well-defined actions and states, such as chess [173]. There is a lack of availability of complete information of the model for the majority of RL problems. Suppose a model-based approach is used without all of the data; in that case, it can be challenging for the agent to learn, and the agent may discover a bias in the learned model, which can lead to an unsatisfactory outcome when it is put into real-time implementation. On the other hand, model-free algorithms are extremely common and relatively easier to modify and implement [173]. Since human ankle movement can't be modelled perfectly, this thesis focuses on model-free RL algorithms.

6.2.3.1 Temporal Difference Learning

Temporal difference (TD) learning is a model-free algorithm where the agent learns from the experience with complete information about the model of the environment [175]. This type of learning employs bootstrapping, where the agent estimates the next action from the learned experience without waiting for the actual outcome. The value function is utilised by the agent to make a prediction, and it is defined as:

$$V(S) \leftarrow V(S) + \alpha(r + \gamma V(S') - V(S)) \tag{6.5}$$

where,
α is the learning rate
$r + \gamma V(S') - V(S)$ is the TD error in the prediction δ_t

6.2.3.2 Actor-Critic Learning

Actor-Critic learning, in which the actor puts out a control signal and the critic examines the agent's behaviour, is one of the most effective RL learning techniques [176, 177]. Pure policy learning is inefficient for training online RL problems. Due to its bootstrapping capability, the TD learning approach may be used to solve this problem. The actor-critic learning approach concurrently incorporates the policy method, state value function, and TD learning. Actor-critic is another type of TD learning which has a different memory structure [173]. Simply said, the actor determines which actions to take, while the policy behaves as the actor. The agent's response or behaviour in the environment is determined by the actor. The critic serves as the value function and provides the actor with feedback based on the action. The critic assists in determining how effective an activity was and how the subsequent action should be modified. Critics constantly assess the results of actions to see if they meet or fall short of expectations. The critic of the actor's performance calculates the value function to evaluate the action. This feedback instructs the RL agent to reconsider its activity to achieve the intended result. The actor's learning technique is based on a policy gradient evaluation, which results in TD error [173]. The chosen action A_t, at state S_t may be evaluated with TD error. If TD error is positive, the current inclination to select a particular action A_t, would be increased; whereas if the error is negative it indicates that the current inclination to select action A_t would be decreased. The actor-critic architecture consists of a feedback loop as shown in Figure 6.5.

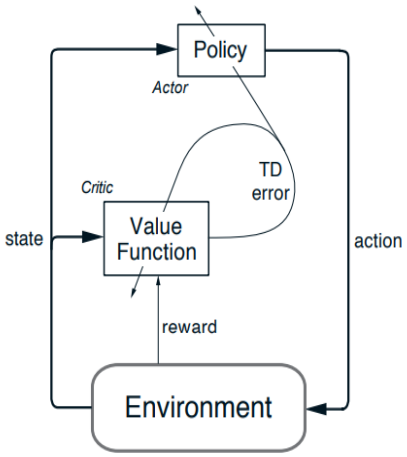


Figure 6.5. Architecture of actor-critic learning [173]

6.2.3.3 Deep Deterministic Policy Gradient

An advanced version of the model-free actor-critic learning for continuous spaces was proposed by Lillicrap et al. [178], known as Deep Deterministic Policy Gradient (DDPG). In this chapter, we use this type of RL learning method to tune the PID controller due to its efficiency and ability to adapt to environments with continuous action spaces. DDPG is developed from two RL algorithms which are Deep Q-Networks (DQN) and the Deterministic Policy Gradient (DPG) algorithm [169, 179]. The DPG method is efficient due to its deterministic policy as it directly outputs the action. In the DQN method, the optimal action is taken by obtaining maximum rewards instead of using the action's optimal values (Q-values). DDPG employs a deterministic policy by combining the actor-critic from DPG and the target network technique from DQN. The learning process is stabilized by developing target networks for both the actor and the critic. DDPG algorithm primarily depends on the actor-critic algorithm and the experience replay buffer [178]. The agent's experience is stored in the replay buffer for later review. By employing an off-policy technique in which an agent learns by performing a variety of actions while being in a state, DDPG learns based on the action-value function.

DQN uses deep neural networks to approximate its functions, which can solve the problem with high dimensions in continuous action spaces [179, 180]. One of the disadvantages of this method is that as the number of different actions to choose from increases, it makes DQN not feasible for training the agent. To limit the number of actions DDPG algorithm chooses the actor-critic approach based on DGP [169]. Here the actor is seen as a deterministic policy and the critic is parameterised as the action-value function. Both networks are represented as deep neural networks. The critic learns from the bellman equation, while the actor learns from the policy. The main goal of this project is to use the RL-based DDPG algorithm to create an adaptive PID tuning controller. The learning procedure for DDPG is given below:

1. Randomly initialise the critic network,
2. Initialise the target network for state and action-value function with weight
3. Initialise the experience buffer replay
4. Start 1st episode M
5. For action exploration, start a random process N.

6. Receive initial observation of the state
7. For the duration of each episode for a period of time (t) = 1, select an action in accordance with the current policy
8. Observe the rewards and the new state after carrying out the action.
9. Store the action and transition parameter to the buffer replay
10, To train both the DDPG critic and actor networks, sample a random minibatch of transition information from buffer replay.
11. Update the critic to increase rewards
12. Update the actor policy
13. Update the target networks
14. Update until convergence

Obtaining PID parameters is the prime objective of the proposed DDPG-based PID tuning. g_p, g_i and $g_{d.}$ PID controller is the environment, and the actions are the gain obtained from the agent. Gain values of the PID controller are received when step 9 of the DDPG learning procedure is executed, which runs in a closed-loop operation. The maximum number of training episodes M are defined before the training starts. We also define the duration of each episode so that the agent can learn about the path it has to follow. When an agent takes an action, it enters a new state, and the resultant change in its environment is recorded so that it can be used for training in the future. The policy parameter of the DDPG algorithm is updated as soon as one episode is completed. It should be taken into account that the PID parameter remains the same for an entire episode. This is done so that the performance of the obtained PID gain can be re-evaluated for the reference trajectory. It would enable the assessment of the controller's output properties for the signal, such as overshooting and undershooting and the rise time in relation to the reference signal. If the agent updates the PID gain values at every step, this will lead to instability and fast-changing of the PID gains [178, 181]. One set of the PID gains are received at each episode. The agent's prior experience is used to update the actor-critic networks. Once the agent is able to follow the reference path with the appropriate gain values, training is complete.

6.2.3.4 Rewards and Observation formulation for RL agent

Uncoupled from the process of supervised learning, in which data is labelled for the purpose of training, reinforcement learning employs the method of trial and error to acquire knowledge by exploring the environment. Developing a

reliable reward function is, therefore, necessary for the reinforcement learning process. Giving the agent a reward is essential each time it successfully carries out an action. When the agent makes a significant mistake, the agent should be subject to the appropriate punishment. A robust reward system would be an effective deterrent against unintentional behaviour. In this context, the aim is to create an adaptive control for ankle rehabilitation in which the robot moves in response to human intent. The reward function caters to error reduction and user safety.

The first proposed reward function is designed to accumulate negative rewards linearly with respect to the error in the reference signal. In contrast, when the error is less than a fixed threshold, positive rewards are given. If the positive rewards are given only when the error is zero, the agent would explore more states away from the target and terminate itself fast to reduce the accumulation of negative rewards. Providing positive rewards when the error is below a threshold lets the agents explore states which are closer to the goal, thereby improving training efficiency. In this experiment, a positive reward of +3 was given when the error was less than or equal to $\pm$ 3 degrees with respect to the reference input signal. If a lower error value is chosen the agent would not be able to explore and would always reach it terminal state. In all other cases, a negative reward of -1.5 was given to the agent. The error values were chosen based on trial-and-error process and it provided the best training results. If the error is equal to zero, the agent is given a +10 reward which is added along with the positive reward. The reference error of the time (t) can be defined as:

$$Ref_{error}\,(t) = \begin{cases} +3 * |Ref_{error}|, & if\ error\ \leq\ \pm 3^{\circ} \\ (10+3) * |Ref_{error}|, & if\ error = 0 \\ -1.5 * |Ref_{error}|, & otherwise \end{cases} \qquad (6.6)$$

The rate at which the robot can move is another aspect to consider. Patients who are engaging in physical rehabilitation activities might experience some level of discomfort if the robot moves at a rapid pace. Because it is difficult to mathematically define human safety, physical measurements, such as robot velocity, should be limited to improve user comfort and safety. The robot doesn't need to reach the desired angle as quickly as possible while the user is undergoing rehabilitation because doing so could be harmful to the user. Consequently, the robot's velocity component is punished with a negative reward of -0.5 for each episode to minimise sudden motion changes and sharp movements. At no point in

the training is the velocity component given positive reinforcement, as this ensures the user's safety. The velocity error for time (t) is defined as:

$$Vel_{error}\,(t) = -0.5^{*}|Vel_{error}| \tag{6.7}$$

Similar to velocity, torque is a physical parameter used by the motor to reach the desired position. When interacting with the robot, the user would experience torque, which is rotational force. High torque could injure the user, so torque receives a negative reward in every training session. The torque value of $\tau(t-1)$ was used as the input for the torque error function in this project. Because torque can cause more damage than velocity, torque is punished more severely than velocity. A threshold torque value was established to encourage the agent to generate lower torque values. If the measured torque exceeded the threshold torque of 2Nm, it would be penalised by a factor of 200. The actual ankle robot can produce a continuous torque of 50Nm, which exceeds the required torque and surpasses the torque produced by a healthy ankle in both males and females [182]. To prevent injuries, the robot should not exceed the threshold value. A torque value of 2Nm is sufficient for the robot to maintain the ankle and foot pedal in place. The torque error of the reward function for time (t) is defined as:

$$\tau_{error}\,(t) = \begin{cases} -|\,\tau - 1|, & if\,|\,\tau - 1| < 2Nm \\ -|\,\tau - 1| * 0.02, & if\,|\,\tau - 1| > 2Nm \end{cases} \tag{6.8}$$

The subsequent essential parameter is the agent's terminal condition (IS done function) for each episode. The condition enables the agent to determine whether it must run for the duration of the entire episode. Since there are numerous available states, it is not necessary to explore all of them. Episodes in which the agent is unable to follow the reference path will not properly train the agent. Therefore punishing the agent with high negative rewards can stop the agent from exploring unnecessary states. For ankle rehabilitation, the states are the various ankle positions within the ranges of ankle dorsiflexion and ankle plantarflexion. If the agent is permitted to move outside of ankle movement range, the user's ankle would be severely injured. Restricting the agent from exploring undesirable states can reduce training time and increase user safety. If non-viable states are explored, this function instructs the RL agent to terminate the current episode. The maximum movement range of the ankle joint is 45.8 degrees for plantarflexion and 30 degrees for dorsiflexion [30, 31]. If the agent explores a state with greater than 15% of the ankle's range of motion, the termination condition is

triggered. If the agent exceeds the standard range of motion by more than 0.7 radiance (40 degrees) for dorsiflexion or less than -0.9 radiance (-51 degrees) for plantarflexion, the function will terminate that episode. A negative reward of -5 is given whenever the agent exceeds the predetermined boundaries to prevent the agent from reaching the terminal condition. The additional 15% is allowed so that the agent has more room to explore and prevent wrongful termination. The definition of the termination condition for time (t) is as follows:

$$Terminal_{error} = \begin{cases} -5, & if > 7 \; or < -9 \\ 0, & Otherwise \end{cases} \tag{6.9}$$

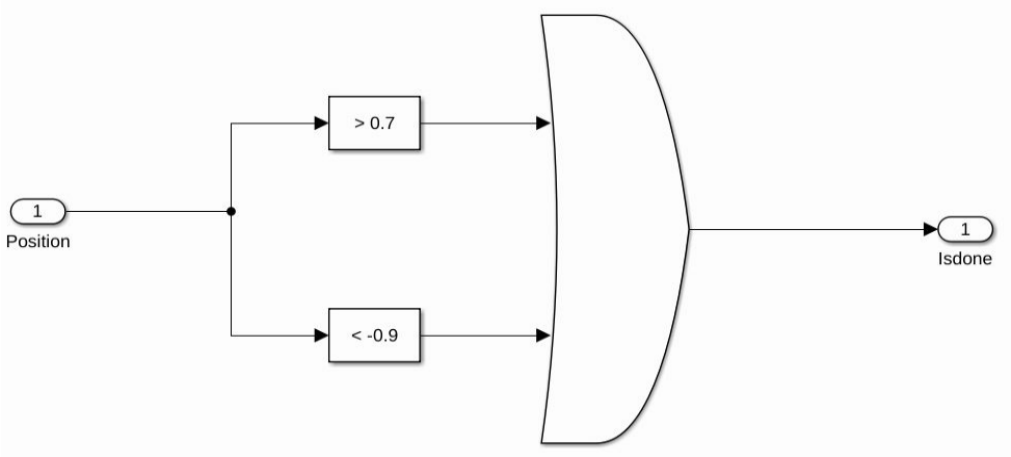


Figure 6.6. Simulink implementation of the terminal condition

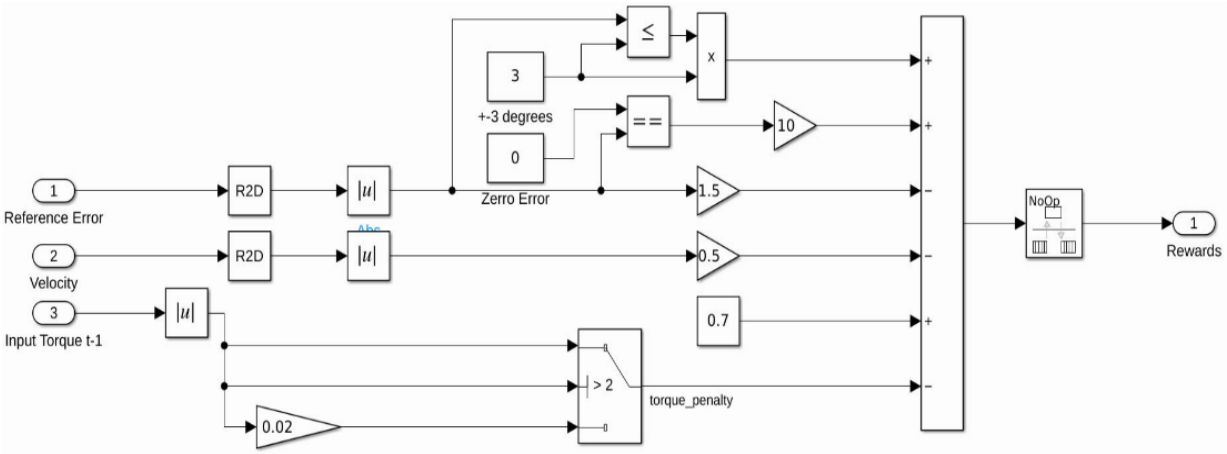


Figure 6.7. Simulink implementation of the reward function

Since the reference training signal is generated in radiance, it is converted to degrees for easier programming and analysis. The terminal condition, if triggered, is added to the output of the rewards. They encourage exploration, and to avoid early termination, a positive reward of 0.7 is awarded to the agent. This reward stays constant and is awarded for every episode. The reward will only be significant if the agent stays within the defined limits and conditions. The total rewards $R(t)$ is the sum of all the reward variables discussed above.

$$R(t) = Ref_{error}(t) + Vel_{error}(t) + \tau_{error}(t) + Terminal_{error} + 0.7 \qquad (7.9)$$

The agent is rewarded with benefits that originate from the surrounding environment. The agent's training observations are used to estimate and update the value functions and policies, but this happens in the background. In this scenario, position and velocity are the observations emitted by the environment. Observations are variables that depend purely on the current state of the agent. If the value function is the only form of observation employed by the agent, in that case, that agent may have only a limited understanding of the state and environment in which it finds itself [173]. When used on their own, value functions are insufficient for directly observing certain information, such as the agent's historical position values. A number of complications can arise if the environment can only be partially observed. One solution for the problem of partial agent observability is to make use of the history of the observations made of the environment in the past. For the purpose of forecasting future values, having access to historical values can be of tremendous assistance. An alternative interpretation of the observation is that it is a replication of the agent's actions.

For rehabilitation application, in this project, we observe five parameters: errors with regard to the reference input, the robot's current position, the velocity of the robot, PID gains, and torque produced by the robot. Since historical values can benefit the agent, the previous values of all the above parameters are taken. The agent's primary aim is to reduce the error when compared to the reference signal; therefore, overserving the reference is key for this application. Therefore, we observe the current error and the previous error till $(t-2)$. For reference errors, the agent observes $Ref_{error}(t), Ref_{error}(t-1)\ and\ Ref_{error}(t-2)$. The reference error is not a parameter directly observed from the robot. It is obtained from the difference in the robot's movement path when compared to the reference path. Unlike the reference error, the position of the robot is observed to understand how the robot moves along the reference path; it does not observe the

error caused by that path. To correct the position of the robot, the agent observes the current position as well as the previous position ($Position\,(t)\ and\ Position\,(t-1)$). As discussed before, for rehabilitation, we do not require the robot to get to the position as fast as possible, therefore the current $Velocity\,(t)$ and previous velocity value $Velocity\,(t-1)$ of each episode is observed. Similar to velocity, torque has been observed for $\tau(t-1)\ and\ \tau(t-2)$. The position, velocity and torque observation are taken directly from the robot's behaviour. Since the aim of the agent is to find the variables of the PID controller, observing the gain generated by the agent is crucial. Here we observe the previous gain generated by the agent, which are, $g_p(t-1)$, $g_i(t-1)$ and $g_d(t-1)$. The accumulation of historical data helps the agent learn from experience and generate the appropriate PID gains. In total, 12 observations are made, and since there are multiple observations, a MUX block is added to make all the observations into a single vector. The implementation of observation parameters and the DDPG agent in Simulink is shown in Figures 6.8 and 6.9, respectively.

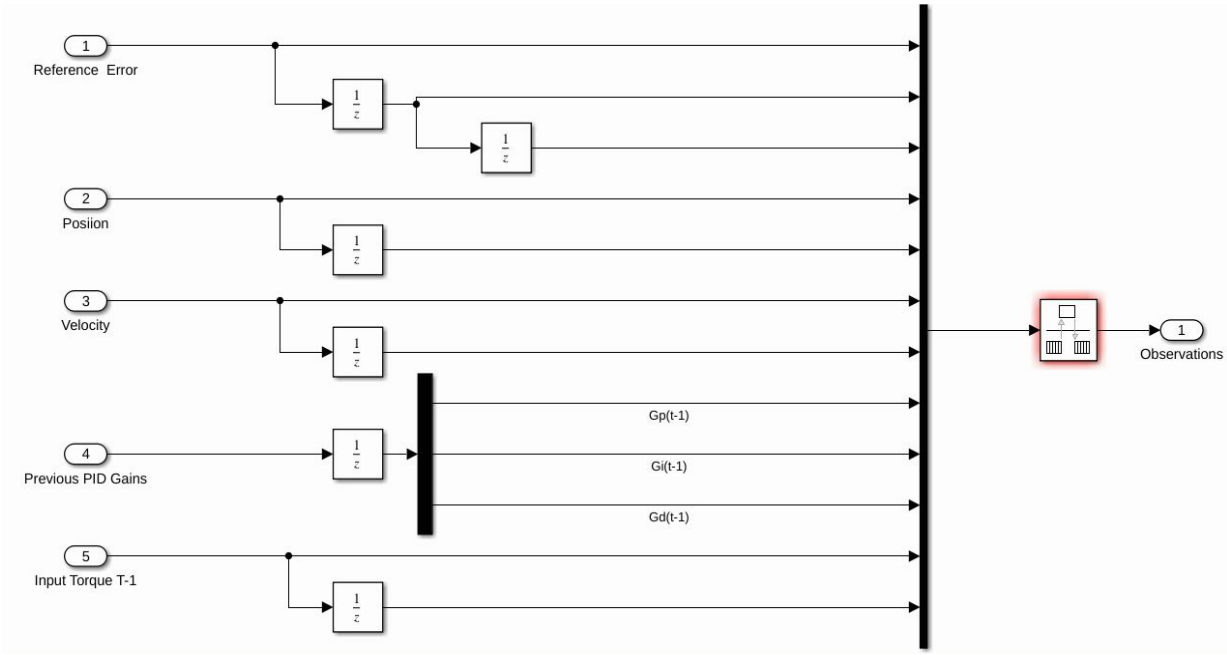


Figure 6.8. Simulink implementation of the observations

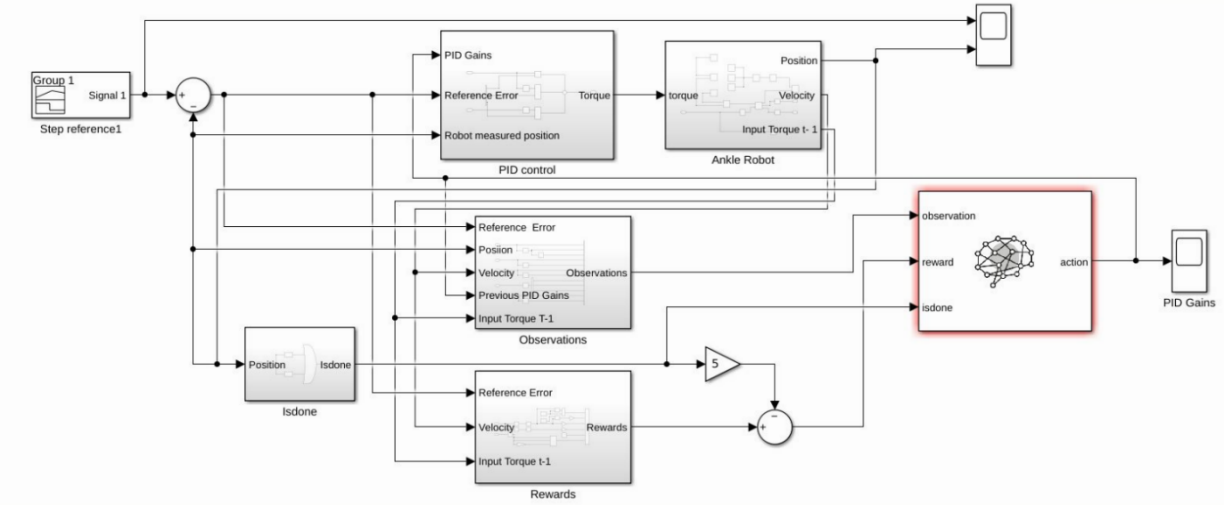


Figure 6.9. Simulink implementation of the RL DDPG agent

6.2.3.5 Agent Learning Procedure

The proposed DDPG agent is designed to perform an action that generates the PID controller's gain parameters g_p, g_i and g_d. The action vector is defined as $A \in a^{3x1}$. Therefore, the agent can only affect the robot's movement by altering the gains of the PID controller. The robot was able to follow the path of a sine wave when the proportional gain was set to 100, the integral gain was set to 5, and the derivative gain was set to 10. Based on this, the action ranges' upper and lower limits were set between 0 and 100. Without this limitation, the agent would explore infinite action ranges. The agent receives 12 parameters for observation which the critic network can use to guide the agent. The actor-critic network's design depends on the required output. Since the robot only moves in one dimension, many hidden layers of the neural network are not required; moreover, having a large number of hidden neurons might not help the agent adapt to the untrained path. The total number of layers required for this project was calculated by taking the average of the total number of observations and actions. Since we have 12 observations and three actions, this averaged to 7.5, which was rounded to 8, therefore using eight hidden neurons in the project.

The critic network is given two inputs: the agent's observation and actions. The observations and actions connect the eight hidden neurons via linear layers known as fully connected layers (fc1). The layers multiply the input by a weight matrix. The first fully connected layer had a weight matrix of 8*12. All the

observations and actions are connected to each of the eight hidden neurons. The fully connected layers are connected to activation functions which are crucial to the neural network as they introduce nonlinearity to the network. This would enable the neural network to learn and solve non-linear problems. The activation function can help to avoid saturation, where the signal can have a negative value. To solve this, we use LeakyRelu ($LR(x)$) [183] activation function which modifies the negative values with a desired factor. The LeakyRelu layer was implemented with a factor of 0.2. It can be defined as:

$$LR(x) = \begin{cases} x \ if & x \geq 0 \\ 0.2 * x \ otherwise \end{cases} \tag{6.10}$$

The LeakyRelu layer was further connected to another fully connected layer (fc2) with a weight matrix of 8*1. The next layer involved the outputs from the fully connected layer from both the observation (fc2) and actions (fc2) to be added with each other. After both the connected layers were added, it was connected to another LeakyRelu activation function. The final network layer provides the critic output. Similar to the critic layer, the actor layer is connected to two fully connected layers with 8 neurons, both of which have a LeakyRelu activation function. And the output of the third fully connected layer (fc3) is connected to a tanh activation function. The function helps the result to stay within the upper and lower bound of 1 and -1 [184]. This allows the agent to learn policies. The function would help convert a positive value to 1 and the negative values to -1 as their outputs. Since the tanh activation function's output is centred at Zero, it's easy to map the values to negativity, neutrality, and positivity extremes. Because its values range from -1 to 1, the hidden layer's output is either 0 or very close to it. It helps to centre the data and makes learning considerably simpler. After the tanh function, actions are generated by the network, which is the gain of the PID controller.

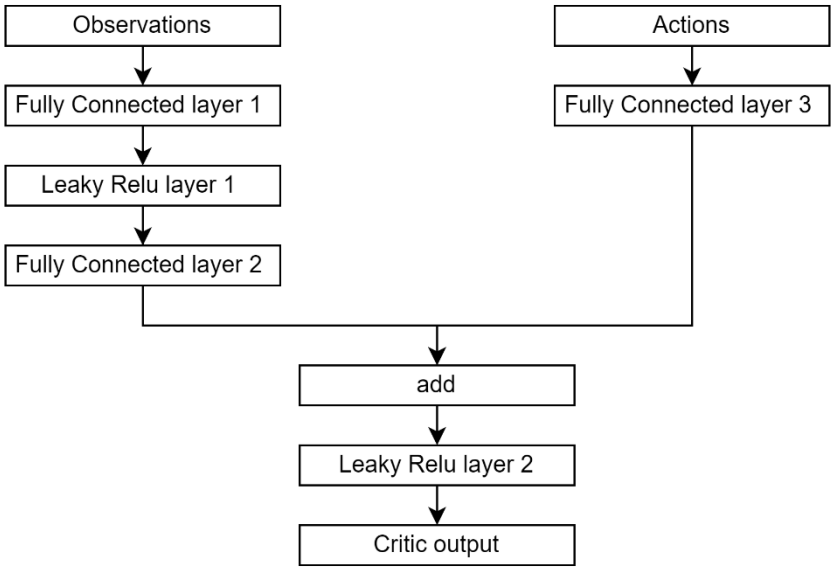


Figure 6.10. Structure of Critic Network

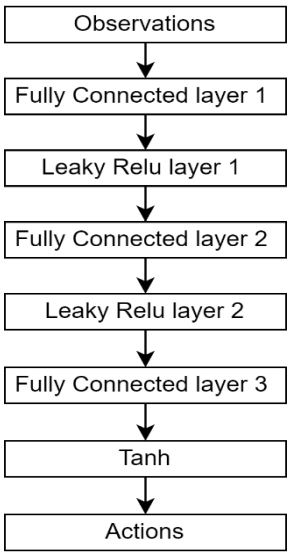


Figure 6.11. Structure of Actor Network

6.2.3.6 Agent Hyperparameters

Hyperparameters are vital components that help control the learning process's behaviour, which is adjusted before training to achieve a particular goal. It helps to produce the optimal result and increase model predictive accuracy.

Tuning these values are depended on the application, and there is no set approach to selecting the correct variables. Most of the hyperparameters chosen for this project were based on previous studies or as MATLAB defaults. MATLAB default values were well-suited for RL-based control applications. For example, Adam optimiser function was used, a default optimiser algorithm provided by MATLAB that updates the learning rate of each network for the actor-critic network [185]. Choosing the appropriate parameters was based on a trial-and-error method where the impact of each parameter was studied to improve the agent's learning.

The learning rate of the agent is a key hyperparameter of reinforcement learning. The rates are set between one and zero. A learning rate close to one will run fast but never find the desired solution, leading to instability while training. However, learning with lower values avoids these risks but would take a lot of time to train. An appropriate learning rate is small enough to converge to an optimal solution but, at the same time, should be big enough to be trained within a set period. 10^{-1}, 10^{-3} and 10^{-6} are the commonly used learning rates for both the actor and critic network [186-189]. Some of the experiments had different learning rates for the actor and critic network, but this did not show any meaningful changes to the overall output. Most of the experiments had a successful outcome with a learning rate of 0.001. An L2 regularisation factor of 0.0001 was used in both networks to avoid overfitting. Weights are pushed closer to zero by L2 regularisation, but they are not made absolutely zero. At each iteration, L2 regularisation operates like a force to eliminate a small fraction of the weights. Therefore, there can be no such thing as zero weight. Along with the L2 regularisation factor, a gradient threshold was set to 1, which helps limit the change in network parameters while training.

The discount factor and the learning rate are essential in reinforcement learning. The value of future rewards are reduced by the discount factor, but when the reward is entirely negative, it reduces the cost of subsequent actions. The discount factor is chosen between 0 to 1. The discount factor determines the extent to which reinforcement learning agents value immediate rewards over those further in the future. If $\gamma = 0$, the agent will only learn only from the action that takes place immediately after a reward, whereas if $\gamma = 1$ the agent will weigh the action based on the total sum of all the future rewards. However, having $\gamma = 1$ can add lots of irrelevant information, which can create a huge variance and make it hard for the agent to learn. Since we only used 1DOF, there are not many

factors to be concerned about in this project; therefore, the discount factor is set to 1 in this experiment.

The next hyperparameter set is associated with how fast the network is updated. Parameters such as the target smooth factor and the target update frequency, impact the solution provided by the network. To better update the actor-critic network, an additional neural network called a target network is used to predict the reward for an action and then compare it to the actual reward to determine the error. The target smooth factor controls the level of smoothness in the target values of the actor and critic network. It helps adjust the temporal difference between the estimated and target Q-values. A larger value results in a slower update, whereas a smaller value results in a more sudden update. The goal is to establish a better balance in the learning process between stability and responsiveness. The network parameters are updated at every step using a smoothing factor. Finding the optimal smoothing factor is challenging and is depended on the application. Here we used the manual tuning method by observing how the performance changes with each stage. MATLAB has provided a default value of 0.001, but this decreased the agent's commission. For this project, 0.005 was taken as the target soothing factor. On the other hand, target update frequency describes the regularity with which the target network is modified. Target network weights are updated at intervals set by the update frequency. In this project, we use MATLAB's default frequency of 1Hz, which is well-suited for this application.

Exploration is the next hyperparameter used by the DDPG agent to find the optimal solution. Agents utilising reinforcement learning algorithms must explore to discover how to predict and control an unknown environment. The DDPG algorithm heavily relies on exploration to help the agent learn new states and actions that may yield greater rewards. Exploration in DDPG frequently involves introducing noise into the action space, such as by employing Gaussian noise, whose standard deviation decreases with time. This encourages the agent to explore alternative courses of action [190]. The standard deviation determines the amount of noise added to the action path. A low standard deviation results in less exploration due to the agent's propensity to repeat past behaviour. A high standard deviation, on the other hand, encourages the agent to stray further from its target by taking actions that are less similar to those it has previously taken. Here, we introduce noise that is within 10% of the desired action path, which lies between -40 and -51 degrees. If the agent is not making significant progress,

increasing the standard deviation will encourage it to explore more, while decreasing it will encourage it to exploit the environment. When annealing noise, setting the correct decay rate for the standard deviation is crucial. The decay rate determines the rate at which the standard deviation decreases with time. When the rate of decay is high, the standard deviation drops down sharply; otherwise, it will decrease slowly. It is necessary to set the decay rate's parameters to enable exploration for a desired training segment. The number of training episodes will also affect the decay rate, which may result in only a few states being explored. The trial-and-error method revealed that a faster decay rate could result in poor training, whereas a slower decay rate, while requiring more time to train, can improve an agent's training. This project selected a standard deviation decay rate of 0.00001.

The last hyperparameter is the experience buffer. An experience buffer is a tuple of data structures that stores the experience of an agent, which consists of the current action A_t, current state S_t, reward r_t and the next state S_{t+1}. These experiences can be used to modify the agent's value function or policy through a method known as experience replay. It can help the agent learn from a wide range of experiences and reduce the correlation between updates, which makes learning more stable and effective. The experiences are stored as batches known as mini-batches. They help average the learned behaviours to prevent large reward swings from diverging and unstable training. The experience buffer lets the agent review and learn from previous situations, improving learning efficiency and stability. The buffer can only hold a certain number of experiences, and as new ones are added, the old ones are taken out. The training time and computational costs will rise significantly if the minibatch size is increased. Since only a fixed number of experiences can be saved, the buffer size would impact training the agent. With a longer buffer, the agent can learn from more experiences, but it also needs more memory to store those experiences. A smaller buffer limits the number of events from which the agent may learn while also using less memory. How long the buffer should be will depend on the task and the available resources. So, the size of the experience buffer and the minibatch are two important hyperparameters that must be considered when experimenting. Since no extensive research has been conducted on experience buffer design, the values are chosen based on the trial-and-error method. Each episode in this project contains 20 experiences; therefore, training 5000 episodes would be a total of 100000 experiences. Here we take the buffer length to be 10% of the total experience, which is 10000. The minibatch size was set to 256, which improved training stability.

6.2.4 System Setup

In this context, the end goal of an adaptive controller is to have a system that can adapt to the user's ankle movement and improve human-robot interaction. The results from Chapter 4 demonstrated how ankle position could be derived from the EMG signal. Therefore, the estimated ankle joint angle from the EMG signal would be used to test the adaptive controller. A random step function with different amplitudes was generated to train the agent. If the agent were trained only for gait movement, it would not be able to adapt to other movements. The experiment's result will show the drawback of training the agent with only gait movement. To enable the agent to start from a random position, a reset function has been implemented where the agent would have a random initial position between ± 10 degrees after each episode. One of the disadvantages of RL-based adaptive controllers is the limitation of human input during real-time implementation. Though the controller is adaptive, it is not necessary to have all the experiences to run all the time safely. Therefore, a filter must be applied to smooth out the sharp movements and mitigate any noise. Here we use a simple smoothing filter to attenuate high frequencies.

6.3 RESULTS

The training and experimentation of the adaptive controller was performed on the UTS high-performance computer (HPC) cluster. A GPU-based node was used, which consisted of two NVIDIA Quadro RTX 6000, each consisting of 24 GB and two intel Xeon CPUs with a RAM of 192 GB. GPU was the device used to train the actor and critic network. Parallel computation was enabled in MATLAB to increase training speed. During training, all DDPG agents that had a reward above ten were saved. The best agent to follow the trained path with minimal error was selected for experimentation and analysis.

The goal of the first experiment is to compare the results of the adaptive controller with the standard PID controller. Several input signals, including sine wave, the trained step function and the ankle gait pattern recorded from an isokinetic dynamometer as explained in chapter 4 of this thesis, were compared. In the following experiment, we test the controller when placed in a dynamic environment. The input torque to the motor is reduced to a maximum of 10% for a period. This is to test real-time situations where motor failure could occur wherein

the motor cannot provide the necessary torque to move the ankle. Calculating the root mean square error (RMSE) of each input signal was used to assess the performance of the adaptive controller against the conventional PID controller. A lower RMSE value indicates a better trajectory indication that the robot follows closely to the reference signal. Root means the square error was used to calculate the difference between the robot's trajectory and the reference trajectory. It enables us to determine the magnitude of the system error.

A. Experiment 1

First, the trained step signal was tested with the classical PID controller (Figure 6.12). The classical PID controller's gain values were determined by averaging the gain values of the adaptive PID controller so that the two could be fairly compared, which was 100 for g_p, 20 for g_i and 4 for g_d. The previously set gain values of 100, 5 and 10 for g_p, g_i and g_d respectively were used to find the limits of the actions of the agent. Comparing the results with the initial set values of the classical PID controller provided very poor results and therefore, was not used in the following experiments.

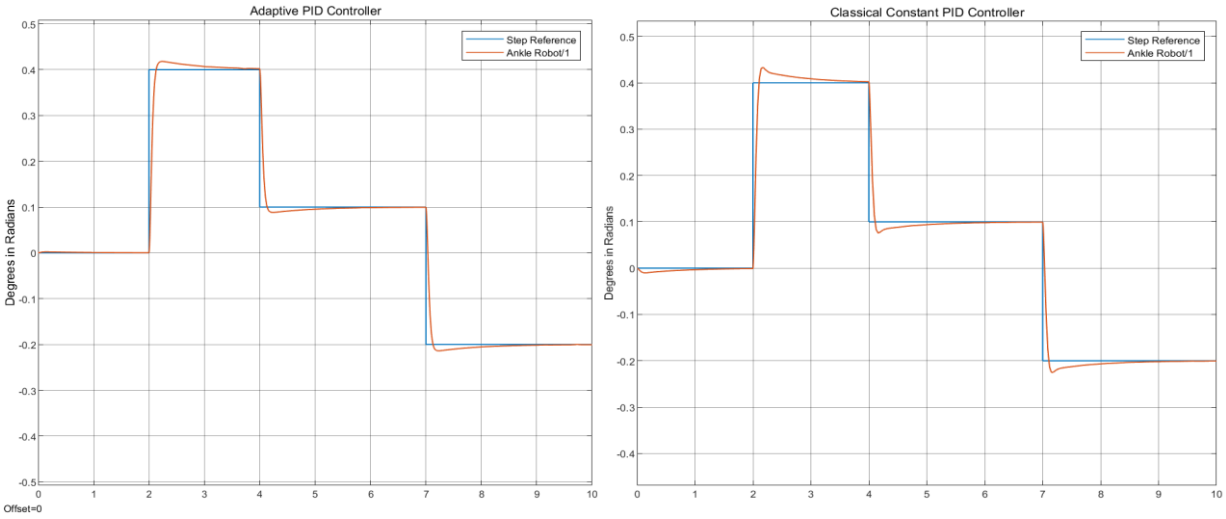


Figure 6.12. Adaptive PID and Constant PID with the trained step function.

The results obtained by the adaptive PID controller and those obtained by the constant PID controller are on par with one another. However, the overshoot of the adaptive controller was only marginally lower than that of the constant PID controller. The constant controller had an overshoot of 0.033 radiance (1.8 degrees) with an RMSE of 0.2426, while the adaptive controller had an overshoot of 0.021 radiance (1.2 degrees) with an RMSE of 0.1152. The adaptive controller also had a lower RMSE. These overshoots are not significant, and the ankle robot is able to get back on track with the reference trajectory it was following. Initially, the constant PID controller was not following the reference signal as it should have. Even though there was some deviation between the two controllers when stepping down and stepping up, the error produced was minimal.

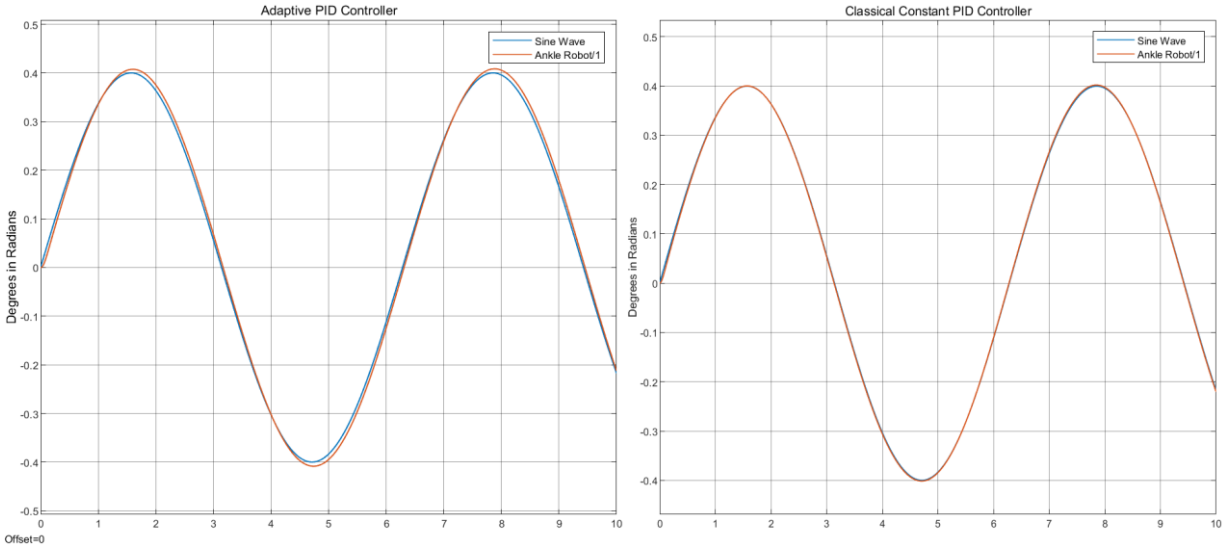


Figure 6.13. Adaptive PID and Constant PID with Sine wave.

In contrast to the trained step signal discussed above, the constant PID controller performed better than the adaptive PID controller when testing for sine wave. The constant PID controller was able to follow the trajectory of the input with an error that was virtually indistinguishable from zero with an RMSE of 0.0025. The adaptive controller had a minimal overshoot of 0.007 radiance, equivalent to 0.4 degrees, and it had an RMSE of 0.01. On both the positive and negative sides of the cycle of the sine wave, there was an overshoot that was precisely the same.

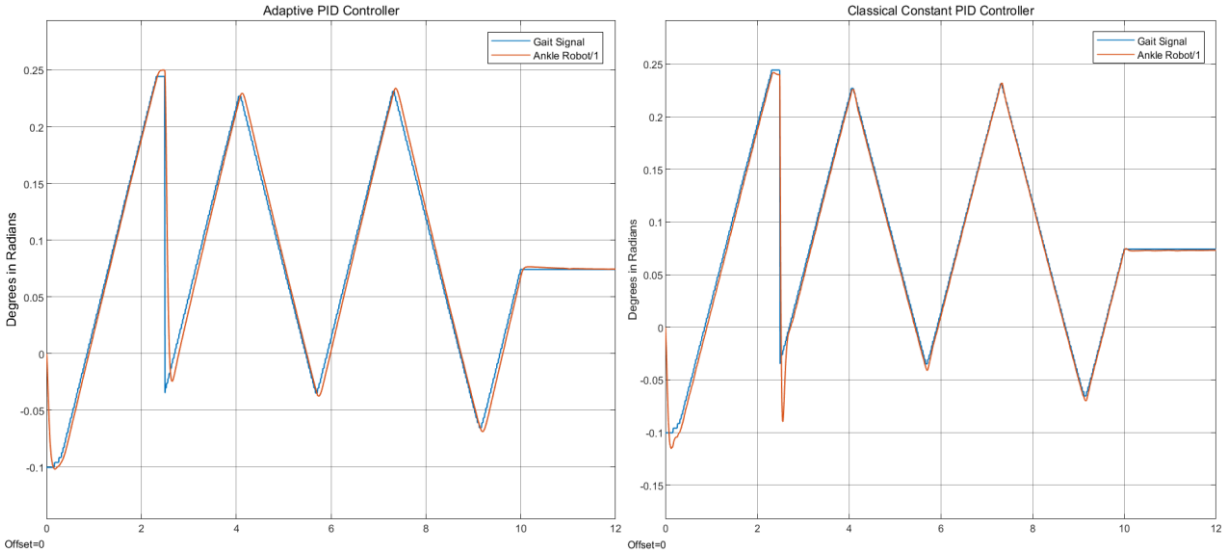


Figure 6.14. Adaptive PID and Constant PID with Gait Signal

Similar to testing with a sine wave signal, the constant PID controller performed marginally better than the adaptive controller. The constant controller's RMSE error was 0.0128, while the adaptive controller's error was 0.0252. The adaptive controller followed with a slight offset to the reference gait signal. The offset was only observed when the ankle position was changed from dorsiflexion to plantarflexion and vice versa. Once the straight path was followed, the controller was able to minimise error. Although the constant controller followed the path with minimal offset, there was an overshoot when the ankle position was changed. The constant controller typically overshoots when changing the path from a straight line to an incline.

Input Signal	RMSE	
	Adaptive Controller	Constant Controller
Step Reference Signal	0.1152	0.2426
Sine Wave	0.01	0.0025
Gait Signal	0.0252	0.0128

Table 6.1. RMSE measurements between Adaptive controller and Constant controller

B. Experiment 2

RL is more suitable for controlling systems with non-linear characteristics. Therefore, we simulate the dynamic environment in the second experiment. To change the environment, we multiply the input torque with a step function that drops its value at a random time. The input torque is multiplied by a step function with one from t = 0 to t = 3, then 0.4 from t = 3 to t = 7, and finally dropped to 0.1 from t = 0.7 to t = 10. The decrease in torque value may resemble motor failure or the inability of the motor to provide the required torque. Sudden changes by the user can also cause changes in the environment. Since the RL agent is not prepared to deal with changes, it is crucial to evaluate the controller's adaptability under dynamic conditions.

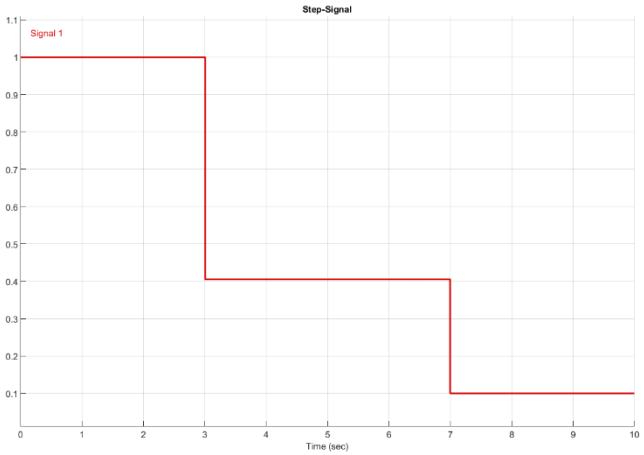


Figure 6.15. Step-signal used to multiply the input torque to change the environment

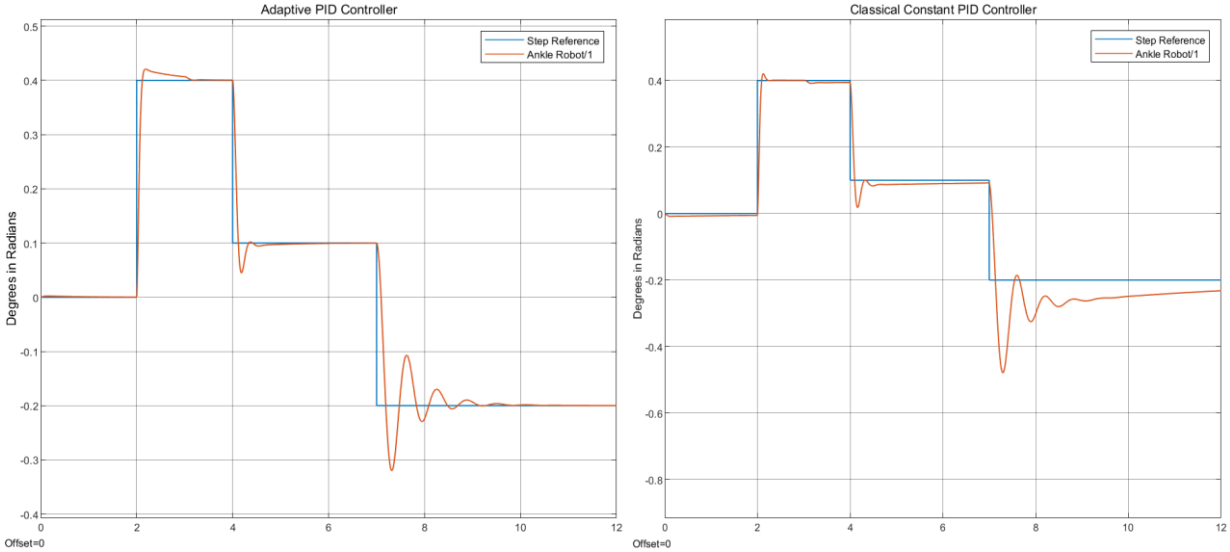


Figure 6.16. Adaptive PID and Constant PID with trained step function under a dynamic environment

When analysing the trained step function in a dynamic environment, we observe an increase in overshoot during the transition to a new position. However, the adaptive controller can quickly return to the desired trajectory while minimising the error relative to the step reference signal. The adaptive controller had a 6-degree overshoot, while the constant controller had a 15-degree overshoot. The adaptive controller's RMSE error was 0.06, while the constant controller was 0.1. The constant PID controller has an offset throughout the environment's changing period. The constant controller requires more time to recover. This is due to the adaptive controller's capacity to update the PID gain over time.

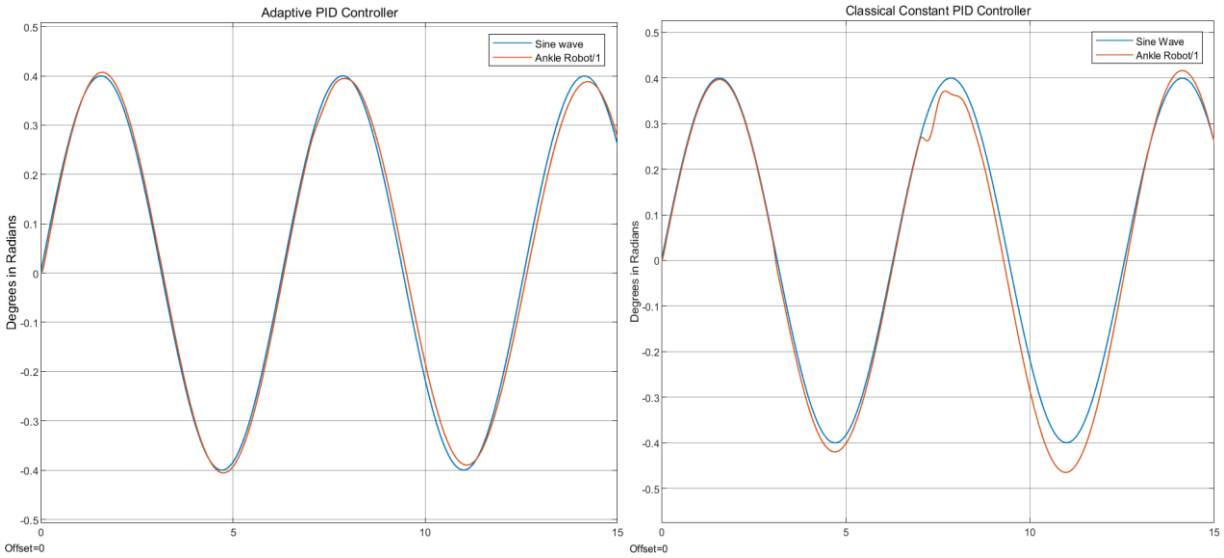


Figure 6.17. Adaptive PID and Constant PID with trained sine signal under a dynamic environment

Like the trained step function, the adaptive controller can follow the sine wave with a small amplitude of error and quickly recover. The robot's trajectory undershot its intended target by a small margin. In contrast, the constant controller can track the trajectory, but it has the largest offset and briefly distorts the robot's path due to the changing environment. An overshoot in the constant controller was also detected during the simulation's entire runtime. The constant controller had an error of 0.03, while the adaptive controller's RMSE was 0.0151.

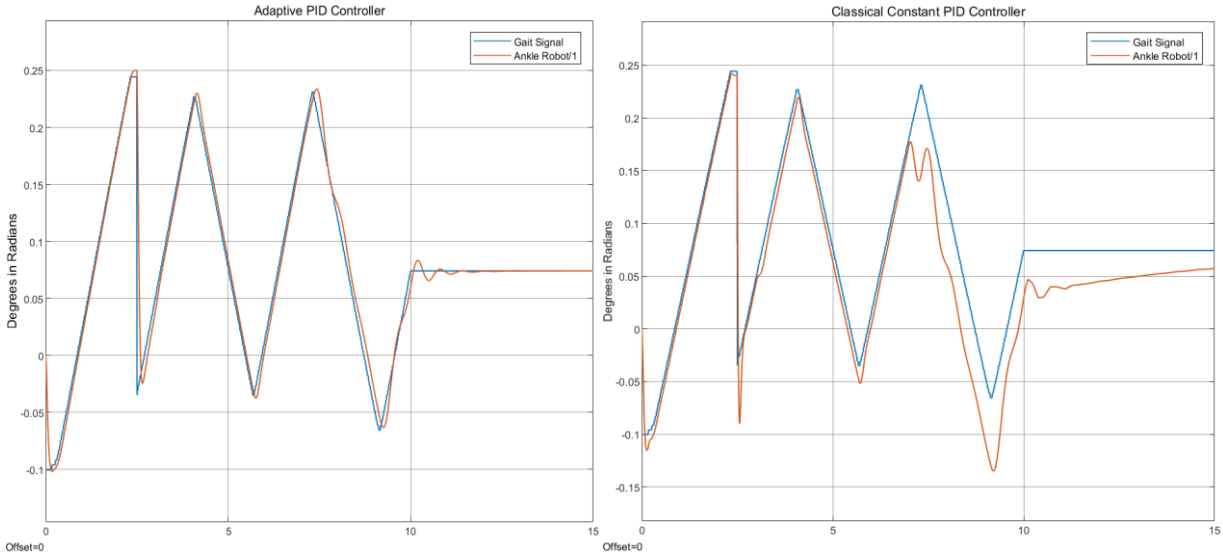


Figure 6.18. Adaptive PID and Constant PID with Gait Signal under a dynamic environment

The constant controller performed poorly when tested for gait signal when the environmental parameters were changed. The robot overshot when the ankle position was changed moving from top to bottom. The robot's trajectory is significantly deformed relative to the reference gait signal. The robot's trajectory also includes a large overshoot, which could harm the user. The constant controller has an RMSE of 0.04 and requires more time to recover. The adaptive controller was capable of tracking the reference gait signal with negligible offsets, and the robot's movements displayed few distortions. This is also due to the gains being updated over time while using the adaptive controller illustrated in Figure 6.19. The adaptive controller had a slight overshoot but rapidly converged with the target trajectory. The RMSE of the adaptive controller was 0.019. The constant PID controller only settles after 30 seconds to the reference path.

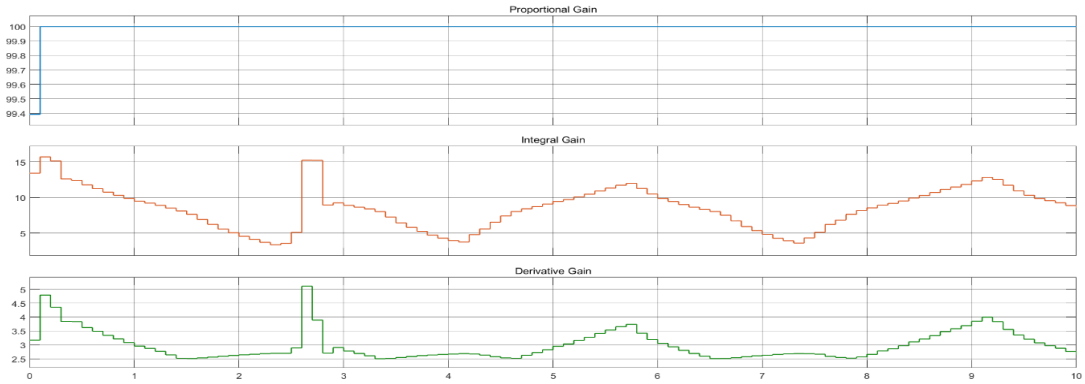


Figure 6.19. Changing gain of the Adaptive PID controller for gait signal under a dynamic environment.

Input Signal	RMSE under dynamic environment	
	Adaptive Controller	Constant Controller
Step Reference Signal	0.06	0.1
Sine Wave	0.0151	0.03
Gait Signal	0.019	0.04

Table 6.2. RMSE measurements between Adaptive controller and Constant controller under dynamic environment

Although the RL agent can follow the reference path with minimal errors, a sudden change in the robot's parameters or the user's ankle motion can cause an overshoot in the input signal that extends the user's ankle beyond its range of motion, which can cause severe injury to the user. To mitigate this effect, human intervention is required; therefore, in this thesis, we employ a smoothing filter to reduce the impact caused by the input signal overshooting. As demonstrated in the experiment 2, under the dynamic environment, the robot overshoots to some degree. To prevent the output of the filter from being applied throughout the motor's working duration, the filter will only be activated if the error relative to the reference signal exceeds 3%. Here, the signal is smoothed for every 150 samples. If we smooth the signal for every sample, it will continue to follow the controller's output path. However, if we smooth the signal for more than 200 samples, the output signal will be deformed. If an overshot occurs, the smoothing filter can also help to slow the motor's movement when the ankle is transitioning between positions. The smoothing filter functions as an external safety feature that is outside of the RL loop that can be fine-tuned to meet the needs of each individual. Figure 6.20 illustrates the application of the smoothing filter when the error of the adaptive controller overshoots the reference signal.

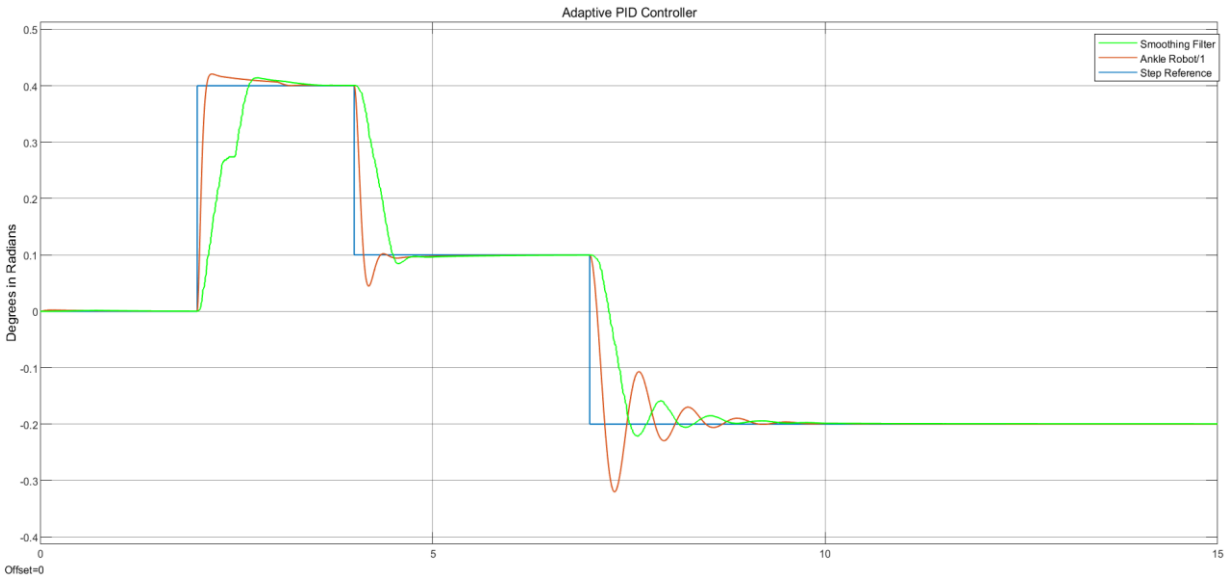


Figure 6.20. Application of smoothing filter on adaptive PID controller under a dynamic environment

6.4 DISCUSSION

The findings show that in a dynamic environment, the adaptive controller works better than the conventional PID controller. Moreover, overshoots, settling, and rise time under the adaptive controller are better controlled. Although the adaptive controller performed marginally poorly in unchanged environments, the errors were acceptable and could be used for rehabilitation. Although the adaptive controller should be more adaptable than the constant PID controller, it should be noted that the adaptive controller has twice the error of the constant PID controller when the environment remains unchanged. This could be the result of a change in the PID gains of the adaptive controller or a decrease in the number of observations required for the agent to take the appropriate actions. The adaptive controller exhibits minimal error when tested with a trained signal in an unaltered environment. This indicates that the error can be reduced with appropriate training. All of the factors mentioned above may explain why the constant PID controller performs better in an unchanged environment. The average acquired PID parameters of the adaptive controller fluctuated dependent on the input signal, however, the output of the constant PID controller did not vary considerably when applying the average gains. As the agent was trained with varying initial values, the adaptive controller was designed to reach the initial

position with minimal overshoot, despite its errors being greater than those of the constant controller.

When an assessment of the changing environment is carried out, the environment shifts after three seconds have passed. The majority of the signal being tested experiences a rapid degradation in tracking. However, in contrast to the constant controller, the adaptive controller is able to make the transition back to the reference path in a shorter amount of time. The adaptive controller can keep frequency and amplitude stable with only a small amount of RMSE, despite signal overshoots and minimal offsets from the reference path. The constant controller is prone to large oscillations and takes a lot of time to reach a stable state. It is also possible to infer that the signal produced by the constant controller may oscillate throughout the duration of the simulation if the controller is tested with other input signals. The simulation mechanics of the ankle robot include a component for gravity, which is the cause of the robot's tendency to oscillate. Since the adaptive controller can keep tracking even when the environment is changed, this controller can also be considered a fault-tolerant controller because of its ability to maintain tracking.

The DDPG is an effective RL algorithm for continuous motions but is extremely sensitive to hyperparameter adjustments. To achieve the best training results, a combination of hyperparameter values was chosen. Changes in exploration parameters and network learning rates led to observable changes. It was observed that slower learning rates produced better results, whereas faster learning rates produced undesirable outcomes. When the standard deviation value which belongs to the exploration parameter was decreased, the reward remained negative throughout all training episodes. Changes in hyperparameters had a significant impact on the RMSE obtained when testing with various input signals. Initially, the RL agent was trained with a sine signal; however, it was observed during testing that the sine wave-trained RL agent performed poorly when tested for other untrained inputs with a very high RMSE value. The robot oscillated throughout the duration of the simulation when the sine-trained agent was tested with a gait signal. An illustration of tracking error by the sine wave trained RL agent is shown in Figure 6.21. This may be due to the agent's limited frequency and amplitude during training, unlike the step function, which had different amplitudes. In this project, the hyperparameters and the selection of the proper training signal were crucial to getting the desired outcome. Rehabilitation is typically a very slow process, so a patient's ankles would be moved very slowly.

When the RL agent was tested for the sine wave at very low frequencies, such as at less than 1 Hz, the controller could follow the sine path without error even in a constantly changing environment. The RMSE was $7.01 * 10^{-6}$ for very slow frequencies.

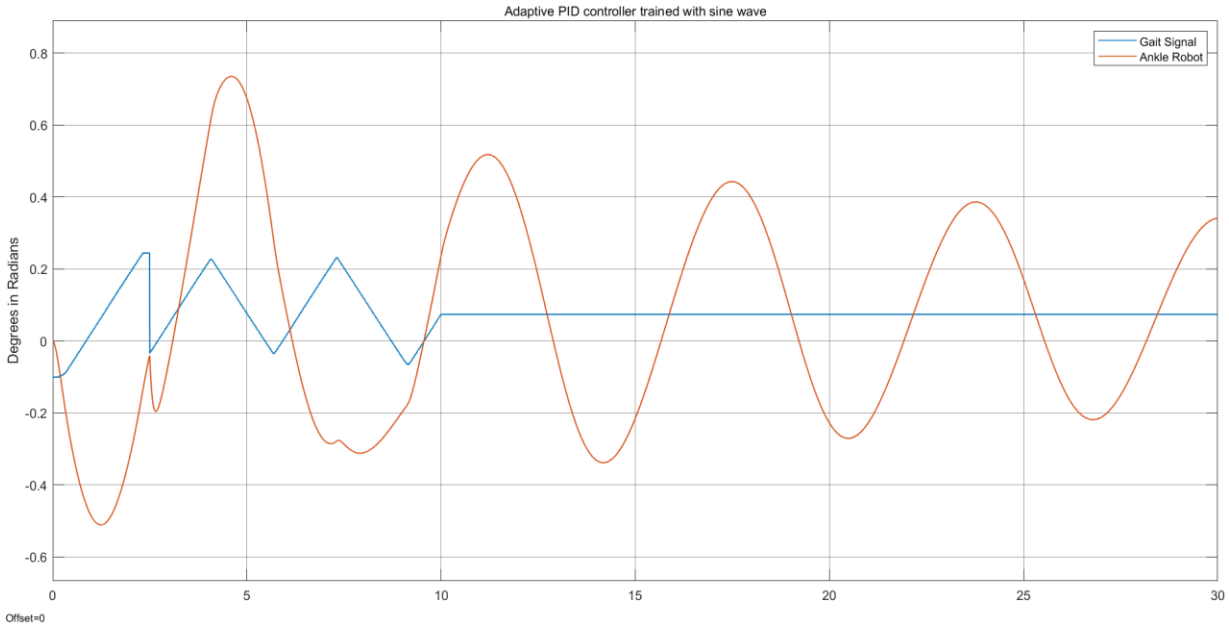


Figure 6.21. Adaptive PID controller trained with a sine wave.

6.5 CONCLUSION

Classical PID controllers remain an indispensable, simple, and effective component for managing linear systems. However, they lack robustness with regard to non-linear systems. Due to the unpredictable nature of the user and the robot's mechanical properties, it is challenging to model human-robot interaction. Changes in muscle activation can also lead to uncertainty while performing rehabilitation exercises. Nonetheless, an adaptive PID controller is able to mitigate problems caused by unforeseen circumstances. The controller has demonstrated its ability to manage untrained inputs in a dynamic environment. The RMSE produced by the adaptive controller is very small and can be justified for use in robotic rehabilitation. Therefore, an adaptive controller can ensure the user's safety when interacting with the robot.

7 CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

To summarise, the ankle rehabilitation system discussed in this thesis is intended to improve human-machine interaction via physiological signals. Physiological data such as EMG from the muscles and information collected from EEG signals from the brain were employed to engage with the robot. EMG signals help the robot to move, even before the movement has occurred whereas, EEG signals may be utilised to allow patients, such as stroke sufferers, to engage with the robot by imagining movement without actually moving their ankle. The thesis is comprised of three modules, each of which is addressed in four chapters. To make the user feel at ease while operating the robot, we begin by creating the mechanical structure for the ankle robot, as described in the first module. The suggested mechanical design may enhance human-robot interaction and expedite rehabilitation. In addition to enhancing safety, the robot's operation has a mechanical brake mechanism that aids therapists in ensuring the user's safety. A separate leg support design may aid the user in completing the activity without experiencing weariness. Chapter 3's recommended design facilitates portability and usability for its consumers. The proposed design can be made adaptable to the requirements of each patient.

Module 2, described in Chapters 4 and 5, provides a mechanism for rehabilitation utilising physiological signals. Chapter 4 categorises EMG signals for five distinct ankle motions using cubic SVM. A combination of five novel features is presented to achieve a classification accuracy of 98.9%. The research also demonstrates that choosing a window size of 130ms ensures that the robot

can react immediately after identifying the planned movement. By combining IMU and EMG, it is possible to examine the orientation of the ankle in relation to the intended movement. Here signal to noise ratio is used to assess the condition of the signal produced by a EMG sensor. In an event where a single EMG sensor produces noises above the set threshold, the control of the robot would be switched to being controlled by the IMU sensor. This increases the user's safety since IMU serves as the ground truth. In the following session, using EMG to estimate the muscle forces during plantarflexion and dorsiflexion of the ankle is studied. A further layer of assurance was provided by real-time classification of user intent using multi-class SVM, reducing the failure risk for the selection of the correct equation and, in turn, reducing processing and execution time. Because of the nonlinear nature of the relationship between EMG and joint torque, this particular mathematical model was used. Accurate solutions to the nonlinear problem may be found using PSO with a fitness function. It was observed that joint torque could be consistently predicted by combining PSO with SVM classification. It lowers the number of sensors used in rehabilitation, and the proposed technique may provide a simpler and more effective solution.

In Chapter 5, we classify ankle movement using an EEG signal. The ankle patterns are classified using motor imagery, in which the user visualises a particular movement. We classified the data by combining five-time domain characteristics with two frequency domain features. A unique approach was developed in which the weights of ICA were utilised to minimise the number of EEG channels while maintaining the same level of accuracy as utilising all EEG channels. In addition, to make the system computationally efficient, the feature dimensions were reduced using the MRMR approach. The suggested technique indicated that the decrease of features and EEG channel had no or very little effect on the accuracy of the classifier. The decreased number of EEG channels makes it very user-friendly. Utilising physiological signals gives the user more control and the ability to direct the abnormal movement of the ankle, thereby aiding a quicker recovery. As the channel numbers are reduced, it also improves computational efficiency

In module 3, a control system that allowed the robot to adapt to the patient's movements was developed. To accomplish this objective, a reinforcement learning agent was utilised to tune the PID controller's parameters. A deep deterministic policy gradient method is employed to train the agent. The agent was trained with a random signal while observing 12 different parameters of the

agent. The controller was evaluated with signals travelling several pathways, including the calculated joint angle from EMG. The controller performed slightly lower than the constant PID controller under an unchanged environment. The error observed for adaptive PID controllers was negligible in an unchanged environment. However, under the dynamic environment, the adaptive controller outperformed the constant PID controller proving to be superior even though the DDPG agent was not trained for these conditions. The outcome illustrates that the robot was able to adjust the PID controller such that it could follow any path other than the one it was trained for.

We could analyse and evaluate every motion of the user by using EMG and EEG data. Giving the individual more control over their own rehabilitation may improve its effectiveness. Furthermore, by enhancing human-machine interaction, the workload of physiotherapists' can be reduced. While this thesis primarily focuses on ankle rehabilitation, the methods discussed may also be employed for upper limb rehabilitation. The methods offered in this thesis have also been tested on an upper limb exoskeleton at the UTS, with similar success in determining human intent. There is a huge market opportunity for telehealth through which practitioners can provide home rehabilitation. The ability to undergo therapy at home, in the present Covid 19 pandemic, is a significant advantage for all its users and the community.

7.2 FUTURE WORK

Despite the fact that an ankle rehabilitation robot has been developed, its effects on individuals with ankle injuries have not yet been investigated. Future studies in this field would first, concentrate on developing a mechanical design for a robot with three degrees of freedom and the ability to enable the four basic ankle motions of dorsiflexion, plantarflexion, inversion, and eversion.

Secondly, the presently used algorithms categorise the data based only on pattern recognition, but research should also concentrate on demonstrating how physiological data may be utilised to diagnose the kind of ankle injury. This would allow us to monitor the physiological alterations caused by various traumas. This association might provide patients with confidence in the robot-assisted rehabilitation routines. Regarding EEG channel reduction, we must also consider utilising a universal strategy applicable to a larger population.

The torque estimate and adaptive PID controller in this thesis are restricted to ankle dorsiflexion and plantarflexion due to hardware limitations. Therefore, research is being undertaken on the estimation of forces during ankle motions other than dorsiflexion and plantarflexion. Likewise, the ankle model used to train the adaptive controller must be updated to account for all ankle motions that might occur in three-dimensional space. Other algorithms, such as transfer learning, should also be used to examine how patients adapt to the machine, since both the machine and the user may learn from one another, therefore minimising the amount of effort required to operate the robot.

Finally, to validate the proposed methodologies, a clinical study is necessary to test them using a large population. Despite the fact that the present pandemic has made it more difficult to get clinical data, Amazon Web Services (AWS) is interested in using this technology to promote telehealth.

8 REFERENCE

1. MedicineNews, N. *Ankle sprains: 10 things you should know*. 2022 3 September 2022]; Available from: https://www.nps.org.au/consumers/ankle-sprains-10-things-you-should-know.
2. Mattacola, C.G. and M.K. Dwyer, *Rehabilitation of the Ankle After Acute Sprain or Chronic Instability.* J Athl Train, 2002. **37**(4): p. 413-429.
3. Alvarez-Perez, M.G., M.A. Garcia-Murillo, and J.J. Cervantes-Sanchez, *Robot-assisted ankle rehabilitation: a review.* Disabil Rehabil Assist Technol, 2020. **15**(4): p. 394-408.
4. Valadao, C.T., et al., *Robotics as a Tool for Physiotherapy and Rehabilitation Sessions.* IFAC-PapersOnLine, 2015. **48**(19): p. 148-153.
5. Blaya, J.A. and H. Herr, *Adaptive control of a variable-impedance ankle-foot orthosis to assist drop-foot gait.* IEEE Trans Neural Syst Rehabil Eng, 2004. **12**(1): p. 24-31.
6. Zhang, M., T.C. Davies, and S. Xie, *Effectiveness of robot-assisted therapy on ankle rehabilitation--a systematic review.* J Neuroeng Rehabil, 2013. **10**: p. 30.
7. Bingjing, G., et al., *Human–robot interactive control based on reinforcement learning for gait rehabilitation training robot.* International Journal of Advanced Robotic Systems, 2019. **16**(2).
8. Buchanan, T.S., et al., *Neuromusculoskeletal modeling: estimation of muscle forces and joint moments and movements from measurements of neural command.* J Appl Biomech, 2004. **20**(4): p. 367-95.
9. Lloyd, D.G. and T.F. Besier, *An EMG-driven musculoskeletal model to estimate muscle forces and knee joint moments in vivo.* Journal of Biomechanics, 2003. **36**(6): p. 765-776.
10. Lobo-Prat, J., et al., *Comparison between sEMG and force as control interfaces to support planar arm movements in adults with Duchenne: a feasibility study.* J Neuroeng Rehabil, 2017. **14**(1): p. 73.

11. Shin, D., J. Kim, and Y. Koike, *A myokinetic arm model for estimating joint torque and stiffness from EMG signals during maintained posture.* J Neurophysiol, 2009. **101**(1): p. 387-401.
12. Dongbao, S., et al., *Design of a wearable upper-limb exoskeleton for activities assistance of daily living*, in *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. 2017. p. 845-850.
13. Krebs, H.I., et al., *Rehabilitation Robotics: Performance-Based Progressive Robot-Assisted Therapy.* Autonomous Robots, 2003. **15**(1): p. 7-20.
14. Schmidt, H., et al., *HapticWalker---a novel haptic foot device.* ACM Transactions on Applied Perception, 2005. **2**(2): p. 166-180.
15. Krebs, H.I., et al., *Robot-aided neurorehabilitation.* IEEE Trans Rehabil Eng, 1998. **6**(1): p. 75-87.
16. Gandolfi, M., et al., *Feasibility and safety of early lower limb robot-assisted training in sub-acute stroke patients: a pilot study.* Eur J Phys Rehabil Med, 2017. **53**(6): p. 870-882.
17. Ao, D., R. Song, and J. Gao, *Movement Performance of Human-Robot Cooperation Control Based on EMG-Driven Hill-Type and Proportional Models for an Ankle Power-Assist Exoskeleton Robot.* IEEE Trans Neural Syst Rehabil Eng, 2017. **25**(8): p. 1125-1134.
18. Ren, Y., et al., *Developing a Wearable Ankle Rehabilitation Robotic Device for in-Bed Acute Stroke Rehabilitation.* IEEE Trans Neural Syst Rehabil Eng, 2017. **25**(6): p. 589-596.
19. Riener, R., et al., *Patient-cooperative strategies for robot-aided treadmill training: first experimental results.* IEEE Trans Neural Syst Rehabil Eng, 2005. **13**(3): p. 380-94.
20. Patton, J.L., et al., *Evaluation of robotic training forces that either enhance or reduce error in chronic hemiparetic stroke survivors.* Exp Brain Res, 2006. **168**(3): p. 368-83.
21. Nazmi, N., et al., *A Review of Classification Techniques of EMG Signals during Isotonic and Isometric Contractions.* Sensors (Basel), 2016. **16**(8).
22. Al-Quraishi, M.S., et al., *Classification of ankle joint movements based on surface electromyography signals for rehabilitation robot applications.* Med Biol Eng Comput, 2017. **55**(5): p. 747-758.
23. Nurhanim, K., et al., *Joint Torque Estimation Model of Surface Electromyography(sEMG) Based on Swarm Intelligence Algorithm for Robotic Assistive Device.* Procedia Computer Science, 2014. **42**: p. 175-182.
24. Merletti, R. and P. Parker, *Electromyography*. Physiology, Engineering, and Noninvasive Applications. 2004.
25. Jacks, A.S. and N.R. Miller, *Spontaneous retinal venous pulsation: aetiology and significance.* J Neurol Neurosurg Psychiatry, 2003. **74**(1): p. 7-9.
26. Jasper, H.H., *Report of the committee on methods of clinical examination in electroencephalography.* Electroencephalography and Clinical Neurophysiology, 1958. **10**(2): p. 370-375.

27. Wang, J., et al., *Enhanced Gamma Activity and Cross-Frequency Interaction of Resting-State Electroencephalographic Oscillations in Patients with Alzheimer's Disease.* Front Aging Neurosci, 2017. **9**: p. 243.
28. Brockett, C.L. and G.J. Chapman, *Biomechanics of the ankle.* Orthop Trauma, 2016. **30**(3): p. 232-238.
29. Zwipp, H. and T. Randt, *Ankle joint biomechanics.* Foot and Ankle Surgery, 1994. **1**(1): p. 21-27.
30. Coetzee, J.C. and M.D. Castro, *Accurate measurement of ankle range of motion after total ankle arthroplasty.* Clin Orthop Relat Res, 2004(424): p. 27-31.
31. Wang, C., et al., *Design and Kinematical Performance Analysis of a 3-RUS/RRR Redundantly Actuated Parallel Mechanism for Ankle Rehabilitation.* Journal of Mechanisms and Robotics, 2013. **5**(4).
32. Hermens, H.J., et al., *European recommendations for surface electromyography.* Roessingh Research and Development, 1999. **8**: p. 13-54.
33. Mademli, L., et al., *Effect of ankle joint position and electrode placement on the estimation of the antagonistic moment during maximal plantarflexion.* J Electromyogr Kinesiol, 2004. **14**(5): p. 591-7.
34. Li, Z., et al., *Compliant training control of ankle joint by exoskeleton with human EMG-torque interface.* Assembly Automation, 2017. **37**(3): p. 349-355.
35. Mamikoglu, U., et al., *Electromyography based joint angle estimation and control of a robotic leg*, in *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. 2016. p. 182-187.
36. Zhou, Z., et al., *Robot-Assisted Rehabilitation of Ankle Plantar Flexors Spasticity: A 3-Month Study with Proprioceptive Neuromuscular Facilitation.* Front Neurorobot, 2016. **10**: p. 16.
37. Englehart, K., et al., *Classification of the myoelectric signal using time-frequency based representations.* Medical Engineering & Physics, 1999. **21**(6-7): p. 431-438.
38. Kyeong, S., et al., *Recognition of walking environments and gait period by surface electromyography.* Frontiers of Information Technology & Electronic Engineering, 2019. **20**(3): p. 342-352.
39. Archer, C.M., et al., *Activity classification in users of ankle foot orthoses.* Gait Posture, 2014. **39**(1): p. 111-7.
40. Al-Mulla, M.R., F. Sepulveda, and M. Colley, *A Review of Non-Invasive Techniques to Detect and Predict Localised Muscle Fatigue.* Sensors, 2011. **11**(4): p. 3545-3594.
41. Alhagry, S., A. Aly, and R. A, *Emotion Recognition based on EEG using LSTM Recurrent Neural Network.* International Journal of Advanced Computer Science and Applications, 2017. **8**(10).
42. Xu, J., et al., *Recognition of EEG Signal Motor Imagery Intention Based on Deep Multi-View Feature Learning.* Sensors (Basel), 2020. **20**(12).

43. Zhou, W., et al., *Epileptic Seizure Detection Using Lacunarity and Bayesian Linear Discriminant Analysis in Intracranial EEG.* IEEE Trans Biomed Eng, 2013. **60**(12): p. 3375-81.
44. Guo, Y. and A. Kareem, *Non-stationary frequency domain system identification using time–frequency representations.* Mechanical Systems and Signal Processing, 2016. **72-73**: p. 712-726.
45. Bishop, C.M., *Neural Networks for Pattern Recognition*. 1995: Oxford University Press Inc.
46. Ding, C. and H. Peng, *Minimum redundancy feature selection from microarray gene expression data.* J Bioinform Comput Biol, 2005. **3**(2): p. 185-205.
47. Peng, H., F. Long, and C. Ding, *Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy.* IEEE Trans Pattern Anal Mach Intell, 2005. **27**(8): p. 1226-38.
48. Goldberger, J., et al., *Neighbourhood components analysis.* Advances in neural information processing systems, 2004. **17**.
49. Kira, K. and L.A. Rendell. *The feature selection problem: Traditional methods and a new algorithm*. in *Aaai*. 1992.
50. Pearson, K., *X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling.* The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 1900. **50**(302): p. 157-175.
51. Nasution, M.Z.F., O.S. Sitompul, and M. Ramli, *PCA based feature reduction to improve the accuracy of decision tree c4.5 classification.* Journal of Physics: Conference Series, 2018. **978**(1): p. 012058.
52. Nogales, R. and M.E. Benalcázar, *Analysis of Feature Selection Methods for Hand Gestures Recognition Based on Machine Learning and the Leap Motion Controller.* SSRN Electronic Journal, 2022.
53. Kaplan, E., et al., *Novel nested patch-based feature extraction model for automated Parkinson's Disease symptom classification using MRI images.* Comput Methods Programs Biomed, 2022. **224**: p. 107030.
54. Cortes, C. and V. Vapnik, *Support-vector networks.* Machine Learning, 1995. **20**(3): p. 273-297.
55. Khokhar, Z.O., Z.G. Xiao, and C. Menon, *Surface EMG pattern recognition for real-time control of a wrist exoskeleton.* Biomed Eng Online, 2010. **9**: p. 41.
56. Alkan, A. and M. Günay, *Identification of EMG signals using discriminant analysis and SVM classifier.* Expert Systems with Applications, 2012. **39**(1): p. 44-47.
57. Gao, S., et al., *A Smart Terrain Identification Technique Based on Electromyography, Ground Reaction Force, and Machine Learning for Lower Limb Rehabilitation.* Applied Sciences, 2020. **10**(8).

58. Al-Qazzaz, N.K., et al., *Discrimination of stroke-related mild cognitive impairment and vascular dementia using EEG signal analysis.* Med Biol Eng Comput, 2018. **56**(1): p. 137-157.
59. Li, X., et al., *Classification of EEG signals using a multiple kernel learning support vector machine.* Sensors (Basel), 2014. **14**(7): p. 12784-802.
60. Phinyomark, A., et al., *Application of Linear Discriminant Analysis in Dimensionality Reduction for Hand Motion Classification.* Measurement Science Review, 2012. **12**(3).
61. Sakai, M., N. Kitaoka, and K. Takeda, *Acoustic Feature Transformation Based on Discriminant Analysis Preserving Local Structure for Speech Recognition.* IEICE Transactions on Information and Systems, 2010. **E93-D**(5): p. 1244-1252.
62. Schafer, K.C., et al., *Real time analysis of brain tissue by direct combination of ultrasonic surgical aspiration and sonic spray mass spectrometry.* Anal Chem, 2011. **83**(20): p. 7729-35.
63. Englehart, K. and B. Hudgins, *A robust, real-time control scheme for multifunction myoelectric control.* IEEE Trans Biomed Eng, 2003. **50**(7): p. 848-54.
64. Fu, R., et al., *Improvement Motor Imagery EEG Classification Based on Regularized Linear Discriminant Analysis.* J Med Syst, 2019. **43**(6): p. 169.
65. Cover, T. and P. Hart, *Nearest neighbor pattern classification.* IEEE Transactions on Information Theory, 1967. **13**(1): p. 21-27.
66. Kim, K.S., et al., *Comparison of k-nearest neighbor, quadratic discriminant and linear discriminant analysis in classification of electromyogram signals based on the wrist-motion directions.* Current Applied Physics, 2011. **11**(3): p. 740-745.
67. McCulloch, W.S. and W. Pitts, *A logical calculus of the ideas immanent in nervous activity.* The Bulletin of Mathematical Biophysics, 1943. **5**(4): p. 115-133.
68. Subasi, A., M. Yilmaz, and H.R. Ozcalik, *Classification of EMG signals using wavelet neural network.* J Neurosci Methods, 2006. **156**(1-2): p. 360-7.
69. Choi, R.Y., et al., *Introduction to Machine Learning, Neural Networks, and Deep Learning.* Transl Vis Sci Technol, 2020. **9**(2): p. 14.
70. Islam, M. and E.T. Hsiao-Wecksler, *Detection of Gait Modes Using an Artificial Neural Network during Walking with a Powered Ankle-Foot Orthosis.* J Biophys, 2016. **2016**: p. 7984157.
71. Lorentzen, J., et al., *Feedforward neural control of toe walking in humans.* J Physiol, 2018. **596**(11): p. 2159-2172.
72. Zhou, H.Y., et al., *Estimating the Ankle Angle Induced by FES via the Neural Network-Based Hammerstein Model.* IEEE Access, 2019. **7**: p. 141277-141286.
73. Saga, N., et al., *Prototype of an Ankle Neurorehabilitation System with Heuristic BCI Using Simplified Fuzzy Reasoning.* Applied Sciences, 2019. **9**(12).

74. Rumelhart, D.E., G.E. Hinton, and R.J. Williams, *Learning representations by back-propagating errors.* Nature, 1986. **323**(6088): p. 533-536.
75. Petrosian, A.A., et al., *Recurrent neural network-based approach for early recognition of Alzheimer's disease in EEG.* Clinical Neurophysiology, 2001. **112**(8): p. 1378-1387.
76. Li, Z., et al., *Muscle Fatigue Tracking with Evoked EMG via Recurrent Neural Network: Toward Personalized Neuroprosthetics.* IEEE Computational Intelligence Magazine, 2014. **9**(2): p. 38-46.
77. Zhai, X., et al., *Self-Recalibrating Surface EMG Pattern Recognition for Neuroprosthesis Control Based on Convolutional Neural Network.* Front Neurosci, 2017. **11**: p. 379.
78. Xia, P., J. Hu, and Y. Peng, *EMG-Based Estimation of Limb Movement Using Deep Learning With Recurrent Convolutional Neural Networks.* Artif Organs, 2018. **42**(5): p. E67-E77.
79. Thomas, J., et al., *EEG CLassification Via Convolutional Neural Network-Based Interictal Epileptiform Event Detection.* Conf Proc IEEE Eng Med Biol Soc, 2018. **2018**: p. 3148-3151.
80. Tamei, T. and T. Shibata, *Fast Reinforcement Learning for Three-Dimensional Kinetic Human–Robot Cooperation with an EMG-to-Activation Model.* Advanced Robotics, 2012. **25**(5): p. 563-580.
81. Kukker, A., et al., *Neural reinforcement learning classifier for elbow, finger and hand movements.* Journal of Intelligent & Fuzzy Systems, 2018. **35**(5): p. 5111-5121.
82. Kennedy, J. and R. Eberhart, *Particle swarm optimization*, in *Proceedings of ICNN'95 - International Conference on Neural Networks*. 1995. p. 1942-1948.
83. Sigalingging, X.K., et al., *Electromyography-based gesture recognition for quadriplegic users using hidden Markov model with improved particle swarm optimization.* International Journal of Distributed Sensor Networks, 2019. **15**(7).
84. Whitley, D., *A genetic algorithm tutorial.* Statistics and Computing, 1994. **4**(2).
85. Fischer, P., et al., *Genetic algorithm based optimization for EMG pattern recognition system.* IFAC Proceedings Volumes, 2009. **42**(13): p. 53-58.
86. Buongiorno, D., et al., *A Linear Approach to Optimize an EMG-Driven Neuromusculoskeletal Model for Movement Intention Detection in Myo-Control: A Case Study on Shoulder and Elbow Joints.* Front Neurorobot, 2018. **12**: p. 74.
87. Van Laarhoven, P.J.M. and E.H.L. Aarts, *Simulated annealing*, in *Simulated Annealing: Theory and Applications*. 1987, Springer Netherlands: Dordrecht. p. 7-15.
88. Parasuraman, S., A.W. Oyong, and V.L. Jauw, *Robot Assisted Stroke Rehabilitation: Joint Torque/Force Conversion from Emg Using Sa Process.* Journal of Mechanics in Medicine and Biology, 2011. **11**(03): p. 691-704.

89. Barrett, R.S., T.F. Besier, and D.G. Lloyd, *Individual muscle contributions to the swing phase of gait: An EMG-based forward dynamics modelling approach.* Simulation Modelling Practice and Theory, 2007. **15**(9): p. 1146-1155.
90. Keemink, A.Q.L., H. van der Kooij, and A.H.A. Stienen, *Admittance control for physical human–robot interaction.* The International Journal of Robotics Research, 2018. **37**(11): p. 1421-1444.
91. Zhuang, Y., et al., *Voluntary Control of an Ankle Joint Exoskeleton by Able-Bodied Individuals and Stroke Survivors using EMG-Based Admittance Control Scheme.* IEEE Trans Biomed Eng, 2020. **PP**.
92. Meng, W., et al., *Recent development of mechanisms and control strategies for robot-assisted lower limb rehabilitation.* Mechatronics, 2015. **31**: p. 132-145.
93. Nizamis, K., et al., *Transferrable Expertise From Bionic Arms to Robotic Exoskeletons: Perspectives for Stroke and Duchenne Muscular Dystrophy.* IEEE Transactions on Medical Robotics and Bionics, 2019. **1**(2): p. 88-96.
94. Liu, Q., et al., *Development of a New Robotic Ankle Rehabilitation Platform for Hemiplegic Patients after Stroke.* J Healthc Eng, 2018. **2018**: p. 3867243.
95. Kiam Heong, A., G. Chong, and L. Yun, *PID control system analysis, design, and technology.* IEEE Transactions on Control Systems Technology, 2005. **13**(4): p. 559-576.
96. Zhuang, Y., et al., *Admittance Control Based on EMG-Driven Musculoskeletal Model Improves the Human–Robot Synchronization.* IEEE Transactions on Industrial Informatics, 2019. **15**(2): p. 1211-1218.
97. Kundu, A.S., et al., *Hand Gesture Recognition Based Omnidirectional Wheelchair Control Using IMU and EMG Sensors.* Journal of Intelligent & Robotic Systems, 2017. **91**(3-4): p. 529-541.
98. Martinez-Hernandez, U., I. Mahmood, and A.A. Dehghani-Sanij, *Simultaneous Bayesian Recognition of Locomotion and Gait Phases With Wearable Sensors.* IEEE Sensors Journal, 2018. **18**(3): p. 1282-1290.
99. Stival, F., et al., *Toward a Better Robotic Hand Prosthesis Control: Using EMG and IMU Features for a Subject Independent Multi Joint Regression Model*, in *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*. 2018. p. 185-192.
100. Huang, Y., et al., *Joint torque estimation for the human arm from sEMG using backpropagation neural networks and autoencoders.* Biomedical Signal Processing and Control, 2020. **62**.
101. Jamwal, P.K., S. Xie, and K.C. Aw, *Kinematic design optimization of a parallel ankle rehabilitation robot using modified genetic algorithm.* Robotics and Autonomous Systems, 2009. **57**(10): p. 1018-1027.
102. Lunenburger, L., et al., *Clinical Assessments Performed During Robotic Rehabilitation by the Gait Training Robot Lokomat*, in *9th International*

Conference on Rehabilitation Robotics, 2005. ICORR 2005. 2005. p. 345-348.
103. Girone, M., et al., *Orthopedic rehabilitation using the "Rutgers ankle" interface*, in *Studies in health technology and informatics*. 2000, IOS Press Ebooks. p. 89-95.
104. Deutsch, J.E., J.A. Lewis, and G. Burdea, *Technical and patient performance using a virtual reality-integrated telerehabilitation system: preliminary finding.* IEEE Trans Neural Syst Rehabil Eng, 2007. **15**(1): p. 30-5.
105. Roy, A., et al., *Robot-Aided Neurorehabilitation: A Novel Robot for Ankle Rehabilitation.* IEEE Transactions on Robotics, 2009. **25**(3): p. 569-582.
106. Selles, R.W., et al., *Feedback-controlled and programmed stretching of the ankle plantarflexors and dorsiflexors in stroke: effects of a 4-week intervention program.* Arch Phys Med Rehabil, 2005. **86**(12): p. 2330-6.
107. Homma, K. and M. Usuba, *Development of Ankle Dorsiflexion/Plantarflexion Exercise Device with Passive Mechanical Joint*, in *2007 IEEE 10th International Conference on Rehabilitation Robotics*. 2007. p. 292-297.
108. Zhou, Z., et al., *A proprioceptive neuromuscular facilitation integrated robotic ankle–foot system for post stroke rehabilitation.* Robotics and Autonomous Systems, 2015. **73**: p. 111-122.
109. Yao, S., et al., *Adaptive Admittance Control for an Ankle Exoskeleton Using an EMG-Driven Musculoskeletal Model.* Front Neurorobot, 2018. **12**: p. 16.
110. Federici, S. and M. Scherer, *Assistive Technology Assessment Handbook, Second Edition*. 2 ed. 2017, Boca Raton: CRC Press. 512.
111. Langer, A., et al., *Trust in socially assistive robots: Considerations for use in rehabilitation.* Neurosci Biobehav Rev, 2019. **104**: p. 231-239.
112. Chatterton, H.J., V.M. Pomeroy, and J. Gratton, *Positioning for stroke patients: a survey of physiotherapists' aims and practices.* Disabil Rehabil, 2001. **23**(10): p. 413-21.
113. Colombo, R., et al., *Design strategies to improve patient motivation during robot-aided rehabilitation.* J Neuroeng Rehabil, 2007. **4**: p. 3.
114. Physio, A. *Kineter Breva*. n.d. [cited 2021 21st Novmeber]; Available from: https://www.antisel-physio.gr/breva/.
115. Fleischer, C. and G. Hommel, *Torque control of an exoskeletal knee with EMG signals.* Vdi Berichte, 2006. **1956**: p. 79.
116. Lloyd, D.G. and T.S. Buchanan, *A model of load sharing between muscles and soft tissues at the human knee during static tasks.* J Biomech Eng, 1996. **118**(3): p. 367-76.
117. Borst, J., et al., *Muscle parameters for musculoskeletal modelling of the human neck.* Clin Biomech (Bristol, Avon), 2011. **26**(4): p. 343-351.
118. Veeger, H.E.J., et al., *Parameters for modeling the upper extremity.* Journal of Biomechanics, 1997. **30**(6): p. 647-652.

119. Oyong, A.W., S. Parasuraman, and V.L. Jauw, *Robot assisted stroke rehabilitation: Estimation of muscle force/joint torque from EMG using GA*, in *2010 IEEE EMBS Conference on Biomedical Engineering and Sciences (IECBES)*. 2010. p. 341-347.
120. Asghari Oskoei, M. and H. Hu, *Myoelectric control systems—A survey.* Biomedical Signal Processing and Control, 2007. **2**(4): p. 275-294.
121. Zajac, F.E., *Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control.* Crit Rev Biomed Eng, 1989. **17**(4): p. 359-411.
122. Guimaraes, A.C., et al., *The EMG-force relationship of the cat soleus muscle and its association with contractile conditions during locomotion.* J Exp Biol, 1995. **198**(Pt 4): p. 975-87.
123. Herzog, W., *History dependence of force production in skeletal muscle: a proposal for mechanisms.* Journal of Electromyography and Kinesiology, 1998. **8**(2): p. 111-117.
124. Milner-Brown, H.S., R.B. Stein, and R. Yemm, *Changes in firing rate of human motor units during linearly changing voluntary contractions.* J Physiol, 1973. **230**(2): p. 371-90.
125. Rabiner, L.R. and B. Gold, *Theory and application of digital signal processing.* Englewood Cliffs: Prentice-Hall, 1975.
126. Metral, S. and G. Cassar, *Relationship between force and integrated EMG activity during voluntary isometric anisotonic contraction.* European Journal of Applied Physiology and Occupational Physiology, 1981. **46**(2): p. 185-198.
127. Ullah, K. and K. Jung-Hoon, *A mathematical model for mapping EMG signal to joint torque for the human elbow joint using nonlinear regression*, in *2009 4th International Conference on Autonomous Robots and Agents*. 2009. p. 103-108.
128. Kennedy, J. and R. Eberhart, *Particle swarm Optimization*. Vol. 4. 1995. 1942-1948 vol.4.
129. Shamsuddin, S.M. and D.P. Rini, *Particle Swarm Optimization: Technique, System and Challenges.* International Journal of Applied Information Systems, 2011. **1**(1): p. 33-45.
130. Zhao, X., H. Sun, and D. Ye. *Ankle rehabilitation robot control based on biological signals*. in *2017 29th Chinese Control And Decision Conference (CCDC)*. 2017.
131. Phinyomark, A., P. Phukpattaranont, and C. Limsakul, *Feature reduction and selection for EMG signal classification.* Expert Systems with Applications, 2012. **39**(8): p. 7420-7431.
132. Hooda, N. and N. Kumar, *Optimal Channel-set and Feature-set Assessment for Foot Movement Based EMG Pattern Recognition.* Applied Artificial Intelligence, 2021. **35**(15): p. 1685-1707.

133. Mejia Tobar, A., et al., *Decoding of Ankle Flexion and Extension from Cortical Current Sources Estimated from Non-invasive Brain Activity Recording Methods.* Front Neurosci, 2017. **11**: p. 733.
134. Li, M., et al., *Recognition Method of Limb Motor Imagery EEG Signals Based on Integrated Back-propagation Neural Network.* Open Biomed Eng J, 2015. **9**: p. 83-91.
135. Alonso-Valerdi, L.M., R.A. Salido-Ruiz, and R.A. Ramirez-Mendoza, *Motor imagery based brain-computer interfaces: An emerging technology to rehabilitate motor deficits.* Neuropsychologia, 2015. **79**(Pt B): p. 354-63.
136. Makeig, S., et al., *Mining event-related brain dynamics.* Trends Cogn Sci, 2004. **8**(5): p. 204-10.
137. Debener, S., et al., *Single-trial EEG-fMRI reveals the dynamics of cognitive function.* Trends Cogn Sci, 2006. **10**(12): p. 558-63.
138. Barbati, G., et al., *Optimization of an independent component analysis approach for artifact identification and removal in magnetoencephalographic signals.* Clin Neurophysiol, 2004. **115**(5): p. 1220-32.
139. Delorme, A. and S. Makeig, *EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis.* J Neurosci Methods, 2004. **134**(1): p. 9-21.
140. Sahonero-Alvarez, G. *A Comparison of SOBI, FastICA, JADE and Infomax Algorithms*. 2017.
141. Lee, T.W., M. Girolami, and T.J. Sejnowski, *Independent component analysis using an extended infomax algorithm for mixed subgaussian and supergaussian sources.* Neural Comput, 1999. **11**(2): p. 417-41.
142. Makeig, S., et al., *Blind separation of auditory event-related brain responses into independent components*. Proc Natl Acad Sci U S A, 1997. **94**(20): p. 10979-84.
143. Lu, W. and J.C. Rajapakse, *ICA with Reference.* Neurocomputing, 2006. **69**(16-18): p. 2244-2257.
144. Ablin, P., J.-F. Cardoso, and A. Gramfort, *Faster Independent Component Analysis by Preconditioning With Hessian Approximations.* IEEE Transactions on Signal Processing, 2018. **66**(15): p. 4040-4049.
145. Rifai, C., et al., *Channels selection using independent component analysis and scalp map projection for EEG-based driver fatigue classification.* Annu Int Conf IEEE Eng Med Biol Soc, 2017. **2017**: p. 1808-1811.
146. Direito, B., et al., *Feature selection in high dimensional EEG features spaces for epileptic seizure prediction.* IFAC Proceedings Volumes, 2011. **44**(1): p. 6206-6211.
147. Subhani, A.R., et al., *MRMR based feature selection for the classification of stress using EEG*, in *2017 Eleventh International Conference on Sensing Technology (ICST)*. 2017. p. 1-4.

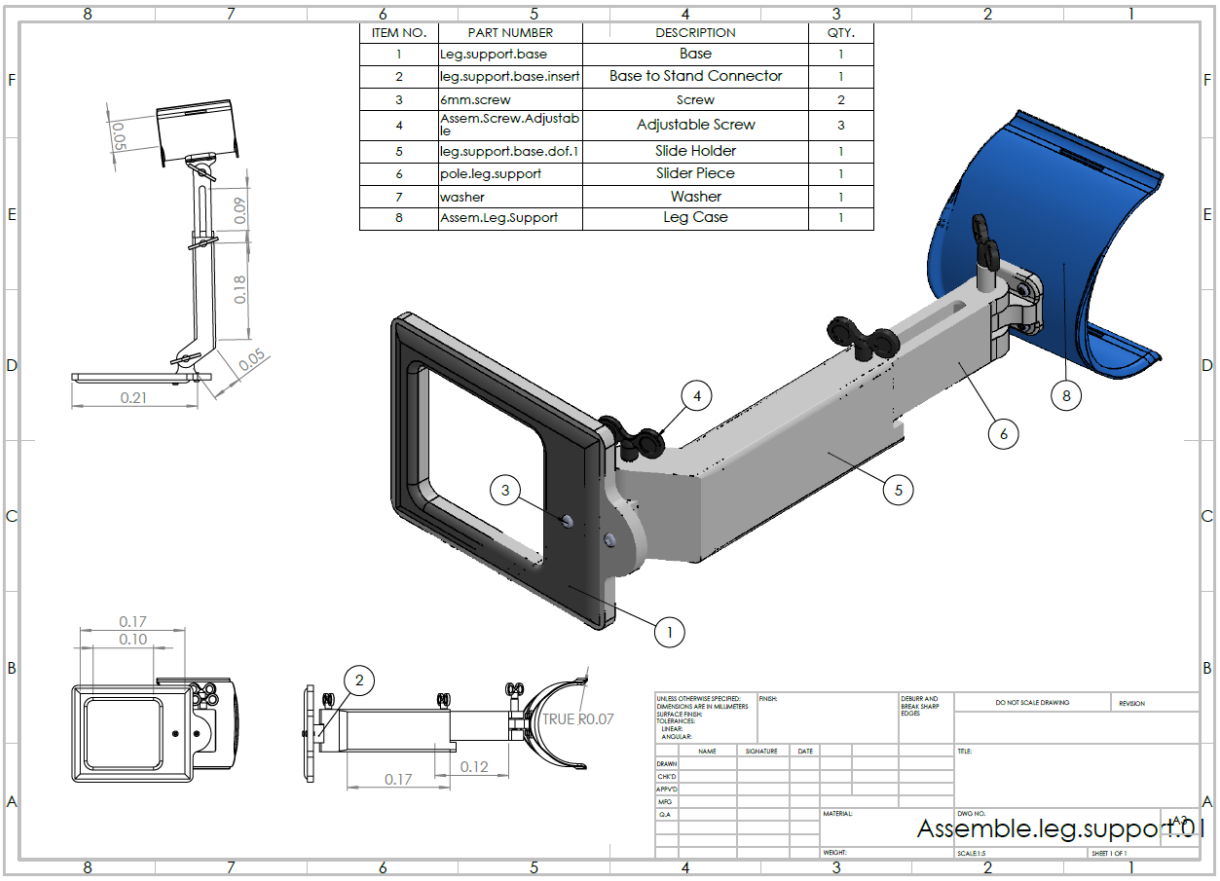
148. Xu, H., et al., *Research on EEG Channel Selection Method for Emotion Recognition*, in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 2019. p. 2528-2535.
149. Zhao, Z., R. Anand, and M. Wang, *Maximum Relevance and Minimum Redundancy Feature Selection Methods for a Marketing Machine Learning Platform*, in *2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. 2019. p. 442-452.
150. Tzimourta, K.D., et al., *Evaluation of window size in classification of epileptic short-term EEG signals using a Brain Computer Interface software.* Engineering, Technology & Applied Science Research, 2018. **8**(4): p. 3093-3097.
151. Candra, H., et al., *Investigation of window size in classification of EEG-emotion signal with wavelet entropy and support vector machine.* Annu Int Conf IEEE Eng Med Biol Soc, 2015. **2015**: p. 7250-3.
152. Bermúdez i Badia, S., et al., *Finding the Optimal Time Window for Increased Classification Accuracy during Motor Imagery*, in *Proceedings of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies*. 2021. p. 144-151.
153. Jiang, D., et al., *Robust sleep stage classification with single-channel EEG signals using multimodal decomposition and HMM-based refinement.* Expert Systems with Applications, 2019. **121**: p. 188-203.
154. Stancin, I., M. Cifrek, and A. Jovic, *A Review of EEG Signal Features and their Application in Driver Drowsiness Detection Systems.* Sensors (Basel), 2021. **21**(11).
155. Donoghue, T., J. Dominguez, and B. Voytek, *Electrophysiological Frequency Band Ratio Measures Conflate Periodic and Aperiodic Neural Activity.* eNeuro, 2020. **7**(6).
156. Sen, B., et al., *A comparative study on classification of sleep stage based on EEG signals using feature selection and classification algorithms.* J Med Syst, 2014. **38**(3): p. 18.
157. Cohen, G.H. and G.A. Coon, *Discussion: "On the Automatic Control of Generalized Passive Systems" (Chien, Kun Li, Hrones, J. A., and Reswick, J. B., 1952, Trans. ASME, 74, pp. 175–183).* Journal of Fluids Engineering, 1952. **74**(2): p. 184-185.
158. Ziegler, J.G. and N.B. Nichols, *Optimum Settings for Automatic Controllers.* Journal of Dynamic Systems, Measurement, and Control, 1993. **115**(2B): p. 220-222.
159. Yamamoto, T. and S.L. Shah, *Design and experimental evaluation of a multivariable self-tuning PID controller.* IEE Proceedings - Control Theory and Applications, 2004. **151**(5): p. 645-652.
160. Yu, D.L., T.K. Chang, and D.W. Yu, *A stable self-learning PID control for multivariable time varying systems.* Control Engineering Practice, 2007. **15**(12): p. 1577-1587.

161. Lewis, F.L. and D. Vrabie, *Reinforcement learning and adaptive dynamic programming for feedback control.* IEEE Circuits and Systems Magazine, 2009. **9**(3): p. 32-50.
162. Lee, H. and N. Hogan, *Investigation of human ankle mechanical impedance during locomotion using a wearable ankle robot*, in *2013 IEEE International Conference on Robotics and Automation*. 2013. p. 2651-2656.
163. Jin, Z., H. Li, and H. Gao, *An intelligent weld control strategy based on reinforcement learning approach.* The International Journal of Advanced Manufacturing Technology, 2018. **100**(9-12): p. 2163-2175.
164. Guan, Z. and T. Yamamoto, *Design of a Reinforcement Learning PID controller*, in *2020 International Joint Conference on Neural Networks (IJCNN)*. 2020. p. 1-6.
165. Kofinas, P. and A.I. Dounis, *Online Tuning of a PID Controller with a Fuzzy Reinforcement Learning MAS for Flow Rate Control of a Desalination Unit.* Electronics, 2019. **8**(2).
166. Sun, Q., et al., *Design and application of adaptive PID controller based on asynchronous advantage actor–critic learning method.* Wireless Networks, 2019. **27**(5): p. 3537-3547.
167. Shipman, W.J. and L.C. Coetzee, *Reinforcement Learning and Deep Neural Networks for PI Controller Tuning.* IFAC-PapersOnLine, 2019. **52**(14): p. 111-116.
168. Shi, Q., et al., *Adaptive neuro-fuzzy PID controller based on twin delayed deep deterministic policy gradient algorithm.* Neurocomputing, 2020. **402**: p. 183-194.
169. Silver, D., et al., *Deterministic Policy Gradient Algorithms*, in *Proceedings of the 31st International Conference on Machine Learning*, P.X. Eric and J. Tony, Editors. 2014, PMLR: Proceedings of Machine Learning Research. p. 387--395.
170. Lakhani, A.I., M.A. Chowdhury, and Q. Lu, *Stability-preserving automatic tuning of PID control with reinforcement learning.* Complex Engineering Systems, 2022. **2**.
171. Zhong, J. and Y. Li. *Toward Human-in-the-Loop PID Control Based on CACLA Reinforcement Learning*. in *International Conference on Intelligent Robotics and Applications*. 2019. Springer.
172. el Hakim, A., H. Hindersah, and E. Rijanto, *Application of reinforcement learning on self-tuning PID controller for soccer robot multi-agent system*, in *2013 Joint International Conference on Rural Information & Communication Technology and Electric-Vehicle Technology (rICT & ICeV-T)*. 2013. p. 1-6.
173. R. Sutton and A. Barto., *Reinforcement Learning: An Introduction*. 2nd ed. 2018, London, England: The MIT Press. 548.
174. Troia, S., et al., *On Deep Reinforcement Learning for Traffic Engineering in SD-WAN.* IEEE Journal on Selected Areas in Communications, 2021. **39**(7): p. 2198-2212.

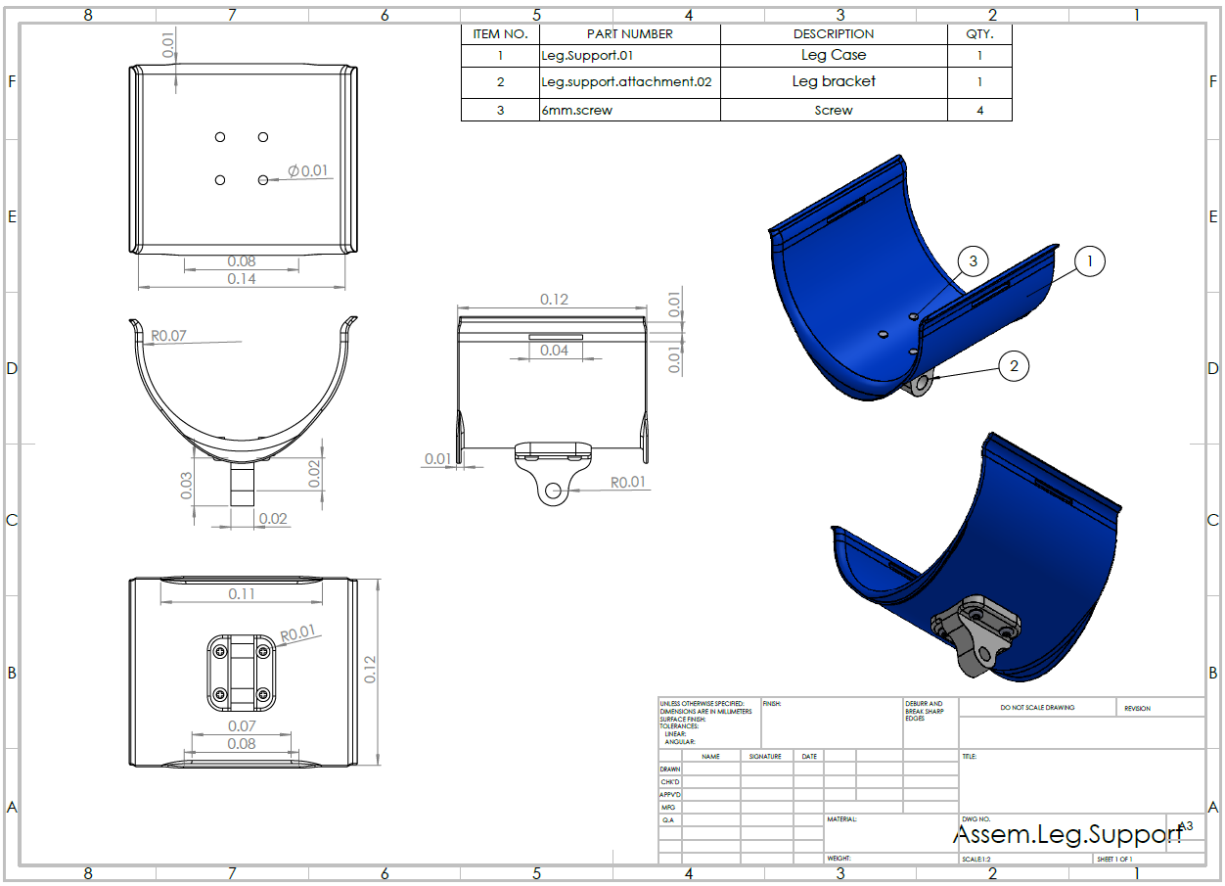
175. Silver, D., R.S. Sutton, and M. Müller, *Temporal-difference search in computer Go.* Machine Learning, 2012. **87**(2): p. 183-219.
176. Shin, J., et al., *Reinforcement Learning – Overview of recent progress and implications for process control.* Computers & Chemical Engineering, 2019. **127**: p. 282-294.
177. Barto, A.G., R.S. Sutton, and C.W. Anderson, *Neuronlike adaptive elements that can solve difficult learning control problems.* IEEE Transactions on Systems, Man, and Cybernetics, 1983. **SMC-13**: p. 834-846.
178. Lillicrap, T.P., et al., *Continuous control with deep reinforcement learning.* arXiv preprint arXiv:1509.02971, 2015.
179. Mnih, V., et al., *Human-level control through deep reinforcement learning.* Nature, 2015. **518**(7540): p. 529-33.
180. Mnih, V., et al., *Playing atari with deep reinforcement learning.* arXiv preprint arXiv:1312.5602, 2013.
181. Lawrence, N.P., et al., *Reinforcement Learning based Design of Linear Fixed Structure Controllers.* IFAC-PapersOnLine, 2020. **53**(2): p. 230-235.
182. Moraux, A., et al., *Ankle dorsi- and plantar-flexion torques measured by dynamometry in healthy subjects from 5 to 80 years.* BMC Musculoskelet Disord, 2013. **14**: p. 104.
183. Maas, A.L. *Rectifier Nonlinearities Improve Neural Network Acoustic Models*. 2013.
184. Ioffe, S. and C. Szegedy, *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, in *Proceedings of the 32nd International Conference on Machine Learning*, B. Francis and B. David, Editors. 2015, PMLR: Proceedings of Machine Learning Research. p. 448--456.
185. Kingma, D.P. and J. Ba, *Adam: A method for stochastic optimization.* arXiv preprint arXiv:1412.6980, 2014.
186. Okafor, E., et al., *Heuristic and deep reinforcement learning-based PID control of trajectory tracking in a ball-and-plate system.* Journal of Information and Telecommunication, 2020. **5**(2): p. 179-196.
187. Dogru, O., et al., *Reinforcement learning approach to autonomous PID tuning.* Computers & Chemical Engineering, 2022. **161**.
188. Mukhopadhyay, R., et al., *Performance Analysis of Deep Q Networks and Advantage Actor Critic Algorithms in Designing Reinforcement Learning-based Self-tuning PID Controllers*, in *2019 IEEE Bombay Section Signature Conference (IBSSC)*. 2019. p. 1-6.
189. Qin, Y., et al., *Improve PID controller through reinforcement learning*, in *2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC)*. 2018. p. 1-6.
190. Plappert, M., et al., *Parameter space noise for exploration.* arXiv preprint arXiv:1706.01905, 2017.

9 APPENDIX

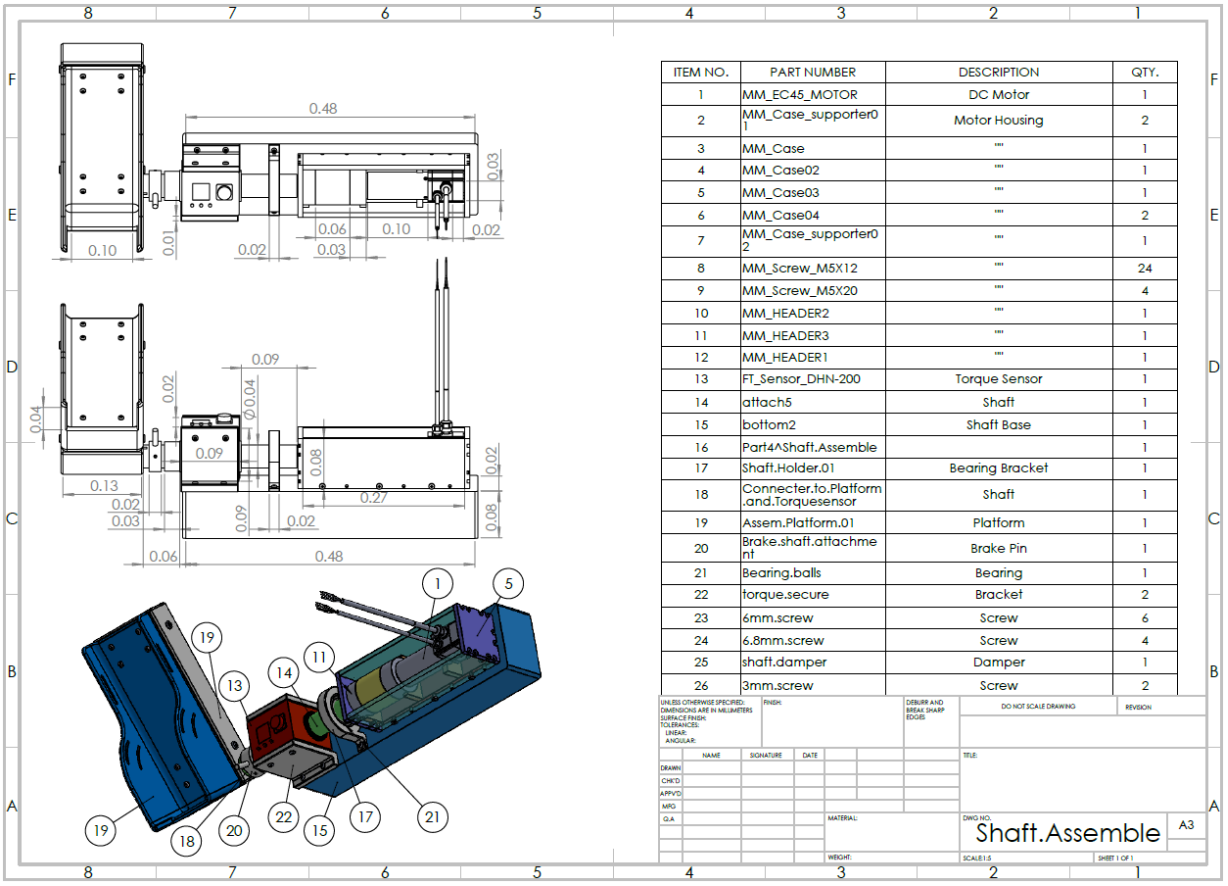
Leg Support Fully Assembly



Leg Calf Muscle Support Assembly



Ankle Robot Shaft Assembly



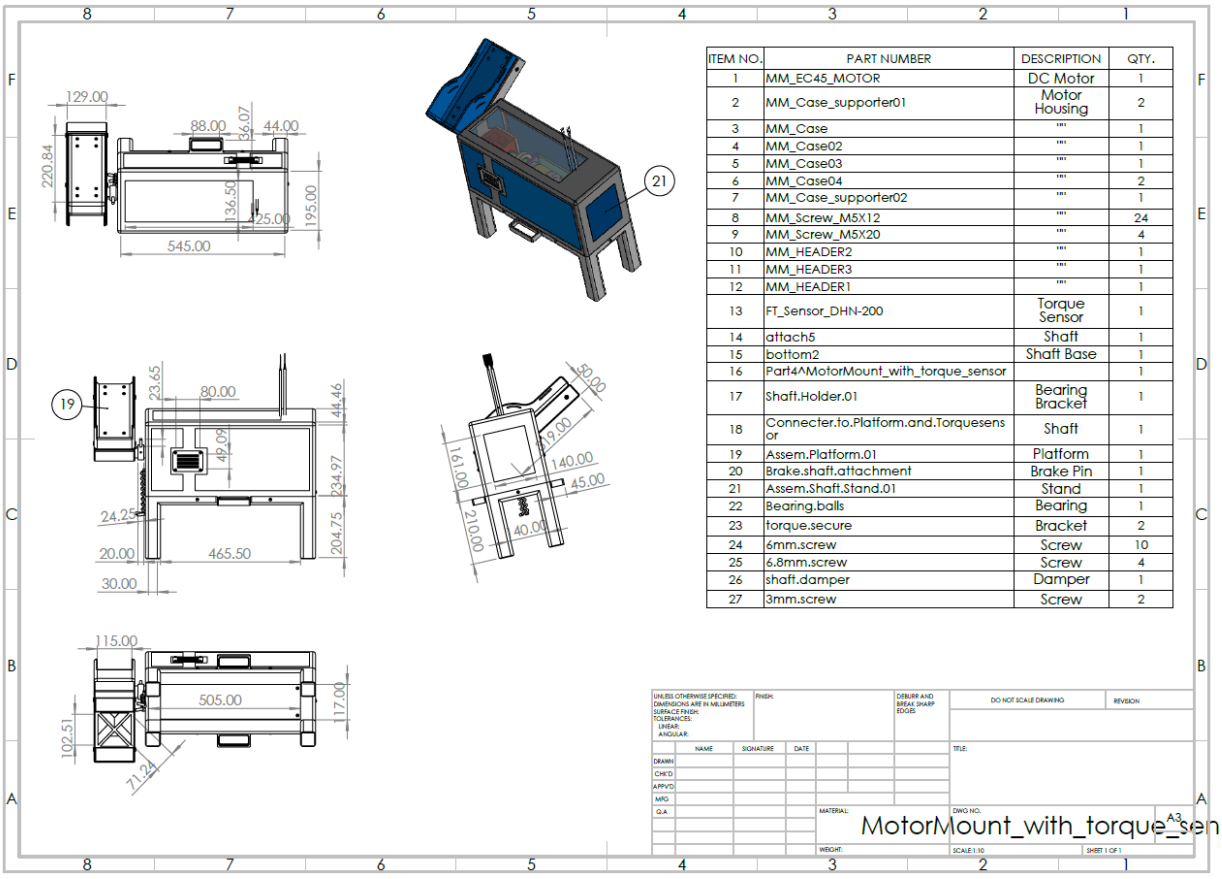
ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	MM_EC45_MOTOR	DC Motor	1
2	MM_Case_supporter01	Motor Housing	2
3	MM_Case	""	1
4	MM_Case02	""	1
5	MM_Case03	""	1
6	MM_Case04	""	2
7	MM_Case_supporter02	""	1
8	MM_Screw_M5X12	""	24
9	MM_Screw_M5X20	""	4
10	MM_HEADER2	""	1
11	MM_HEADER3	""	1
12	MM_HEADER1	""	1
13	FT_Sensor_DHN-200	Torque Sensor	1
14	attach5	Shaft	1
15	bottom2	Shaft Base	1
16	Part4^Shaft.Assemble		1
17	Shaft.Holder.01	Bearing Bracket	1
18	Connecter.to.Platform.and.Torquesensor	Shaft	1
19	Assem.Platform.01	Platform	1
20	Brake.shaft.attachment	Brake Pin	1
21	Bearing.balls	Bearing	1
22	torque.secure	Bracket	2
23	6mm.screw	Screw	6
24	6.8mm.screw	Screw	4
25	shaft.damper	Damper	1
26	3mm.screw	Screw	2

UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

MATERIAL:
WEIGHT:
TITLE:
DWG NO. Shaft.Assemble
A3
SCALE:1:5
SHEET 1 OF 1

Ankle Robot Casing assembly



ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	MM_EC45_MOTOR	DC Motor	1
2	MM_Case_supporter01	Motor Housing	2
3	MM_Case	"	1
4	MM_Case02	"	1
5	MM_Case03	"	1
6	MM_Case04	"	2
7	MM_Case_supporter02	"	1
8	MM_Screw_M5X12	"	24
9	MM_Screw_M5X20	"	4
10	MM_HEADER2	"	1
11	MM_HEADER3	"	1
12	MM_HEADER1	"	1
13	FT_Sensor_DHN-200	Torque Sensor	1
14	attach5	Shaft	1
15	bottom2	Shaft Base	1
16	Part4^MotorMount_with_torque_sensor		1
17	Shaft.Holder.01	Bearing Bracket	1
18	Connecter.to.Platform.and.Torquesensor	Shaft	1
19	Assem.Platform.01	Platform	1
20	Brake.shaft.attachment	Brake Pin	1
21	Assem.Shaft.Stand.01	Stand	1
22	Bearing.balls	Bearing	1
23	torque.secure	Bracket	2
24	6mm.screw	Screw	10
25	6.8mm.screw	Screw	4
26	shaft.damper	Damper	1
27	3mm.screw	Screw	2

Ankle footplate Assembly