



Addis Ababa University
Addis Ababa Institute of Technology
School of Electrical and Computer Engineering
MSc. in Control Engineering
**Neural Network Based Lower Limb Prosthetic Control
Using Super Twisting Sliding Mode Control**

By: Adisu Tadesse

Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Masters of Science in Electrical Control Engineering

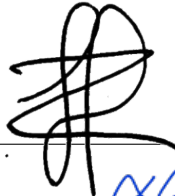
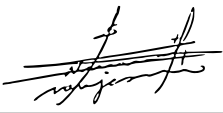
Advisor: Dr. Chala Merga (PhD)
January 2025.
Addis Ababa, Ethiopia



Addis Ababa University
Addis Ababa Institute of Technology
School of Electrical and Computer Engineering
**Neural Network Based Lower Limb Prosthetic Control
Using Super Twisting Sliding Mode Control**

By: Adisu Tadese

APPROVED BY BOARD OF EXAMINERS

Name	Signature	Date
(School Dean)		
Chala Merga (PhD)		20 January 2025
(Advisor)		
Lebework Hegerch		Jan 31 / 2025
(Internal Examiner)		
Nebiyu Tenaye		January 30, 2025
(External Examiner)		

Declaration

This is to declare that the thesis/dissertation entitled “Neural Network Based Lower Limb Prosthetic Control Using Super-Twisting Sliding Mode Control (STSMC),” submitted in partial fulfillment of the requirements for a Masters of Science in Control Engineering, is a record of original work carried out by me and has never been submitted to any other institution to get any other degree or certificates. The assistance I received during this research have been duly acknowledged.

January 2025

Acknowledgements

I extend my sincere gratitude to God for His grace, strength, and paternal love during my academic path to this master's level. Divine kindness has facilitated my excellence and achievement in all academic pursuits. I would like to express my deep gratitude to my advisor, Dr. Chala Merga, for his steadfast support throughout my MSc studies and research, as well as his patience, enthusiasm, broad knowledge, and, most importantly, his confidence in my capabilities. His counsel was essential during the research and preparation of the thesis. I like to convey my sincere appreciation to my parents for their encouragement and emotional support in the successful completion of my thesis, as well as to my classmates for their invaluable contributions throughout my research.

Abstract

A prosthetic is an artificial limb employed to substitute lost anatomical structures due to disease, injury, or trauma. In this paper, an approach for controlling the prosthetic leg using a surface Electromyography (sEMG) signal, an artificial neural network (ANN), and super twisting sliding mode control (ST-SMC) is presented. The triggering signal is extracted from surface electromyographic (sEMG) signals recorded from the nine muscles at the lower limb, which are grouped into three movement groups. Intense signal analysis that includes filtering, rectification, and linearization is done to obtain the reference sEMG to the NN from the raw sEMG because the raw sEMG data can't be used as training data for NN. An artificial neural network (ANN) predicts the joint angle for walking, upstairs and downstairs using the processed sEMG signals of the muscles. The super-twisting sliding mode control (ST-SMC) is used to regulate the motion of the prosthetic joints in accordance with the specified reference trajectories. The kinematic model is formulated using forward and inverse kinematics principles, which determine the position and orientation of the prosthetic leg. The dynamic model is based on a fixed coordinate system of human lower leg modeling, formulated by the Euler-Lagrange principle. Rather than using a simple triple pendulum model, the proposed approach presents a more realistic model of the human lower leg. Matlab software's signal analyzer, neural network fitting packages, and MATLAB/Simulink are used for signal analysis, neural network training, and model simulation. The dynamic system modeling and ST-SMC controller design are implemented. Parameter change analysis and disturbance analysis were done to show that the controller is robust against internal parameter changes and external environmental changes. From the simulation results, it is observed that training the neural network with processed data and implementing the ST-SMC results in increased regression value and decreased mean squared error (MSE) in trajectory tracking instead of using SMC.

Keywords: Prosthesis, surface Electromyography (sEMG), Artificial neural network (ANN), Super twisting Sliding Mode Control (ST-SMC).

Table of contents

List of figures	v
List of tables	vii
Acronyms	viii
1 INTRODUCTION	1
1.1 Background of the Study	1
1.2 Statement of the Problem	4
1.3 Objectives	5
1.3.1 General Objectives	5
1.3.2 Specific Objective	5
1.4 Significance of the Study	5
1.5 Methodology	6
1.6 Contributions	6
1.7 Scope of the Thesis	7
1.8 Outlines of the Thesis	7
2 LITERATURE REVIEW	8
2.1 Theoretical Background	8
2.1.1 Human Giant Cycle	8
2.1.2 Data Extraction Method	9
2.1.3 Overview of Prostatic Control Techniques	10
2.2 Limitations and Future Trends in Prostatic Leg Control	12
2.2.1 Prostatic Leg Control Limitations	12
2.2.2 Future Trends on Prostatic Leg Control	13
3 Methodology	14
3.1 sEMG Signal Acquisition and Processing	14
3.2 System Modeling Approach	16
3.3 Neural Network Training Method	17
3.4 Control Method for the Prosthetic Leg	18

4	System Modeling	20
4.1	Kinematic Modeling and Calculation	20
4.2	Forward Kinematics	20
4.3	Inverse Kinematics Analysis	23
4.3.1	Velocity Analysis	25
4.4	Dynamic Modeling and Calculation	27
4.5	sEMG Data Collection and Preparation	32
4.6	FFNN for Angle Estimation	34
5	CONTROLLER DESIGN	39
5.1	State equation of the system	39
5.2	SMC Controller Design	42
5.2.1	Conventional SMC Controller Design	42
5.3	Super Twisting Sliding Mode Control (ST-SMC)	45
5.4	Dc Motor Modeling	47
6	Result and Discussion	51
6.1	Model Validations	51
6.1.1	sEmg Signal Proccesing	54
6.2	Joint Angle Smotthing Using Savitzky-Golay filter	57
6.2.1	FFNN Training	59
6.3	controller design	61
6.3.1	Walking, DNS and UPS trajectory tracking	61
6.3.2	Parameter change Analysis	68
6.3.3	Distubance Analysis	69
7	Conclusion and Future Works	71
	Appendix Appendix	72
.1	Proof of Stability	72
.2	Program for sEMG signal processing	75
.3	programs to call another experimental Subject	77
.4	Matlab Program for NN Training	78
.5	Simulink model	79
	References	80

List of figures

1.1	Block diagram of active lower limb prosthetics control.	4
3.1	Flow chart of the NN-based lower limb prosthetic controller design process.	19
4.1	Kinematics for DH-parameter Analysis.	21
4.2	Inverse kinematics θ_1	23
4.3	Inverse kinematics θ_3	24
4.4	Inverse kinematics θ_2	25
4.5	Human lower limb for dynamic modeling	28
4.6	FF-Neural network.	37
5.1	ST-SMC Block diagram	45
5.2	ST-SMC Block diagram	48
5.3	Applied actuator torque τ	48
6.1	Plant Hip angle open loop response for model validation.	52
6.2	Plant Knee angle open loop response for model validation.	53
6.3	Plant Ankle angle open loop response for model validation.	53
6.4	Raw sEMG VastusMedialis	54
6.5	Shows low-pass filtered and High-pass filtered Vastusmedialis from raw data.	55
6.6	Shows the full wave rectified and the Linera Vastus mediales after further processing.	55
6.7	Shows how the raw sEMG Vastusmedialis data is converted into a linear signal to be used as an input to NN.	55
6.8	Shows how the raw Tensorfascialata and Upper Tibialisanterior sEMG data converted into a linear signal to be used as an input to NN.	56
6.9	Shows how the raw Rectusfmurus and Semimembrosis sEMG data converted into a linear signal to be used as an input to NN.	56
6.10	Shows how the raw MedIalgastrocnemius and TibialisAnterior sEMG data are converted into a linear signal to be used as an input to NN.	56
6.11	Shows Hip angle Unfiltered vs Filtered.	57
6.12	Shows Knee angle Unfiltered vs Filtered.	58

6.13	Shows Ankle angle Unfiltered vs Filtered.	58
6.14	NN trained Hip angle.	59
6.15	Regrasion value for Hip angle training.	59
6.16	NN trained Knee angle.	60
6.17	Regrasion value for knee angle training.	60
6.18	NN trained Ankle angle.	60
6.19	Regrasion value for Ankle angle training.	60
6.20	STSMC and SMC hip angle.	61
6.22	STSMC and SMC knee angle.	62
6.24	STSMC and SMC Ankle angle.	63
6.26	RMS value for hip angle ST-SMC and SMC controlled.	64
6.27	RMS value for knee angle ST-SMC and SMC controlled.	64
6.28	RMS value for ankle angle ST-SMC and SMC controlled.	65
6.29	DNS Hip joint trajectory ST-SMC and SMC controlled.	65
6.30	DNS Knee joint trajectory ST-SMC and SMC controlled.	66
6.31	DNS Ankle joint trajectory ST-SMC and SMC controlled.	66
6.32	UPS hip joint trajectory ST-SMC and SMC controlled.	67
6.33	UPS knee joint trajectory ST-SMC and SMC controlled.	67
6.34	UPS ankle joint trajectory ST-SMC and SMC controlled.	67
6.35	STSMC Hip angle for Subject 1, Subject 2, and Subject 3.	68
6.36	STSMC Knee angle for Subject 1, Subject 2, and Subject 3.	68
6.37	STSMC ankle angle for Subject 1, Subject 2, and Subject 3.	69
6.38	ST-SMC controlled Hip angle with and without disturbance.	70
6.39	ST-SMC controlled Knee angle with and without disturbance.	70
6.40	ST-SMC controlled Ankle angle with and without disturbance.	70
1	Simulink Model	79

List of tables

4.1	DH Parameters.	22
4.2	Constants for model simplification.	30
6.1	Parameter for subject 1	52
6.2	Leg model open loop response for model validation.	53
6.3	sEMG signals filtering parameters	54
6.4	Savitzky-Golay filter parameter for Smoothing	57
6.5	Controller design parameters	61
6.6	Energy consumption of prosthetic for SMC and ST-SMC.	63
6.7	Average MSE values for walking trajectory tracking.	64
6.8	Controller design parameters for DNS.	65
6.9	Controller design parameters for UPS.	66
6.10	MSE(deg) analyses for parameter change.	68
6.11	MSE(deg) analyses with and without disturbance.	69

Acronyms

Acronyms / Abbreviations

AI	Artificial Intelligence
ALLP	Active Lower Limb Prosthesis
AMI	Agonist–antagonist Mayo-Neuronal Interface
ANN	Artificial Neural Network
BCIs	Brain Computer Interfaces
BP	Back-Propagation
DH	Denavit–Hartenberg
DNS	Down Stairs
DOF	Degrees Of Freedom
EEG	Electroencephalography
EMG	surface Electromyography
FFNN	Feed Forward Neural Network
IK	FK Forward Kinematics
FL	Fuzzy Logic
HMI	Human machine interface
iEMG	intramuscular Electromyogram
IK	IK Inverse Kinematics
IMUs	Inertial Measurement Units
LMA	Levenberg-Marquardt Algorithm
LR	Loading Response

MATLAB Matrix Laboratory

MAV Mean Absolute Value

MLP Multi-Layer Perceptron

PID Proportional Integral Derivative

RFC Random Forest Classifier

RNN Recurrent Neural Network

SMC Sliding Mode Control

STSMC Super Twisting Sliding Mode Control

UPS Up- Stairs

ZCV Zero-Crossing Value

Chapter 1

INTRODUCTION

1.1 Background of the Study

The continuous attempt to improve the quality of life for people who have undergone lower limb amputations has driven significant advancements in the field of active lower limb Prosthetic recently. Advanced control systems on powered prosthetic limbs may enable reestablishment of normal gait patterns and facilitate the performance of many everyday tasks like stair climbing and other activities [1]. The key challenges, current technological advancements, and future directions in the area of active lower limb prosthesis control research and development are underlined in this paper.

The most prevalent kind of amputation in the world is lower extremity amputation [2], which has a major negative physiological and physical effect on patients. The 2022 report of world health organization (WHO) indicates that people between the ages of 45 and 59 have the greatest rate of amputation [3], which may be caused by illnesses, accidents, or congenital deficiencies. A Lower leg amputation is a concerning issue. Research indicates that from the total population of the world a total of 57.7 million individuals have had a lower limb amputated by 2019 [4], which is a significant amount and needs technological intervention. There are different types of amputations which are defined in scientific terms as hemipelvectomy or trans-pelvic amputation, above-the-knee or transfemoral amputation, knee-level amputation, below-the-knee or transtibial amputation, and partial foot or ankle amputation [2]. Two primary categories of prostheses have been established. Passive prostheses (which uses an external force to perform movements), some times they can be aesthetics only (which mimics the appearance of a natural limb without adding functionality), or it could be a supporting tool (which can be modified to perform specific movements based on the desired activity) [3]. The other type is active prostheses (which are prostheses that use external actuators to provide movement and also provides control, providing high performance and functionality at the expense of complexity) [4].

Modern lower leg Prosthetic are classified into two categories. Passive Prosthetic, which are unpowered mechanical limbs that depend on the user's residual muscular

strength and have elastic joints; and active (powered) prostheses, which are equipped with sensors and motors to facilitate movement. One of the key motivations for the development of active lower leg prosthetics is the need to get over the disadvantages of traditional passive prosthetic limbs. Traditional prosthetic limbs greatly reduce the quality of life of amputees by missing the support and ability required for complex movements like up-stairs and down-stairs. In order to address this challenge, scientists have focused on developing active prosthetic limbs that reproduce the natural behavior of the human leg by means of advanced sensors, actuators, and control systems. In the past few decades, electronic and control algorithm advancements have been the catalyst for significant technical advancements in active prosthetic in a variety of domains, including complex mobility and comfort. The human body is a complicated system that involves both mental and physical coordination. As a result, a number of technologies have been used in order to fulfill the locomotion requirements of the end user. Artificial intelligence and machine learning algorithms in the field of lower leg prostheses are increasingly integral to the control and adaptability of lower leg prostheses, helping address unique challenges like predicting user intent and adapting to complex terrains.

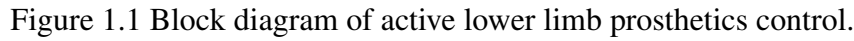
The advancement of lower limb prosthetics has been shaped by advancements in medicine and control engineering. The damage to the lower limb due to amputation is recognized to impact mobility and occupational capacity; thus, contemporary prosthetics strive to restore that lost functionality to the highest degree possible. Lower leg Prosthetic can be classified based on design and technology, ranging from simple mechanical limbs to sophisticated robotic systems engineered to mimic natural movement. The objective of contemporary prosthetic devices is to ensure that the movements of standing, walking, and running resemble those of a typical individual. The biomechanical architecture of the lower extremities in a healthy human body has three joints, which are the hip joint, the knee joint, and the ankle joint, each contributing to the gait cycle, which is categorized into stance and swing phases.

Despite major advancements in prosthetic legs in recent years, their control continues to pose multiple challenges. Human gait is dynamic and involves precise coordination of the hip, knee, and ankle joints, especially when transitioning between different terrains, such as stairs or uneven surfaces. Prosthetic control systems struggle to replicate this intricate coordination. Many prosthetics are designed for flat surfaces and do not adapt seamlessly to environmental changes, affecting the user's efficiency and stability. Also Even though sEMG signals have the potential to operate prosthetic, they are still difficult to interpret. This problem is made worse by porous electrodes, user-specific differences in muscle contractions, and various skin impedances. Such discrepancies may result in unpredictable prosthetic control.

Emerging developments in the lower extremity prosthetic limb advance through materials science, robotics, and artificial intelligence, making it imperative for the industry to pursue these developments. These trends include bionic limbs, which are prosthetic

devices engineered to accept EMG signals and provide a direct connection to the user's nervous system, therefore granting the user control over the prosthesis. The osseointegration is an approach that facilitates a secure connection between the prosthetic limb and the bone, providing the user with enhanced comfort and control. Adaptive prosthesis encompasses prosthetic devices augmented with machine learning that facilitates the adaptation of the user's walking style for enhanced mobility. A multitude of control mechanisms exists for lower limb amputation, each possessing distinct advantages and downsides. In this paper, challenges and limitations on the existing control methods are discussed, and finally, future directions in research and development in lower leg prostheses is discussed. Myoelectric control (sEMG based) utilizes electrical signals from the muscles to operate the prosthetic limb [5]. The sensors on the prosthetic leg detect these impulses and transmit them to a computer, which then controls the leg's motors. Myoelectric control is the most prevalent method for lower leg prosthetics due to its relative ease of use and the absence of the need for supplementary surgical procedures. Nevertheless, individuals with nerve damage or muscle atrophy may encounter difficulties in utilizing myoelectric control. Mechanical control employs levers and cables to manipulate the prosthetic leg [6]. The joints of the prosthetic leg are actuated by the user by levers and cables. Employing mechanical control is quite simple and does not require electronics or batteries. Nevertheless, individuals with restricted strength or range of motion may encounter difficulties in utilizing mechanical control. Hybrid control integrates components of myoelectric and mechanical control [7]. The prosthetic leg is equipped with sensors that detect myoelectric signals, in addition to levers and wires that the user can use to control the prosthetic leg's movement. Hybrid control provides the user with enhanced command over the prosthetic leg; yet, it is more intricate and challenging to operate compared to myoelectric or mechanical control. Brain-computer Interfaces (BCI) utilize neural impulses to operate the prosthetic limb [8]. The user wears a cap equipped with electrodes that monitor cerebral activity. The brain signals are transmitted to a computer, which subsequently regulates the motors in the leg. The individual is instructed to utilize a certain concept, such as flexing a knee, for control. Thoughts produce electrical activity in nerve cells and generate brainwaves. A chip may be implanted in the brain to detect electrical activity, or electrodes can be positioned on the scalp to capture brainwaves. BCIs remain in the nascent phase of development, yet they possess the capacity to transform prosthetic control.

In this paper, sEMG signals are filtered and rectified so that they can be used as input by a neural network, which subsequently translates them into the corresponding robotic leg motions. The proposed methodology utilizes a neural network and sEMG data to forecast the intended movement of the robotic leg. A robust ST-SMC is implemented, which is favorable due to its robustness against uncertainties, adaptability to varying user needs, and ability to handle nonlinear dynamics of human movement.



Despite significant advancements in prosthetic technologies, the challenge of controlling lower-limb prostheses in a natural and intuitive manner persists. The prosthetic should allow adapting to different terrains and aspects, such as walking pace or activities like stair ascent, without jerky movements or unreasonable delays. The control systems of prosthetic legs must meet the requirements of accurate relative motion of the joints, easy effort for the user, and accommodating muscle activation signals that vary with time. Thus, the sEMG signals extracted from the different muscle groups must be processed and fed to the neural network, which generates the required trajectory, which will be enforced by the sliding mode controller. However, there remains a challenge to properly control the prosthetic leg limbs in accordance with the outputs of the neural network in normal conditions because:

- Therefore, a neural network-based ST-SMC controller approach whose reference is from FFNN-predicted angle (sEMG as input to the NN and measured joint angle as target to the NN) is a promising solution to tackle the above problem. The STSMC technique,

characterized by its resilience against system uncertainties and external disturbances, offers an effective means to achieve the required joint trajectory. In this scenario, the neural network supplies reference angles derived from the sEMG inputs, whereas the ST-SMC guarantees that the prosthesis adheres to these angles with precision.

1.3 Objectives

1.3.1 General Objectives

The main or general objective of the research is to design, model, and control Active Lower Limb Prosthesis (ALLP) using an Artificial Neural Network (ANN) based on ST-SMC. The ANN is used to produce the reference trajectory from sEMG signals, and the ST-SMC is to force the system to track the defined trajectory.

1.3.2 Specific Objective

- To gather, process, and classify sEMG signals from lower limb muscles.
- To train the ANN from the gathered sEMG signal.
- To find the mathematical model of the system.
- To design an ANN-based ST-SMC controller for controlling the ALLP.
- Compare the performance of ST-SMC with the traditional conventional mode controller SMC.
- To simulate the overall system using MATLAB Simulink.

1.4 Significance of the Study

The utilization of ANNs and ST-SMC for active lower limb prosthetic control can markedly enhance the mobility, autonomy, and quality of life for individuals with lower limb amputations. The unique advantages of ALLP utilizing ANNs and ST-SMC encompass:

- Enhanced gait and balance: an artificial neural network-based ST-SMC can be utilized to create more safety for the patients by enhancing the gait balance. Hence decreasing the chances of falls and enhancing mobility.
- Decreased energy consumption: It has been observed that controller modeling through ANNs together with SMC can produce efficient control systems.
- Enhanced quality of life: The utilization of active prosthetic lower limb control may substantially enhance the quality of life for individuals with lower limb amputations by improving gait, balance, and energy expenditure.

- It offers an appropriate model for evaluating various recently developed control algorithms.

1.5 Methodology

The study will be achieved through the following methodology:

- Literature Survey: Start with a review of literature, which includes reading books, publication papers, and thesis works to get essential information, concepts, and ideas that will assist in the development of the thesis.
- sEMG signal analysis performed using a Matlab signal analyzer. The different muscle groups that are responsible for different movements are processed to get linear data. Because raw, noisy data affects the training of the neural network.
- Obtaining the mathematical model of the Prosthetic from defined mathematical laws that describe the behavior of the system.
- ST-SMC has been designed to enforce the required trajectory, which is robust and insensitive to external disturbances, unmodeled dynamics, and parameter uncertainties.
- A FFNN using Matlab neural network fitting app has been designed to generate the reference angles from sensed sEMG signals
- MATLAB/Simulink Simulation has been utilized for comparison of gait cycle tracking performance for conventional SMC with ST-SMC.

1.6 Contributions

In this paper, the precision, adaptability, and efficacy of prosthetic limbs are improved by our proposed approach. The following are the main contributions in this regard:

- Utilizes a fixed coordinate system human lower leg modeling approach, with the origin situated at the hip joint, aiming for modeling human motion to perform complex tasks.
- Processes sEMG signals to address challenges like high noise and interference, ensuring accurate muscle-intent interpretation for natural prosthetic control.
- Uses a feed-forward neural network to generate reference angles for hip, knee, and ankle joints of the prosthesis, translating user intent from sEMG into precise prosthetic reference angles.

- ST-SMC strategy for accurate trajectory tracking of joint angles, compensating for external disturbances and internal parameter changes.
- System adaptable to various activities such as walking, stair ascent and descent, and standing, enhancing functionality in diverse environments.

ANNs and ST-SMC can significantly improve mobility, autonomy, energy consumption, and quality of life for individuals with lower limb amputations. The model also offers a suitable model for evaluating control algorithms.

1.7 Scope of the Thesis

The scope of this thesis is to develop a mathematical model of the system, design, and simulation of a lower leg prosthetic whose reference is the human gait cycle for walking, up-stairs, and down-stairs, which will be generated by a trained neural network whose input is a linearized sEMG signal data set that is obtained by intense signal processing of raw sEMG signals from nine muscle groups. Signal processing, modeling of the system, and controller design will be implemented in a MATLAB/Simulink environment.

1.8 Outlines of the Thesis

The thesis is organized into six chapters, including the introduction in chapter 1. The rest of the thesis is organized as follows:

- Chapter 2: Describes the literature review, The theoretical background of the human gait cycle, neural network application for sEMG signals, types of SMC used for prosthetic leg control, and related works are discussed.
- Chapter 3: This chapter deals with methodologies for sEMG signal gathering, preparation, and also kinematic and dynamic modeling of the three links Prosthetic leg design is described.
- Chapter 4: Discuss mathematical system modeling and verification. FFNN training from sEMG signals to obtain the reference trajectory is also discussed.
- Chapter 5 discusses the design of SMC and ST-SMC for the prosthetics and analyzes the stability of the controller.
- Chapter 6: Presents the results obtained along with a discussion on those results briefly.
- Chapter 7: Concludes the work done in this thesis, adding recommendations and future works to be done.

Chapter 2

LITERATURE REVIEW

This review focuses on the control of lower limb prosthetic and takes readers from an initial understanding through to some key contemporary advances in prosthesis control, describing how they connect as a timeline towards solutions for future challenges within controllers. Prosthetics can give amputees their legs and mobility back, but making a control system so seamless that it works intuitively and flexibly is still incredibly challenging. These areas of investigation include EMG-controlled systems, BCI integration, and machine learning predictive algorithms for movement as well as user intent recognition. Based on the above solutions Recent studies emphasize the need for prosthetic systems that respond accurately to user commands and adjust to different terrains and motion speeds and many researchers at different institutes are working to achieve this. The review provided the analysis of control modalities and adaptive techniques related work, along with harnessing AI technology in leg prosthetic to understand literature gaps and future direction for user comfort depending on improved functionality.

2.1 Theoretical Background

2.1.1 Human Giant Cycle

The human giant cycle is defined as the sequential advancement of one foot (leg) followed by the other, subsequently returning the initial foot to perpetuate the cycle of movement [9]. Consequently, walking is a cyclical action, and analyzing one cycle can elucidate the entire duration of walking. The walking cycle can be delineated into two primary phases: the stance phase, characterized by the foot's contact with the ground, and the swing phase, during which the foot is elevated and unencumbered. The gait cycle for stair ascending and descending are discussed as follows [10]. The gait cycle for stair ascending comprises two primary phases, akin to level walking. The stance phase transpires when the foot makes contact with the ground. This phase of stair climbing is categorized into weight acceptance (loading response), pull-up, forward continuation, and foot clearance. The swing phase transpires when the foot is disengaged from the step. It comprises the

following phases: foot elevation, forward swing, and foot placement. The stairs descent gait cycle for an individual involves the first stage as an orientation step, consisting of the following phases: approach, postural adjustments, initial contact, loading response (LR), stability, and terminal posture, as will be explained thereafter. The approach step begins when the leading foot hits the first step, with the heel once more serving as the point of contact to absorb shocks. Next, the body weight is transferred to the leading leg. In the transfer of the weight on the front leg, the knee bends. As a natural response to maintain or lower the body's center of gravity, the knee bends as the body shifts forward. Therefore, the weight is limited on the front leg in order to allow its positioning and balance. The trailing leg lifts off the stairs, readying the individual for a descent. Contact with the next step occurs when the weight transfers forward and the person levels to move down the stairs. The leg advancement requires momentum and equilibrium during the swing phase. The individual descends and manages to stand firmly on the surface, whereby the standing position is maintained

2.1.2 Data Extraction Method

The extraction of data from the remaining part of the amputee and the interpretation of the acquired signal are essential for the effective engagement of the amputee and the prosthetic. This section discusses three methodologies of acquiring locomotion intention data for predicting the movement of the prosthetics.

One of the techniques of data extraction is the electromyogram signal acquisition method [11]. There are two kinds of myoelectric signal techniques. surface electromyogram (sEMG), which is placed on the surface of the skin without any surgical intervention, and intramuscular electromyogram (iEMG) [12], which is utilized on the prosthetic that are surgically connected; its high invasiveness prevents it from being widely used. By detecting the electrical difference between stimulated muscle and a reference location, sensors positioned on the joints are used to detect the exact position of the prosthesis. However, studies on this method [13] in residual limbs during various activities show that noise generated by nearby muscles and the environment can affect the performance of the system. To overcome the limitation of the iEMG method, a study used sEMG in combination with a surgery method called agonist-antagonist mayo-neuronal interface (AMI) amputation, using sensory feedback, making the signal clearer for use [14].

The other method is brain-computer interfaces (BCIs) which is a more advanced technique. In this technology, in order to ascertain a desire to move, manipulate, or interact with the surroundings, brain activity is recorded using brain-computer interfaces or electroencephalography (EEG) [15]. The majority of brain-computer interfaces rely on electroencephalography (EEG) signals, which are widely used because of their affordability, mobility, and spatiotemporal responsiveness. Based on signal frequencies ranging from $0.1Hz$ to more than $100Hz$, EEG electrodes are chosen with consideration for the

brain lobes involved in the movement task [16]. They are then categorized into mu, delta, theta, alpha, beta, and gamma based on their rhythmic activity.

Lastly in this review of data acquisition for prosthetic systems, intention prediction based on environment interaction is an additional option. In this method, future motions are anticipated based on the interaction between sensors and the outside world. Among the most often utilized pieces of equipment are cameras, force plates, force sensors, and inertial measurement units (IMUs) [17]. To carry out the user-desired action, data must be analyzed. To find patterns associated with the different movement intentions, the obtained signals must be preprocessed to remove any noise coming from the position of the electrodes and cables, and then machine learning methods (feature extraction, feature selection, and classification) are used[18]. As a complement, some researchers are working on feedback systems to enable communication from the prosthetic to the human cognition system to increase accuracy and improve control. The use of vibrotactile (signals generated by vibrating actuators or motors in contact with the user's skin) actuators allow's the residual limbs to feel what the actual prosthesis is touching [19]. Some advancements in data acquisition have been made in the medical industry. Ankle control has been demonstrated by surgery to restore the muscle relationship between agonist and antagonist (the muscle that is contracting and the muscle that is relaxing) muscles (AMI) in [20]. This study also shows better performance and capabilities by reshaping the nerves to adapt to an electrode insertion into peripheral nerves, which has given users confidence by reducing the likelihood of falling when walking on uneven terrain [21].

2.1.3 Overview of Prostatic Control Techniques

Contemporary control methodologies for lower limb prosthetics emphasize enhancing precision, accuracy, and the naturalness of the prosthetic operation. Several of the most promising novel strategies encompass the following publications. Contemporary control methodologies for lower limb prosthetic focus on improving the precision, accuracy, and naturalness of prosthetic operations, with promising novel strategies outlined in various publications.

As highlighted in their research, the research conducted by the authors [22] and [23] indicates that external perturbations applied to a real foot and integrated into the dynamic model of the ankle-knee active prosthetic, including vertical and horizontal ground reaction forces as well as external disturbances affecting the ankle joint from uneven terrain. A combination of fuzzy logic and a PID controller is used to regulate the model during a standard walking gait cycle. The findings presented by the authors [24] examine the dynamic modeling and parameter identification of the lower limb, as well as the control of a 2-DOF joint prosthetic, due to uncertainties, significant nonlinearity, imbalance issues, and external perturbations that may arise during movement. The backstepping control technique, based on Lyapunov theory, was used to ensure system stability and improve

dynamic performance, with the Bat algorithm optimization technique refined to enhance the controller's performance. In their analysis, the authors [25] and [26] employed the sliding mode control methodology, a nonlinear method characterized by notable accuracy, robustness, and ease of tuning and implementation. Parameters like sliding gain, slope, and thickness are considered to ensure system adhesion to the sliding surface. The study indicates in the following papers [26] and [27] presented a control mechanism based on neural networks that may be trained to understand the correlation between the user's intended movements and the associated myoelectric signals. This data can subsequently be utilized to regulate the prosthetic limb with greater precision and accuracy. The produced artificial neural network (ANN) in this paper is a time-delay feed-forward multilayer neural network design. This allows the networks to produce continuous estimations of joint kinematics based on sEMG, utilizing time-varying inputs and accommodating the delay between the initiation of EMG activity and mechanical movement. Additionally, there is one hidden layer alongside an input and output layer, totaling six layers. A sigmoidal transfer function was employed in the hidden layer of each neural network. The quantity of neurons in the input layer is contingent upon the number of sEMG inputs. A linear transfer function was employed in the output layer to allow the neural outputs to assume any value. A team of researchers at the University of California, Berkeley [28], developed a neural network-based prosthetic control system capable of accurately predicting the user's intended movements. The technology underwent evaluation in a clinical environment and demonstrated a substantial enhancement in the movement and autonomy of amputees.

As noted by the authors in [29] and [30], machine learning algorithms can be employed to create prosthetic control systems that adapt to the user's specific demands and preferences. A machine learning algorithm could be employed to create a prosthetic control system that autonomously adapts to the user's walking speed or terrain. Implementing machine learning classifiers to determine the locomotor mode of users of assistive devices. Deep learning models were compared to the machine learning model utilizing the random forest classifier (RFC). These algorithms were typically employed as high-level controllers capable of autonomously adjusting the functionality of lower limb prosthetic, exoskeletons, or orthoses. Also Researchers at the Massachusetts Institute of Technology (MIT) created a machine learning-driven prosthetic control system that can adjust to the user's specific requirements and preferences [31]. The technology underwent evaluation in a clinical environment and showed enhancements in the ambulation velocity and energy efficiency of amputees.

The researchers [32], [33] and [34] argue that proposed brain-computer interfaces (BCIs) enable users to manipulate prosthetic devices using cognitive processes. Brain-computer interfaces (BCIs) remain in the nascent phase of development, yet they possess the capacity to transform prosthetic control. A group of researchers at the University of Pittsburgh created a brain-computer interface (BCI) prosthetic control system enabling

users to manipulate a prosthetic limb with cognitive commands. The technology underwent evaluation in a clinical environment and demonstrated safety and efficacy. Alongside these novel strategies, researchers are endeavoring to enhance the efficacy of conventional prosthetic control methods, including myoelectric control [35] and mechanical control [36].

2.2 Limitations and Future Trends in Prostatic Leg Control

2.2.1 Prostatic Leg Control Limitations

Although there have been improvements in technology, there have been a number of dominant issues that have stunted the growth of lower limb prosthetic control, with real-time adaptability, stability, and comfort being the most common ones. Some control systems have great challenges that come up with adapting themselves to change, such as parameters, the terrain in real-time, and even the slope and the surface. Even if there are strides made with sensors and adapted algorithms, devices are often unable to mimic the human leg's functionality in different environments to the fullest. Consequently, making transitions from one terrain to another is still an issue that tends to affect the users' stability and confidence when the users are walking [37].

The most important thing when controlling the prosthetic is real-time response, but the fact that sensory data can take time to process causes lags when adjusting to abrupt movements or uneven surfaces. For instance, even though some advanced sensors like IMUs and EMG electrodes are now available in some modern prostheses, it is still very complicated to integrate these sensors without lag. To remedy these lags and create a continuous experience for users, sensor integration should be more rapid and enhance efficiency [38].

Since different individuals require a prosthesis that meets their individual needs, limited user-specific adaptation technologies pose a great challenge. Even if there were recent studies on the personalization of protasis, a number of devices still lack the appropriate fine-tuning that would match individual gait patterns, which in turn ensures great comfort when being used as it reduces inefficiencies for the user. For this same reason, studies have shown that adding customized control and feedback systems may enhance movement naturalness; yet, difficulties remain in attaining strong and flexible real-time customization [39]. Lastly, one of the limitations of both passive and active prosthetics is safety concerns due to the device's high power output, which can cause abrupt movements. Although state-of-the-art prosthetic designs focus on safety layers, without universal safety standards, the reliability of these components varies greatly across devices, and increasing stability within a prosthesis to stiffer constructs might diminish users' security during rapid movements [40]. The issues that arise in these situations emphasize further studies on sensor fusion,

increased computation speed, and adaptive control strategies are crucial for the future of an efficient lower limb prosthesis.

2.2.2 Future Trends on Prostatic Leg Control

Recent studies conducted regarding the control of lower limb prosthetics have reported some areas that need intervention. These gaps include the need for human prosthetics interaction (HPI) enhancement [41], real-time adaptability issues, insufficient control accuracy, and effective energy management. Despite existing challenges, significant advances have been made in brain-computer interfaces (BCI) [8] and electromyography (EMG)-based systems with sliding mode and PID control techniques, but using these techniques, real-time operation and noise-related problems persist when walking on uneven and dynamic surfaces. Also, the current prosthetic control mechanisms fail to cooperate well with the person's gait cycle, which would necessitate the development of more precise but user-specific movement pattern recognition aimed at better control of the prosthetic limb [42]. Further, it remains difficult to develop prostheses that automatically adjust to different terrains and user-specific biomechanics while at the same time retaining compactness and low energy consumption. While AI and machine learning algorithms can be incorporated into these prosthetics to allow for better prediction of the user's intent, implementation in most applications is however limited by the need for hardware to be embedded into the Prosthetic [42]. They propose steps that will be important in enhancing lower limb prosthetic to be user-friendly, responsive, and capable of adapting to a variety of changing scenarios.

Chapter 3

Methodology

This section outlines the procedures for designing a prosthetic leg control system using surface electromyography (sEMG) signals, Neural network , and control algorithms executed in MATLAB. This procedure entails collecting sEMG signals, training the neural network to produce a reference trajectory, and the formulation of a control scheme that utilizes the retrieved characteristics to regulate the movement of a prosthetic limb. All these steps are performed in Matlab environment.

3.1 sEMG Signal Acquisition and Processing

The study involves collecting sEMG signals from multiple subjects, focusing on lower limb muscles like quadriceps and hamstrings, using a high-quality device with a band-pass filter, and recording muscle activation patterns through various activities like walking, stair climbing, and descent. Also, the processes involve measuring joint angles simultaneously using motion capture sensors during activities to create training data for a neural network. Preprocess data to remove noise, normalize, and segment it for training. To mitigate the constraints of existing datasets, the Shenzhen Institute of Advanced Technology Lower Limb Motion Dataset (SIAT-LLMD) was utilized [43]. The referenced study devised an experiment facilitating the concurrent collection of nine channels of surface electromyography (sEMG), joint angles (kinematics), and joint torques (kinetics), along with their respective labels, from 40 healthy participants who are executing 16 distinct lower limb movements, which will be measured using the sEMG sensor. In this study, a male subject aged 23, weighing 64 kg and measuring 1.68 m, was selected, although the subject could be altered as desired. Electrodes were firmly attached to the right leg's tensor fascia lata, rectus femoris, vastus medialis, semimembranosus, upper and lower tibialis anterior, lateral and medial gastrocnemius, and soleus muscles using double-sided tape in order to obtain the nine-channel sEMG signals. The signals were sampled at different frequencies using a wireless acquisition system. Following the guidelines set out

by the institute of neurology's department of clinical neurophysiology was a must in this investigation [44]. The steps undertaken comprise:

- **Electrode Placement:** Affix surface electrodes to the principal muscles (e.g., quadriceps, hamstrings) of the residual limb.
- **Signal Acquisition:** Employ a sEMG signal acquisition system (such as Delays or comparable devices) to capture muscle activity. **Sampling Rate:** Confirm that the sampling rate is sufficiently high (often between 500 Hz and 1000 Hz) to accurately capture the frequency components of the sEMG signal.
- **Preprocessing** includes low-pass and high-pass filters applied to the raw sEmg signal, then it will be rectified using full-wave rectification of the signal by obtaining its absolute value. Then feature extraction using MAV is implemented. Finally, linerization is performed. At this point, the signal is ready to be used as input to the NN. The measured joint angle needs noise elimination, therefore employing the Savitzky-Golay filter to eradicate noise and artifacts (e.g., resulting from motion or external electrical interference).

Mean Average Value (MAV) Linearization

Feature extraction from sEMG signals often requires the linearization of the non-linear raw sEMG signal into a format that better corresponds with a linear relationship, especially when the aim is to associate sEMG features with factors like joint angle. The gait cycle for lower limb movement generally comprises stages including heel strike, stance, toe-off, and swing. Dividing the sEMG data into different stages facilitates focused investigation of the characteristics. The time domain characteristics of EMG were used to classify movement patterns. Four types of features are retrieved, which are considered very important. The mean absolute value (MAV), zero-crossing value (ZC), slope sign changes (SSC), and waveform length (WL) can be used to find the best sEMG control methods for devices that help people with their lower limbs [45].

Savitzky–Golay Filter for Joint Angle Smoothing

The Savitzky-Golay filter is a digital filter used to smooth a set of data points while preserving the shape and features of the signal, such as peak heights and widths [46]. It is commonly used in signal processing for smoothing noisy data. The filter fits a polynomial to a set of data points within a window and then evaluates the central point of the window using this polynomial. The order specifies the polynomial that fits the data in a window. Window length is the number of data points across all the filtering windows; their value varies because larger smoothing windows smooth out the noise and smaller smoothing windows capture the detail of the important features. Derivative order specifies the derivative of the filtered data.

3.2 System Modeling Approach

Forward kinematics enables the calculation of the position and orientation of the end-effector (the foot of the prosthetic limb) based on the joint angles, utilizing the Denavit-Hartenberg (DH) parameters approach. Forward kinematics calculations frequently employ the DH convention, a systematic approach to allocating coordinate frames to a robotic system's linkages. The position and orientation of the end-effector can be expressed by combining the transformation matrix for each link, which is obtained using DH parameters. The general transformation matrix from one link to the next in a 3D space is given by:

$$T_i^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

where T_i^{i-1} is the transformation matrix from frame $i - 1$ to frame i , a_i is the link length (distance between consecutive z -axes), α_i is the twist angle (angle between consecutive z -axes), d_i is the link offset (distance between consecutive x -axes along z_i) and θ_i is the joint angle (rotation around z_i).

Inverse kinematics calculates the necessary joint angles ($\theta_1, \theta_2, \theta_3$) for a specified end position (x, y) of the Prosthetic. Solving the inverse kinematics for a three-link system may necessitate complex equations and could require numerical methods or iterative approaches such as Newton-Raphson or gradient descent for practical execution. Instead, the geometrical method of inverse kinematics for a three-link prosthetic leg involves determining the joint angles that will position the leg's end-effector (the foot) at a desired location in space. For a three-link planar system (hip, knee, and ankle joints), we can solve for these angles using trigonometric relationships.

The dynamics of the three-link prosthetic leg elucidate the correlation between the exerted joint torques and the resultant motion, frequently grounded in Newtonian principles or Lagrangian mechanics. The Euler-Lagrange equation is fundamental in classical mechanics and is derived from the principle of least action. It provides the equations of motion for a system based on its Lagrangian, which is a function describing the difference between the kinetic energy K and potential energy P of the system. Euler-Lagrange Equation for a system with generalized coordinates q_i and generalized velocities $\dot{q}_i$, the Lagrangian L is defined as:

$$L(q_i, \dot{q}_i, t) = T(q_i, \dot{q}_i) - V(q_i) \quad (3.2)$$

The Euler-Lagrange equation is then given by:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = 0 \quad (3.3)$$

where q_i represents the generalized coordinates (positions) and $\dot{q}_i$ represents their time derivatives (velocities).

3.3 Neural Network Training Method

The artificial neural network is designed based on the prototype of the human biological brain system. It consists of several interconnected fundamental processing units referred to as neurons. These neurons receive input signals from the environment. Signals are altered by interconnected weights, and during a training process, neurons are activated by transfer functions to yield a specific output. FNN represents the architecture that regulates data flow within the network. It utilizes forward propagation to produce predictions and back-propagation to modify the network's weights, enabling learning from data. Although limited in some applications compared to advanced architectures like recurrent or convolutional neural networks, FNNs are essential in machine learning and deep learning [47].

The Levenberg-Marquardt method, utilized for neural network training, is frequently regarded as an enhancement of the backpropagation algorithm, specifically an optimized iteration aimed at expediting and refining the training process of neural networks. It employs fundamental backpropagation principles to compute gradients and modify weights, while incorporating supplementary optimization techniques to enhance training efficiency and stability. The Levenberg-Marquardt (LM) algorithm is frequently utilized for training neural networks, particularly for nonlinear least-squares issues. This represents a synthesis of two algorithms, the Gradient Descent and Gauss-Newton methods, which are weight adjustment strategies employed to minimize error and enhance accuracy, although they may become unstable when distant from the optimal point. The LM algorithm reconciles gradient descent with Gauss-Newton methods, modifying the present solution to reduce inaccuracy. FFNNs have layers that link input to output and may accommodate finite input-output mapping challenges with a single hidden layer and an adequate number of neurons.

This paper develops a three-layer FFNN to determine the model structure and order for estimating the reference angle signal. The FFNN structure is designed and illustrated in Fig. 4.6. The complex nonlinear mapping from sEMG to joint angles uses the nonlinear tansig function for the midlayer and the linear purelin function for the output layer as transfer functions. The feedforward network is among the most often employed varieties of ANN. The architecture of a FFNN is nonlinear, with the output generated from the input using a feedforward arrangement. The multi-layer perceptron (MLP) is a type of FFNN consisting of input, hidden, and output layers. The MLP utilizes the back-propagation

(BP) learning mechanism [48]. The activation (or transfer) function, $f(x)$, facilitates the link between a node's input and output inside a network. The subsequent are categories of activation functions, purelin or linear activation functions, written mathematically as:

$$f(x) = x \quad (3.4)$$

Tansig activation function, given as

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (3.5)$$

3.4 Control Method for the Prosthetic Leg

The design steps involve system identification from the Euler-Lagrangian method, defining a sliding surface $S(\theta, \dot{\theta})$, designing the SMC control law to ensure that the sliding surface is reached and maintained. Set the control gains k and λ to balance tracking accuracy and robustness, and implement the STSMC control law to further enhance performance and reduce chattering, tune the parameters, α and β . The system is then simulated in MATLAB/Simulink to evaluate tracking performance in various scenarios, such as walking and stair climbing. Finally fine-tuning the control gains to minimize the tracking error and ensure stability based on the Lyapunov stability theorem. The overall mathematical principles to be used in the process are described below.

System dynamics can be described by the following formula. The system dynamics parameters $M(\theta)$, $C(\theta, \dot{\theta})$ and $G(\theta)$ must be determined in the system modeling.

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau \quad (3.6)$$

SMC control law, which has an equivalent and switching part, can be determined using the formula below.

$$\tau = M(\theta) (\ddot{\theta}_d - \lambda \dot{e}) + C(\theta, \dot{\theta})\dot{\theta}_d + G(\theta) - k \cdot \text{sign}(S) \quad (3.7)$$

ST-SMC Control Law has its own control law, which can be described below, and the stability of the system must be described.

$$\tau = -k_1 \cdot \text{sign}(S) - \int k_2 \cdot \text{sign}(S) dt \quad (3.8)$$

The sliding surface for SMC control can be described below. The derivative of the sliding surface determines the dynamics of the sliding surface

$$S = \dot{e} + \lambda e \quad (3.9)$$

In this system, the overall process is that the sEMG signals collected from the skin surface above the user's muscles are filtered and preprocessed in order to make the signal clearer. Then, using these signals, the NN tries to interpret the given signals and produces a reference signal reflecting the desired prosthetic limb movement. Then the signal is fed to the Sliding Mode Controller (SMC) to provide accurate control in such a way that the movement becomes smooth and natural. This approach enhances functionality and feel in prosthetics by offering the user better control and responsiveness.

Flowchart of the Process includes:

- sEMG Signal Acquisition from a proven data source.
- Signal processing of the sEMG signal (noise filtering, rectification, linerization).
- Feedforward neural network training by using the Levenberg-Marquardt algorithm, whose input is the sEMG signal and the outputs are joint angles.
- Control the Prosthetic leg (Using a SMC and STSMC control algorithm).

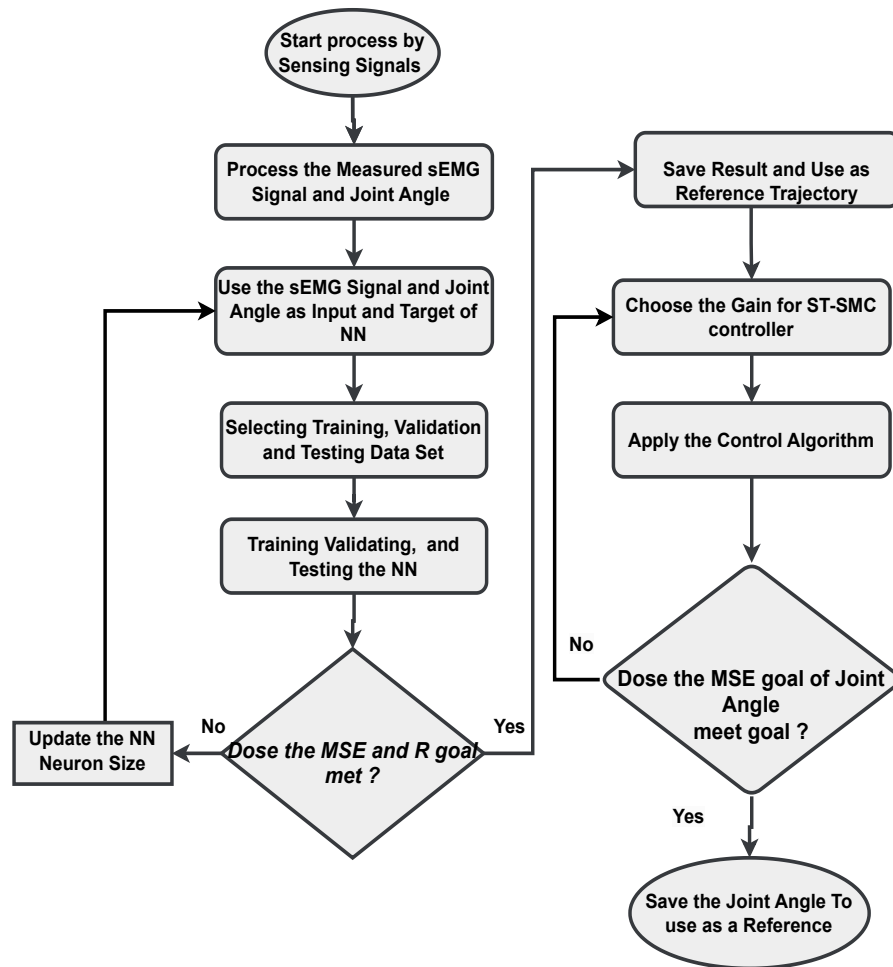


Figure 3.1 Flow chart of the NN-based lower limb prosthetic controller design process.

Chapter 4

System Modeling

4.1 Kinematic Modeling and Calculation

Kinematic modeling is the analysis of motion independent of the factors that induce it. Kinematics in robotics and biomechanics examines the correlation between joint angles, link lengths, and the spatial position and orientation of the end effector (e.g., foot or hand). Kinematic modeling aims to delineate the motion of a system or process, generally employing joint coordinates and link parameters.

4.2 Forward Kinematics

The task involves determining the end-effector's position and orientation using joint parameters, utilizing a linkage mechanism with a cartesian coordinate system at each link. The relationship between each link is depicted by a homogeneous transformation matrix using the DH approach [49]. Fig. 4.1 illustrates the coordinate system of the exoskeleton robot legs designed for lower limb rehabilitation. The hip, knee, and ankle joints possess a rotating degree of mobility. The specifications of the connecting rods are presented. The forward kinematics solution involves determining the location and orientation of the end effector's coordinate system within the base coordinate system, given the known joint displacements or angles. Each homogeneous transformation A_i is expressed as a product of four fundamental transformations. The transformation matrix of the connecting rod i in relation to the adjacent link $i - 1$ is given below.

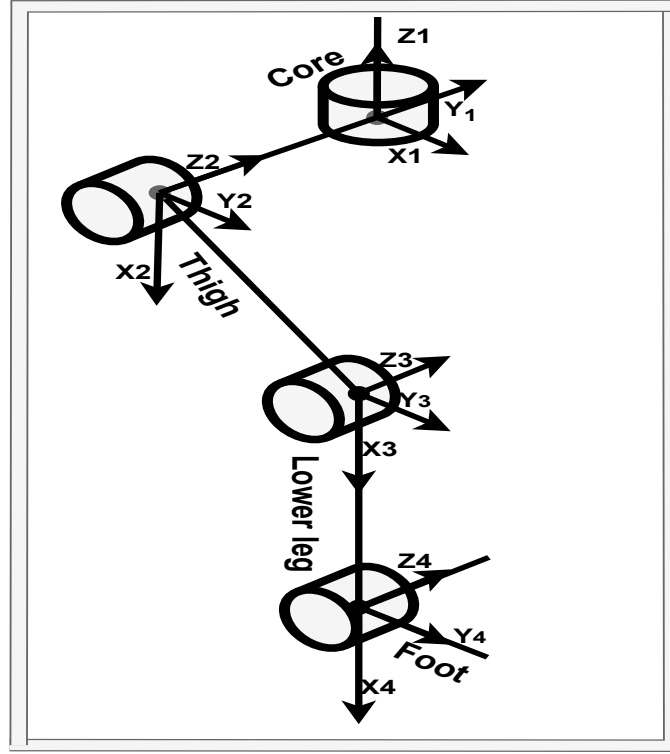


Figure 4.1 Kinematics for DH-parameter Analysis.

There are the following transformation matrices between lower limb linkage coordinate systems:

$${}^{i-1}_i Y = \text{Rot}_{z, \theta_i} \text{Trans}_{z, d_i} \text{Trans}_{x, \alpha_i} \text{Rot}_{x, \alpha_i} \quad (4.1)$$

$$= \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & \alpha_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & \alpha_i \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{i-1}_i Y = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\sin \alpha_i \cos \theta_i & -a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

where θ_i is the rotation angle between X_i and X_{i-1} measured about Z_i , α_{i-1} is the twist the angle between lines along joints $i-1$ and i measured about common perpendicular X , a_{i-1} is the link length between the lines along joints $i-1$ and i along common perpendicular, and d_i is the link offset along Z from the line parallel to X_i to the line parallel to and X_{i-1} .

The above parameters are denavit-hartenberg (DH) parameters. The DH for a three-link prosthesis is shown in Table 4.1. The homogeneous transformation from one link to another is achieved by multiplying a continuous sequence of matrices connecting the two links.

Joint	θ_i	d_i	a_i	α_i
1	θ_1	d_1	0	$\frac{\pi}{2}$
2	θ_2	d_2	a_1	0
3	θ_3	0	a_2	0
4	θ_4	0	a_3	0

Table 4.1 DH Parameters.

Using equation 4.2 and substituting the dh parameters in Table 4.1 results.

$$\begin{aligned}
{}^0_1Y &= \begin{bmatrix} -s\theta_1 & 0 & c\theta_1 & 0 \\ c\theta_1 & c\theta_1 & s\theta_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^1_2Y &= \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & a_1c\theta_2 \\ s\theta_2 & c\theta_2 & 0 & a_1s\theta_2 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^2_3Y &= \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & a_2c\theta_3 \\ s\theta_3 & c\theta_3 & 0 & a_2s\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^3_4Y &= \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & a_3c\theta_4 \\ s\theta_4 & c\theta_4 & 0 & a_3s\theta_4 \\ 0 & 0 & 1 & a_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

The transformation matrix of a coordinate system K can be derived by multiplication in the coordinate system J .

$${}^j_kY = \prod_{i=j}^{k-1} {}^i_{j+1}Y \quad (4.3)$$

Consequently, the end of the prosthesis's position and pose matrix is:

$${}^0_4Y = {}^0_1Y {}^1_2Y {}^2_3Y {}^3_4Y \quad (4.4)$$

$${}^0_4Y = \begin{bmatrix} c\theta_1c\theta_{234} & -s\theta_{234}c\theta_1 & s\theta_1 & d_2s\theta_1 + a_1c\theta_1c\theta_2 + a_2c\theta_1c\theta_2c\theta_3 - a_2c\theta_1s\theta_2s\theta_3 \\ s\theta_1c\theta_{234} & -s\theta_1s\theta_{234} & -c\theta_1 & a_1s\theta_1c\theta_2 - d_2c\theta_1 + a_2s\theta_1c\theta_2c\theta_3 - a_2s\theta_1s\theta_2s\theta_3 \\ s\theta_{234} & c\theta_{234} & 0 & a_1s\theta_2 + a_1c\theta_2s\theta_3 + a_2s\theta_2c\theta_3 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

$${}^0_4Y = {}^0_1Y {}^1_2Y {}^2_3Y {}^3_4Y = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.6)$$

where

$$n_x = c\theta_1c\theta_{234}, n_y = s\theta_1c\theta_{234}, n_z = s\theta_{234}$$

$$o_x = -s\theta_{234}c\theta_1, o_y = -s\theta_1s\theta_{234}, o_z = c\theta_{234}$$

$$a_x = s\theta_1, a_y = -c\theta_1, a_z = 0$$

$$p_x = d_2s\theta_1 + a_1c\theta_1c\theta_2 + a_2c\theta_1c\theta_2c\theta_3 - a_2c\theta_1s\theta_2s\theta_3$$

$$p_y = a_1s\theta_1c\theta_2 - d_2c\theta_1 + a_2s\theta_1c\theta_2c\theta_3 - a_2s\theta_1s\theta_2s\theta_3$$

$$p_z = a_1s\theta_2 + a_1c\theta_2s\theta_3 + a_2s\theta_2c\theta_3 + d_1$$

In these equations: $c\theta_1 = \cos \theta_1, s\theta_1 = \sin \theta_1, c\theta_2 = \cos \theta_2, s\theta_2 = \sin \theta_2, c\theta_3 = \cos \theta_3, s\theta_3 = \sin \theta_3, c\theta_4 = \cos \theta_4, s\theta_4 = \sin \theta_4$.

4.3 Inverse Kinematics Analysis

Inverse kinematics analysis involves determining the necessary joint parameters (angles or locations) to accomplish a certain position and orientation of the end-effector [50]. This problem is frequently more intricate than forward kinematics due to the potential for many solutions or the absence of solutions, contingent upon the structure and configuration of the prosthesis. Considering Fig. (4.2) for the derivation of θ_1 .

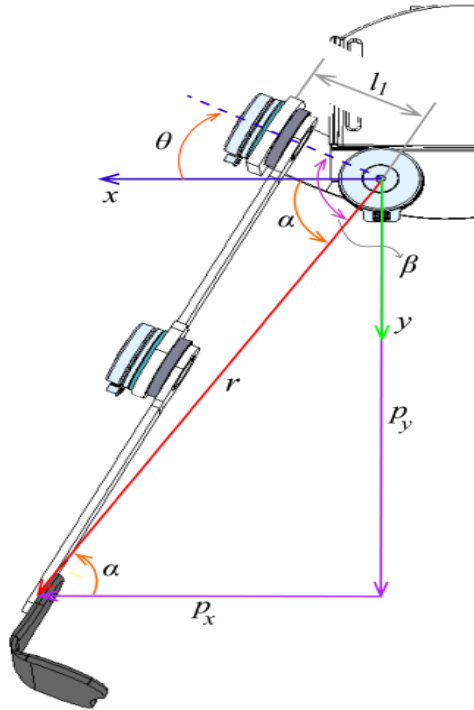


Figure 4.2 Inverse kinematics ₁

Implementing simple geometrical relationships to obtain the parameters below for the derivation of θ_1 from Fig. 4.2.

$$\begin{aligned} r^2 &= p_x^2 + p_y^2 \\ \alpha &= \tan^{-1} \frac{p_y}{p_x} \\ \beta &= \cos^{-1} \left(\frac{l_1}{r} \right) \end{aligned} \quad (4.7)$$

Since $\theta_1 = \beta - \alpha$ we obtain

$$\theta_1 = \cos^{-1} \left(\frac{l_1}{\sqrt{p_x^2 + p_y^2}} \right) - \tan^{-1} \left(\frac{p_y}{p_x} \right) \quad (4.8)$$

Applying the Pythagoras theorem in Fig. 4.4 to find the angle at the knee.

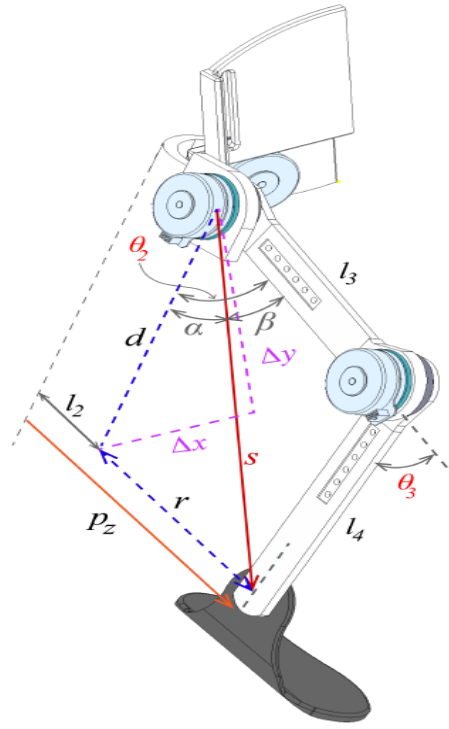


Figure 4.3 Inverse kinematics θ_3 .

$$d^2 + r^2 = l_3^2 + l_4^2 + 2l_3l_4 \cos \theta_3 \quad (4.9)$$

The variables in (4.9) can be using a geometric relationship as:

$$\begin{aligned} r &= P_z - l_2 \\ d^2 &= \Delta^2x + \Delta^2y \end{aligned} \quad (4.10)$$

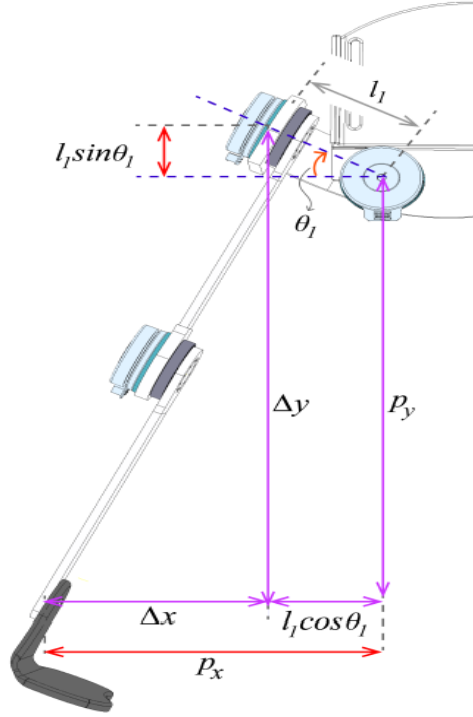
Furthermore, from Fig. 4.4.

$$\begin{aligned} \Delta x &= P_x - l_1 \cos \theta_1 \\ \Delta y &= P_y + l_1 \sin \theta_1 \end{aligned} \quad (4.11)$$

Hence, substituting (4.10) and (4.11) in to (4.9) θ_3 can be described as:

$$\theta_3 = \cos^{-1} \left(\frac{(\Delta x^2) + (\Delta y^2) + (P_z - l_2)^2 - l_3^2 - l_4^2}{2l_3l_4} \right) \quad (4.12)$$

Next calculating for θ_2

Figure 4.4 Inverse kinematics θ_2 .

$$\theta_2 = \alpha + \beta$$

$$\alpha = \sin^{-1} \left(\frac{r}{s} \right)$$

$$\beta = \sin^{-1} \left(\frac{l_4 \sin \theta_3}{s} \right)$$

where $s = \sqrt{d^2 + r^2}$

we obtain

$$\theta_2 = \sin^{-1} \left(\frac{(P_z - l_2)}{\sqrt{d^2 + r^2}} \right) + \sin^{-1} \left(\frac{l_4 \sin \theta_3}{\sqrt{d^2 + r^2}} \right) \quad (4.13)$$

where,

$$\sqrt{d^2 + r^2} = \sqrt{l_3^2 + l_4^2 + 2l_3l_4 \cos \theta_3}$$

4.3.1 Velocity Analysis

The Jacobian matrix is crucial for understanding the relationship between prosthetic foot velocity and each joint speed, representing the speed transformation ratio from joint space to operating space. The relationship is given as:

$$\dot{x} = \begin{bmatrix} v \\ w \end{bmatrix} = J(\dot{q}) \quad (4.14)$$

where $\dot{x}$ represents the end-effector velocity state, v_n represents the end-effector's linear velocity of origin, ω_n represents its angular velocity, and $\dot{q}$ represents the joint velocities.

$$\dot{x} = \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} J_{l_1} & J_{l_2} & J_{l_3} \\ J_{a_1} & J_{a_2} & J_{a_3} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \quad (4.15)$$

A Jacobian matrix is a $m \times n$ matrix where m represents the degrees of freedom of the end-effector (foot) space and n the degrees of freedom of the mechanism (actuated joint variables). For a n joint prosthesis, the Jacobian matrix can be written as:

$$J = \begin{bmatrix} J_1 & J_2 & J_3 \dots J_n \end{bmatrix} \quad (4.16)$$

Furthermore, a revolute joint J_i is given by

$$J_i = \begin{bmatrix} z_{i-1}^0 \times P_n^{i-1} \end{bmatrix} \quad (4.17)$$

To implement (4.17), we must determine each joint axis's position and orientation. If the DH protocol (4.6) is used ${}^0z_{i-1}$ and ${}^{i-1}p_n$ can be determined as:

$${}^0z_{i-1} = {}^0Y_{i-1}(1:3,3) \quad (4.18)$$

$${}^{i-1}p_n = {}^0Y_n(1:3,3) - {}^0Y_{i-1}(1:3,3) \quad (4.19)$$

From (4.18) and (4.19) the following matrix elements are obtained.

$$\begin{aligned} z_{i-1}^0 &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \dots, z_1^0 = z_2^0 = \begin{bmatrix} \cos \theta_1 \\ \sin \theta_1 \\ 0 \end{bmatrix} \\ P_3^0 &= \begin{bmatrix} l_1 c \theta_1 - l_3 c \theta_2 s \theta_1 - l_4 c \theta_2 c \theta_3 s \theta_1 + l_4 s \theta_1 s \theta_2 s \theta_3 \\ l_1 s \theta_1 + l_3 c \theta_1 c \theta_2 + l_4 c \theta_1 c \theta_2 c \theta_3 - l_4 c \theta_1 s \theta_2 s \theta_3 \\ l_2 + l_4 s \theta_{23} + l_3 s \theta_2 \end{bmatrix} \\ P_3^1 &= \begin{bmatrix} l_1 c \theta_1 - l_3 c \theta_2 s \theta_1 - l_4 c \theta_2 c \theta_3 s \theta_1 + l_4 s \theta_1 s \theta_2 s \theta_3 \\ l_1 s \theta_1 + l_3 c \theta_1 c \theta_2 + l_4 c \theta_1 c \theta_2 c \theta_3 - l_4 c \theta_1 s \theta_2 s \theta_3 \\ l_4 s \theta_{23} + l_3 s \theta_2 \end{bmatrix} \\ P_3^2 &= \begin{bmatrix} -l_4 c \theta_{23} s \theta_1 \\ l_4 c \theta_{23} c \theta_1 \\ l_4 s \theta_{23} \end{bmatrix} \end{aligned} \quad (4.20)$$

where $\theta_{23} = \theta_2 + \theta_3$

$$J = \begin{bmatrix} x_{11} & -s\theta_1 (l_4 s\theta_{23} + l_3 s\theta_2) & l_4 s\theta_{23} s\theta_1 & c\theta_1 \\ x_{12} & -c\theta_1 (l_4 s\theta_{23} + l_3 s\theta_2) & -l_4 s\theta_{23} c\theta_1 & -c\theta_1 (l_4 s\theta_{23} + l_3 s\theta_2) \\ 0 & l_4 c\theta_{23} + l_3 c\theta_2 & l_4 c\theta_{23} & x_{12} \\ 0 & c\theta_1 & c\theta_1 & s\theta_1 \\ 0 & s\theta_1 & s\theta_1 & l_4 c\theta_{23} \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (4.21)$$

$$x_{11} = -l_1 s\theta_1 - l_3 c\theta_1 c\theta_2 + l_4 c\theta_1 s\theta_2 s\theta_3 - l_4 c\theta_1 c\theta_2 c\theta_3 \quad (4.22)$$

$$x_{12} = l_1 c\theta_1 - l_3 c\theta_2 s\theta_1 - l_4 c\theta_2 c\theta_3 s\theta_1 + l_4 s\theta_1 s\theta_2 s\theta_3$$

Hence, given the joint velocities $\dot{q} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3]$, the end-effector velocities, v_n and ω_n can be computed from

$$\begin{bmatrix} v_n \\ \omega_n \end{bmatrix} = J \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \quad (4.23)$$

4.4 Dynamic Modeling and Calculation

The prosthesis model, designed to capture the dynamics of human lower limb movements, is precisely constructed using the Euler-Lagrange formulation [51]. This mathematical framework is used for the system shown in Fig. 4.5, consisting of three lower limb segments (thigh, lower leg, and foot) linked by three joints (hip, knee, and ankle). Each segment of the lower leg is represented as a stiff bar with distinct properties. The properties include length L_n , mass m_n , and the proximal center of gravity l_n . The segments are oriented at angular positions θ_n in relation to the vertical axis. The Lagrangian for the system is:

$$L = K - P \quad (4.24)$$

where K and P are kinetic and potential energy, respectively.

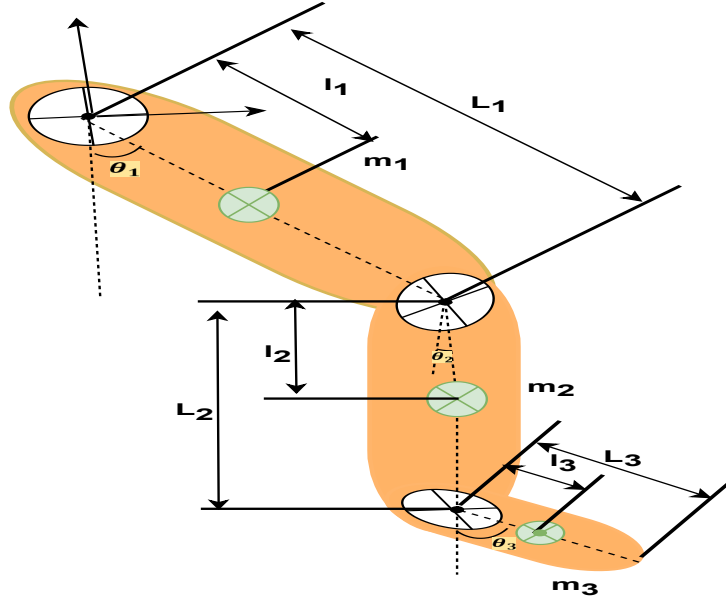


Figure 4.5 Human lower limb for dynamic modeling

The kinetic energy K (4.25) and potential energy of the leg link P (4.29) are expressed as follows: variable v_n in (4.25) denotes linear velocity which are given in (4.27), whereas J_n symbolizes the moment of inertia, and $\dot{\theta}_n$ represents angular velocity.

$$K = \frac{m_1 v_1^2}{2} + \frac{J_1 \dot{\theta}_1^2}{2} + \frac{m_2 v_2^2}{2} + \frac{J_2 \dot{\theta}_2^2}{2} + \frac{m_3 v_3^2}{2} + \frac{J_3 \dot{\theta}_3^2}{2} \quad (4.25)$$

The linear velocity is defined as the magnitude of the first derivative of the position vector r_n of the center of gravity for each segment $v_n = |\dot{r}_n|$. The centers of gravity for the thigh, lower leg, and foot are located at coordinates r_1 , r_2 , and r_3 , respectively, which are given as:

$$\begin{aligned} r_1 &= \begin{bmatrix} l_1 \sin \theta_1 \\ -l_1 \cos \theta_1 \end{bmatrix} \\ r_2 &= \begin{bmatrix} L_1 \sin \theta_1 + l_2 \sin \theta_2 \\ -L_1 \cos \theta_1 - l_2 \cos \theta_2 \end{bmatrix} \\ r_3 &= \begin{bmatrix} L_1 \sin \theta_1 + L_2 \sin \theta_2 + l_3 \sin \theta_3 \\ -L_1 \cos \theta_1 - L_2 \cos \theta_2 - l_3 \cos \theta_3 \end{bmatrix} \end{aligned} \quad (4.26)$$

Consequently, the linear velocities of the thigh, lower leg, and foot are defined as:

$$\begin{aligned} v_1 &= \dot{\theta}_1 l_1 \\ v_2 &= \sqrt{\dot{\theta}_1^2 L_1^2 + \dot{\theta}_2^2 l_2^2 + 2\dot{\theta}_1 \dot{\theta}_2 L_1 l_2 \cos(\theta_1 - \theta_2)} \\ v_3 &= \sqrt{\begin{aligned} &\dot{\theta}_1^2 L_1^2 + \dot{\theta}_2^2 L_2^2 + \dot{\theta}_3^2 l_3^2 + \\ &2\dot{\theta}_1 \dot{\theta}_2 L_1 L_2 \cos(\theta_1 - \theta_2) + 2\dot{\theta}_2 \dot{\theta}_3 L_2 l_3 \cos(\theta_2 - \theta_3) \\ &+ 2\dot{\theta}_1 \dot{\theta}_3 L_1 l_3 \cos(\theta_1 - \theta_3) \end{aligned}} \end{aligned} \quad (4.27)$$

Inserting (4.27) into (4.25) yields (4.28) for the system's kinetic energy.

$$\begin{aligned}
 K = & \frac{m_1 \dot{\theta}_1^2 l_1^2}{2} + \frac{J_1 \dot{\theta}_1^2}{2} + \frac{m_2 \dot{\theta}_1^2 L_1^2}{2} + \frac{m_2 \dot{\theta}_2^2 l_2^2}{2} + \frac{J_2 \dot{\theta}_2^2}{2} + m_2 \dot{\theta}_1 \dot{\theta}_2 L_1 l_2 \cos(\theta_1 - \theta_2) \\
 & + \frac{m_3 \dot{\theta}_1^2 L_1^2}{2} + \frac{m_3 \dot{\theta}_2^2 L_2^2}{2} + \frac{m_2 \dot{\theta}_3^2 l_3^2}{2} + m_3 \dot{\theta}_1 \dot{\theta}_2 L_1 L_2 \cos(\theta_1 - \theta_2) \\
 & + m_3 \dot{\theta}_2 \dot{\theta}_3 L_2 l_3 \cos(\theta_2 - \theta_3) + m_3 \dot{\theta}_1 \dot{\theta}_3 L_1 l_3 \cos(\theta_1 - \theta_3) + \frac{J_3 \dot{\theta}_3^2}{2}
 \end{aligned} \quad (4.28)$$

The potential energy is delineated by (4.29).

$$P = m_1 g h_1 + m_2 g h_2 + m_3 g h_3 \quad (4.29)$$

In this equation, g signifies gravity and h_n indicates the height of the center of gravity in relation to the coordinate system.

The height of the center of gravity for the thigh, lower leg, and foot may be articulated in relation to the angular displacement as follows:

$$\begin{aligned}
 h_1 &= -l_1 \cos \theta_1 \\
 h_2 &= -L_1 \cos \theta_1 - l_2 \cos \theta_2 \\
 h_3 &= -L_1 \cos \theta_1 - L_2 \cos \theta_2 - l_3 \cos \theta_3
 \end{aligned} \quad (4.30)$$

Inserting (4.30) into (4.29) yields (4.31) for the system's potential energy.

$$\begin{aligned}
 P = & -m_1 g l_1 \cos \theta_1 - m_2 g L_1 \cos \theta_1 - m_2 g l_2 \cos \theta_2 - m_3 g L_1 \cos \theta_1 \\
 & - m_3 g L_2 \cos \theta_2 - m_3 g l_3 \cos \theta_3
 \end{aligned} \quad (4.31)$$

Substituting (4.28) and (4.31) into (4.24), the Lagrangian of the system is obtained as follows:

$$\begin{aligned}
 L = & \frac{m_1 \dot{\theta}_1^2 l_1^2}{2} + \frac{J_1 \dot{\theta}_1^2}{2} + \frac{m_2 \dot{\theta}_1^2 L_1^2}{2} + \frac{m_2 \dot{\theta}_2^2 l_2^2}{2} + \frac{J_2 \dot{\theta}_2^2}{2} + m_2 \dot{\theta}_1 \dot{\theta}_2 L_1 l_2 \cos(\theta_1 - \theta_2) \\
 & + \frac{m_3 \dot{\theta}_1^2 L_1^2}{2} + \frac{m_3 \dot{\theta}_2^2 L_2^2}{2} + \frac{m_2 \dot{\theta}_3^2 l_3^2}{2} + m_3 \dot{\theta}_1 \dot{\theta}_2 L_1 L_2 \cos(\theta_1 - \theta_2) + \frac{J_3 \dot{\theta}_3^2}{2} \\
 & + m_3 \dot{\theta}_2 \dot{\theta}_3 L_2 l_3 \cos(\theta_2 - \theta_3) + m_3 \dot{\theta}_1 \dot{\theta}_3 L_1 l_3 \cos(\theta_1 - \theta_3) + m g l_1 \cos \theta_1 \\
 & + m_2 g L_1 \cos \theta_1 + m_2 g l_2 \cos \theta_2 + m_3 g L_1 \cos \theta_1 + m_3 g L_2 \cos \theta_2 + m_3 g l_3 \cos \theta_3
 \end{aligned} \quad (4.32)$$

The Euler-Lagrange formulation is defined by (4.33) from the lagrangian equation (4.24), where k denotes the degree of freedom, q_k represents the set of generalized coordinates, and Q_k represents the set of external (non-conservative) forces acting on the system. The torque T_n is applied at each joint, and the viscous damping is denoted by the coefficient b_n . The Euler-Lagrange formulation is derived for the hip, knee, and ankle in (4.34), (4.35)

and (4.36) respectively,

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_k} \right) - \frac{\partial L}{\partial \theta_k} = Q_k \quad (4.33)$$

where $k = 1, 2, 3$ for the three joints.

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_1} \right) - \frac{\partial L}{\partial \theta_1} = \tau_1 - b_1 \dot{\theta}_1 - b_2 (\dot{\theta}_1 - \dot{\theta}_2) \quad (4.34)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_2} \right) - \frac{\partial L}{\partial \theta_2} = \tau_2 - b_2 (\dot{\theta}_2 - \dot{\theta}_1) - b_3 (\dot{\theta}_2 - \dot{\theta}_3) \quad (4.35)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_3} \right) - \frac{\partial L}{\partial \theta_3} = \tau_3 - b_3 (\dot{\theta}_3 - \dot{\theta}_2) \quad (4.36)$$

The aforementioned equations provide the nonlinear second-order differential equations (4.37), (4.38), and (4.39), which characterize the motion of the thigh, lower leg, and foot, respectively.

$$\begin{aligned} & \ddot{\theta}_1 (m_1 l_1^2 + m_2 L_1^2 + m_3 L_1^2 + J_1) + \ddot{\theta}_2 [m_2 L_1 l_2 \cos(\theta_1 - \theta_2)] + m_3 L_1 l_2 \cos(\theta_1 - \theta_2) \\ & + \ddot{\theta}_3 [m_3 L_1 l_3 \cos(\theta_1 - \theta_3)] + \dot{\theta}_2^2 [m_2 L_1 l_2 \sin(\theta_1 - \theta_2) + m_3 L_1 L_2 \sin(\theta_1 - \theta_2)] \\ & + \dot{\theta}_3^2 [m_3 L_1 l_3 \sin(\theta_1 - \theta_3)] + m_1 g l_1 \sin \theta_1 \\ & + m_2 g L_1 \sin \theta_1 + m_3 g L_1 \sin \theta_1 \\ & = \tau_1 - b_1 \dot{\theta}_1 - b_2 (\dot{\theta}_1 - \dot{\theta}_2) \end{aligned} \quad (4.37)$$

$$\begin{aligned} & \ddot{\theta}_2 (m_2 l_2^2 + m_3 L_2^2 + J_2) + \ddot{\theta}_1 [m_2 L_1 l_2 \cos(\theta_1 - \theta_2)] + m_3 L_1 l_2 \cos(\theta_1 - \theta_2) \\ & + \ddot{\theta}_3 [m_3 L_2 l_3 \cos(\theta_2 - \theta_3)] + \dot{\theta}_1^2 [-m_2 L_1 l_2 \sin(\theta_1 - \theta_2) - m_3 L_1 L_2 \sin(\theta_1 - \theta_2)] \\ & + \dot{\theta}_3^2 [m_3 L_2 l_3 \sin(\theta_2 - \theta_3)] + m_2 g l_2 \sin \theta_2 + m_3 g L_2 \sin \theta_2 \\ & = \tau_2 - b_2 (\dot{\theta}_2 - \dot{\theta}_1) - b_3 (\dot{\theta}_2 - \dot{\theta}_3) \end{aligned} \quad (4.38)$$

$$\begin{aligned} & \ddot{\theta}_3 (m_3 l_3^2 + J_3) + \ddot{\theta}_2 [m_3 L_2 l_3 \cos(\theta_2 - \theta_3)] + \ddot{\theta}_1 [m_3 L_1 l_3 \cos(\theta_1 - \theta_3)] \\ & + \dot{\theta}_2^2 [-m_3 L_2 l_3 \sin(\theta_2 - \theta_3)] + \dot{\theta}_1^2 [-m_3 L_1 l_3 \sin(\theta_1 - \theta_3)] + m_2 g l_3 \sin \theta_3 \\ & = \tau_3 - b_3 (\dot{\theta}_3 - \dot{\theta}_2) \end{aligned} \quad (4.39)$$

To simplify differential equations (4.37), (4.38) and (4.39), certain parameters are consolidated into the following constants of Table 4.2.

$c_1 = m_1 * l_1^2 + m_2 * L_1^2 + m_3 * L_1^2 + J_1$	$c_8 = m_2 * l_2^2 + m_3 * L_2^2 + J_2$
$c_2 = m_2 * L_1 * L_2$	$c_9 = m_3 * L_2 * L_3$
$c_3 = m_3 * L_1 * L_2$	$c_{10} = m_2 * g * L_2$
$c_4 = m_3 * L_1 * L_3$	$c_{11} = m_3 * g * L_2$
$c_5 = m_1 * g * l_1$	$c_{12} = m_3 * l_3^2 + J_3$
$c_6 = m_2 * g * L_1$	$c_{13} = m_3 * g * l_3$
$c_7 = m_3 * g * L_1$	

Table 4.2 Constants for model simplification.

As a result of these simplifications, (4.40), (4.41), and (4.42) are obtained.

$$\begin{aligned}
 & \tau_1 - b_1 \dot{\theta}_1 - b_2 (\dot{\theta}_1 - \dot{\theta}_2) \\
 &= c_1 \ddot{\theta}_1 + (c_2 + c_3) \ddot{\theta}_2 \cos(\theta_1 - \theta_2) + c_4 \ddot{\theta}_3 \cos(\theta_1 - \theta_3) \\
 &+ (c_2 + c_3) \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) + c_4 \dot{\theta}_3^2 \sin(\theta_1 - \theta_3) + c_5 \sin \theta_1 \\
 &+ c_6 \sin \theta_1 + c_7 \sin \theta_1
 \end{aligned} \tag{4.40}$$

$$\begin{aligned}
 & \tau_2 - b_2 (\dot{\theta}_2 - \dot{\theta}_1) - b_3 (\dot{\theta}_2 - \dot{\theta}_3) \\
 &= c_8 \ddot{\theta}_2 + (c_2 + c_3) \ddot{\theta}_1 \cos(\theta_1 - \theta_2) + c_9 \ddot{\theta}_3 \cos(\theta_2 - \theta_3) \\
 &- (c_2 + c_3) \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) + c_9 \dot{\theta}_3^2 \sin(\theta_2 - \theta_3) + c_{10} \sin \theta_2 \\
 &+ c_{11} \sin \theta_2
 \end{aligned} \tag{4.41}$$

$$\begin{aligned}
 & \tau_3 - b_3 (\dot{\theta}_3 - \dot{\theta}_1) \\
 &= c_{12} \ddot{\theta}_3 + c_9 \ddot{\theta}_2 \cos(\theta_2 - \theta_3) + c_4 \ddot{\theta}_1 \cos(\theta_1 - \theta_3) \\
 &- c_9 \dot{\theta}_2^2 \sin(\theta_2 - \theta_3) - c_4 \dot{\theta}_1^2 \sin(\theta_1 - \theta_3) + c_{13} \sin \theta_3
 \end{aligned} \tag{4.42}$$

Equations (4.40), (4.41), and (4.42) are solved for the highest order derivatives $\ddot{\theta}_1$, $\ddot{\theta}_2$, and $\ddot{\theta}_3$ respectively, to obtain the following set of motion equations:

$$\begin{aligned}
 \ddot{\theta}_1 = \frac{1}{c_1} [& - (c_2 + c_3) \ddot{\theta}_2 \cos(\theta_1 - \theta_2) - c_4 \ddot{\theta}_3 \cos(\theta_1 - \theta_3) \\
 & - (c_2 + c_3) \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) - c_4 \dot{\theta}_3^2 \sin(\theta_1 - \theta_3) \\
 & - (c_5 + c_6 + c_7) \sin \theta_1 + \tau_1 - b_1 \dot{\theta}_1 - b_2 (\dot{\theta}_1 - \dot{\theta}_2)]
 \end{aligned} \tag{4.43}$$

$$\begin{aligned}
 \ddot{\theta}_2 = \frac{1}{c_8} [& - (c_2 + c_3) \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - c_9 \ddot{\theta}_3 \cos(\theta_2 - \theta_3) \\
 & + (c_2 + c_3) \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) - c_9 \dot{\theta}_3^2 \sin(\theta_2 - \theta_3) \\
 & - (c_{10} + c_{11}) \sin \theta_2 + \tau_2 - b_2 (\dot{\theta}_2 - \dot{\theta}_1) - b_3 (\dot{\theta}_2 - \dot{\theta}_3)]
 \end{aligned} \tag{4.44}$$

$$\begin{aligned}
 \ddot{\theta}_3 = \frac{1}{c_{12}} [& - c_9 \ddot{\theta}_2 \cos(\theta_2 - \theta_3) - c_4 \ddot{\theta}_1 \cos(\theta_1 - \theta_3) \\
 & + c_9 \dot{\theta}_2^2 \sin(\theta_2 - \theta_3) + c_4 \dot{\theta}_1^2 \sin(\theta_1 - \theta_3) - c_{13} \sin \theta_3 \\
 & + \tau_3 - b_3 (\dot{\theta}_3 - \dot{\theta}_1)]
 \end{aligned} \tag{4.45}$$

The plant model of human lower limb motion in the sagittal plane was executed in MATLAB Simulink, using the non-linear second-order differential equations Eq. 4.45 as seen in the block diagram in Appendix.

4.5 sEMG Data Collection and Preparation

Muscle group identification of the corresponding lower limb movement is necessary to train the neural network for better hip flexion, knee flexion, and ankle flexion performance [52], [53]. Nine muscles given in Table 6.3 are divided into three categories, which will be responsible for the three joints given below.

- Hip

Movement on the hip is primarily affected by the following muscle groups: Tensor-fascialata Weak hip abduction and hip rotation. The functions of the rectus femoris are to flex the thigh at the hip joint and extend the leg at the knee joint. Whereas the semimembranosus extends the hip joint and flexes the knee joint.

- Knee

At the knee Medialgastrocnemius is responsible for plantar flexion of the foot. Vastus medialis works with the other quadriceps muscles to help extend the knee joint. Lastly Lateral gastrocnemius functions include plantar flexion of the foot, assisting in knee flexion.

- Ankle

The muscles at the ankle include the upper tibialis anterior; it acts to dorsiflex and invert the foot. Additionally, the lower tibialis anterior supports the main foot dorsiflexor. Finally, Soleus is responsible, foot down.

sEMG Signal Analysis

SENIAM guidelines for surface EMG include a high-pass filter with a cutoff frequency of $10Hz - 20Hz$ and a low-pass filter with a cutoff frequency approaching $500Hz$ in most instances [54]. The Nyquist rate is twice the frequency that must be recorded accurately. The theorem can be applied inversely. The Nyquist frequency is the maximum frequency that a device with a specified sample rate can accurately measure, defined as half of that sample rate. The low-frequency cutoff of the bandpass filter eliminates baseline drift, which is sometimes linked to movement, and eradicates any DC offset. The standard range for the low-frequency cutoff is $5Hz$ to $20Hz$. Suppose the mean value of the signal is not zero prior to high-pass or band-pass filtering. In that case, it will be rendered zero or nearly zero afterward, as these filters eliminate low-frequency components of a signal. The high-frequency cutoff of a band-pass filter eliminates high-frequency noise and prevents aliasing in the recorded signal. The high-frequency cutoff must be sufficiently elevated to ensure that quick on-off bursts of the EMG remain distinctly recognizable. Standard values range from $200Hz$ to $1kHz$. Signal processing is performed using the following equations: low-pass, high-pass, rectification, linearization, and feature extraction using MVA. All the required parameters and the results are discussed in Table 6.3.

Low-pass filters are used to remove high-frequency noise from the sEMG signal. The transfer function can describe a general first-order low-pass filter as:

$$H(s) = \frac{\omega_c}{s + \omega_c} \quad (4.46)$$

where s is the complex frequency variable and ω_c is the cutoff frequency.

In the time domain (for discrete signals), the low-pass filter is:

$$y[n] = \alpha x[n] + (1 - \alpha)y[n - 1] \quad (4.47)$$

where $y[n]$ is the filtered signal, $x[n]$ is the input signal. α is a constant determined by the cutoff frequency and sample rate $\alpha = \frac{T}{T+RC}$ (where T is the sampling interval and RC is a time constant).

High-pass filters are used to remove low-frequency noise, such as motion artifacts, from sEMG signals. A general first-order high-pass filter is:

$$H(s) = \frac{s}{s + \omega_c} \quad (4.48)$$

where ω_c is the cutoff frequency. In the discrete domain:

$$y[n] = \alpha(y[n - 1] + x[n] - x[n - 1]) \quad (4.49)$$

where $\alpha = \frac{RC}{RC+T}$ (RC is the time constant and T is the sampling interval, $x[n]$ and $x[n - 1]$ are the current and previous input samples, $y[n]$ is the filtered output).

The full-wave rectified sEMG signal can be described mathematically as:

$$y[n] = |x[n]| \quad (4.50)$$

where $y[n]$ is the rectified sEMG signal and $x[n]$ is the raw sEMG signal at time n .

MAV for Walking Gait Feature Extraction

The mean absolute value (MAV) is a commonly used feature in electromyography (EMG) signal processing, especially for tasks like muscle activity detection and pattern recognition. MAV provides a simple measure of the average muscle activity in a given time window, capturing the signal's intensity and amplitude. In analyzing sEMG signals for a human gait cycle, the choice of the time window for feature extraction is critical for accurately capturing the muscle activation patterns associated with walking phases. For Mean Absolute Value (MAV), a common time window range is 100 to 250(*ms*). The sEMG signal is split into overlapping windows with a 50% overlap to make sure there is continuity between phases. However, this may need to be changed depending on the study's goal and the type of gait cycle analysis being done. For the human walking gait cycle, the corresponding time windows are listed in Table 6.4. The equation below describes the

MAV analysis.

$$MAV_i = \frac{1}{N} \sum_{k=1}^N |sEMG(i)| \quad (4.51)$$

where N is the number of samples in a chosen time window. This method provides a linear estimate of the signal's amplitude over time.

The signal analyses are done on Matlab Simulink to obtain the processed sEMG signals in the result section.

Joint Angle Signal Processing

A Savitzky–Golay filter is a digital filter utilized on a collection of digital data points to smooth the data, hence enhancing its precision without altering the signal [55]. This is accomplished by a procedure called convolution, which involves fitting consecutive subsets of neighboring data points with a low-degree polynomial using the linear least squares method. When data points are uniformly distributed, an analytical solution to the least-squares equations can be derived, represented by a singular set of "convolution coefficients" applicable to all data subsets, yielding estimates of the smoothed signal (or its derivatives) at the central point of each subset. For a set of input points x_n , the smoothed value y_n is given by:

$$y_n = \sum_{k=-m}^m c_k \cdot x_{n+k} \quad (4.52)$$

where x_n is the input signal at point n , c_k are the filter coefficients, and m is half the window length (the total window length is $2m + 1$).

The coefficients are determined by fitting a polynomial of degree d to the windowed data points. These coefficients are calculated using a least-squares approach, ensuring the polynomial minimizes the fitting error for the data points within the window.

4.6 FFNN for Angle Estimation

The processed EMG signals are used as inputs for the FFNN, and the measured joint angles are used as targets to train the network so that it can guess the joint reference angle that goes with the sensed sEMG input. Determining the input, output, and hidden layers is a crucial part of the design. The input layer corresponds to the number of features extracted from the sEMG signals. In this paper, 3 features are extracted from 9 muscles; thus, the input layer will have 36 neurons. The hidden layers must contain layers with a sufficient number of neurons to capture the complex mapping between the sEMG features and joint angles. Typically, 1 – 2 hidden layers with 50 – 100 neurons are sufficient. For this design, 1 hidden layer with 50 neurons is implemented. The output layer will have 3 neurons corresponding to the predicted joint angles (hip, knee, and ankle). Tansing activation functions, an activation function for hidden layers, and a linear activation for the

output layer are implemented. For training the neural network, a training algorithm called the Levenberg-Marquardt (LM) backpropagation algorithm is used, which is efficient for small-to-medium-sized datasets and provides fast convergence [56]. The data is split into training (70%), validation (15%), and test (15%) sets to evaluate performance.

The process of training the network starts by feeding the input sEMG feature vectors and the corresponding joint angles. The weights are updated iteratively based on the back-propagation algorithm, and the training stops when the validation error reaches a minimum or a preset number of epochs is reached. To analyze the performance of the NN, a mean squared error (MSE) method is used, which determines the difference between the predicted and actual joint angles used. Regression of the output and the target is performed, which indicates the best training for a regression value close to 1. The suggested multilayer perceptron network is illustrated in Fig. 4.6. As shown in the figure, the proposed FFNN network comprises three nodes x_i for $1 \leq i \leq 3$ at the input layer, one hidden layer, and one node at the output layer. The adjustable weight sets are denoted by w_{ij} and u_{jk} where w_{ij} joins the i_{th} input node to the j_{th} neuron in the hidden layer, while the j_{th} output in the hidden layer is connected to the k_{th} node in the output layer by u_{jk} ; y_k ($k = 1$ in this case) is the output layer. whereas b_k ($k = 1$) is linked with the node in layer 2, b_j is associated with the j th neuron in the first network layer (i.e., layer 1). The neuron and the output node are therefore coupled to changeable bias values b_j and b_k . Every network layer contains the weights w , the summing and multiplication operators, the bias b , and the activation functions layer 1 is expressed as:

$$y_k = \left(\sum_{i=1}^3 w_{ij} x_i + b_j \right) \quad (4.53)$$

where y_k is the output layer, while k is the kt neuron in the output layer, but since $k = 1$, If y_k represents the output layer and k represents the kt neuron in the output layer; nevertheless, given that $k = 1$, $y_k = y_1$. Layer 2 can be written as:

$$y_1 = \varphi_2 \left(\sum_{j=1}^n u_{j1} \varphi_1 \left(\sum_{i=1}^3 w_{ij} x_i + b_j \right) + b_1 \right) \quad (4.54)$$

where n represents the total count of neurons in the hidden layer and y_1 denotes the output layer. Consequently, each layer N is further defined by the network.

$$y_1 = \varphi_N \left(\sum_{i=1}^v w_{kl} \dots \frac{\varphi_2 \left(\sum_{j=1}^n u_{jk} \frac{\varphi_1 \left(\sum_{i=1}^m \frac{w_{ij} x_j + b_j}{\text{layer1}} \right)}{\text{layer2}} + b_k \right)}{\text{layer N}} + \dots + b_1 \right) \quad (4.55)$$

Where v represents the maximum quantity of neurons in layer N , 1 denotes the index of the l_{th} neuron in layer N , and N signifies the total number of layers in the network.

Employing layer 2 with a linear activation function in both the hidden and output layers, with n neurons in the hidden layer and the corresponding layer and input weights multiplied, (4.55) is subsequently reformulated as:

$$\begin{aligned} y &= [H^*G]x_i + [H^*b_1] + b_2 \\ y &= H[G^*x_i + b_1] + b_2 \end{aligned} \quad (4.56)$$

where H = is layer weight = $[1 \times n]$ double matrix, G = is input weight = $[n \times 3]$ double matrix, x_i = is input vector = $\begin{bmatrix} 3 & x & 1 \end{bmatrix}$ matrix (sEMG signals at each joint), b_1 = is layer 1 bias = $[n \times 1]$ double matrix, and b_2 = is layer 2 bias = constant (c).

Therefore, if

$$[H * G] = \begin{bmatrix} \partial & \lambda & \rho \end{bmatrix} \quad (4.57)$$

Then

$$[H * G] * X_i = \begin{bmatrix} \partial & \lambda & \rho \end{bmatrix} * \begin{bmatrix} 3 \times 1 \end{bmatrix} = \begin{bmatrix} \partial & \lambda & \rho \end{bmatrix}^* \begin{bmatrix} sEMG_1 \\ sEMG_2 \\ sEMG_3 \end{bmatrix} \quad (4.58)$$

The equation for the proposed model is presented as follows,

$$y = \partial sEMG_1 + \lambda sEMG_2 + \rho sEMG_3 + \mu + c \quad (4.59)$$

where $\partial; \lambda; \rho; \mu$ and c are constant values.

The proposed model equation for the Tansig activation function in the hidden layer and the linear activation function in the output layer would be expressed utilizing the same approaches

$$y = \partial \left(\frac{2}{1 + \exp -2(\lambda x_i + b)} \right) + c \quad (4.60)$$

Filtration of the neural network's input data is required due to the aforementioned causes.

- High-quality training data is crucial as it significantly influences the model's performance.
- Noisy data may adversely impact backpropagation, potentially compromising its outcomes.
- Training backpropagation models can be time-consuming and may need significant effort to optimize their performance.

- Backpropagation necessitates a matrix-oriented methodology, perhaps resulting in additional complications.

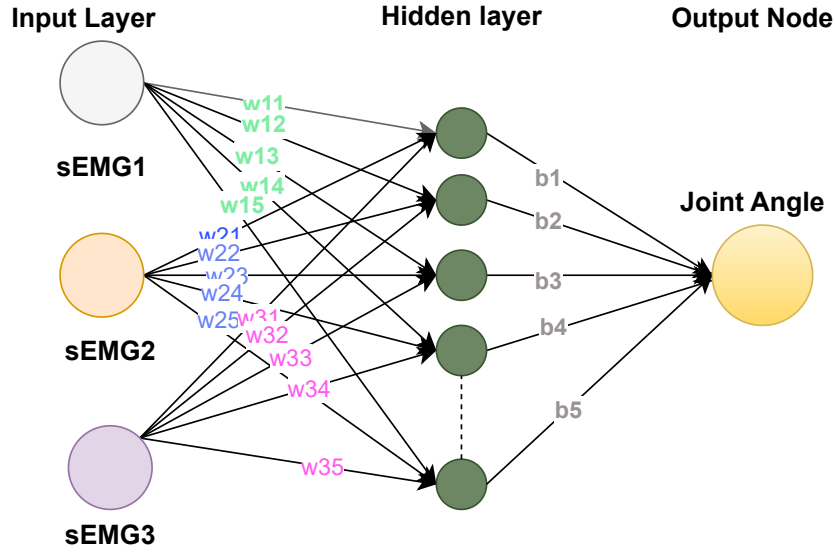


Figure 4.6 FF-Neural network.

Fig. 4.6 Block diagram of the NN. The muscles responsible for hip flexion are the tensor fasciae latae and rectus femoris, which function to flex the thigh at the hip joint and extend the leg at the knee joint. The sEMG signals are input into the neural network to train it, with the target being the signal obtained from measuring the hip angle.

A neural network training algorithm called Levenberg-Marquardt (LM) is used. The aim of the algorithm is to minimize a nonlinear least squares problem by iteration, which is the error between the target outputs and predicted outputs, by adjusting the network's weights. It combines the gradient descent approach and Gauss-Newton algorithm which makes it faster and more robust for small or medium data sets. The aim is to minimize the error between the target outputs and predicted outputs by adjusting the network's weights. For this, we define the error function as:

$$E(w) = \frac{1}{2} \sum_{i=1}^N (y_i - f(x_i; w))^2 \quad (4.61)$$

where $E(w)$ is the error to be minimized, w represents the network's weights, y_i is the target output, $f(x_i; w)$ is the neural network's output for input x_i , and N is the number of data points.

The LM algorithm iteratively updates the weights w using the following update rule:

$$w_{k+1} = w_k - [J^T J + \lambda I]^{-1} J^T e \quad (4.62)$$

where w_k is the weight vector at iteration k , J is the Jacobian matrix of partial derivatives

of the network errors with respect to the weights (i.e. $J = \frac{\partial e}{\partial w}$), e is the vector of errors at the current weights (i.e. $e = y - f(x; w_k)$), I is the identity matrix, and λ is a damping factor.

The algorithm's behavior is influenced by the damping factor λ , which is adjusted during the training process. A large λ results in a gradient descent-like behavior, while a small λ resembles a Gauss-Newton method. The damping factor decreases when the error decreases, allowing faster convergence, and increases when the error increases.

The neural network training outcomes were assessed utilizing the mean square error (MSE). The mean squared error (MSE) is defined as:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - x_i)^2 \quad (4.63)$$

where y_i represents the anticipated value, x_i represents the observed value, and N represents the quantity of data points.

Chapter 5

CONTROLLER DESIGN

Sliding mode control (SMC) is a prevalent nonlinear robust control methodology. Its resilience to modeling uncertainty renders it unique [57]. The fundamental concept of SMC is to compel the system's states to attain and traverse a predetermined sliding surface. Upon reaching this surface, the system demonstrates favorable characteristics, including stability, robustness, and tracking. The control law is non-continuous and transitions swiftly to maintain the system on the sliding surface. Owing to its resilient characteristics, it is extensively utilized for the regulation of robotic manipulators, autonomous robots, and power system control applications.

The primary disadvantage of SMC is chattering. The SMC's structure incorporates a relay-like function, resulting in the control action alternating between two extremes. As a result of the supplementary control effort, subsequent phases necessitate further control actions to counteract the prior dynamic effects induced by the additional control measures, perpetuating the process. The primary limitation of SMC is chattering. Chattering results in erratic motion, auditory disturbance, and energy depletion. To eliminate this problem, numerous chattering suppression strategies are already accessible [58], including the sigmoid function, non-linear saturation function, higher-order sliding mode, and ST-SMC, which are widely utilized. ST-SMC is elucidated in [59], [60], [61], [62] and [63]. Numerous chatter suppression strategies are already accessible [58], including the sigmoid function, non-linear saturation function, higher-order sliding mode, and super twisting control, which are widely utilized. This study focuses on the design of standard SMC and ST-SMC.

5.1 State equation of the system

The subsequent discussion pertains to state-space modeling and the development of a conventional SMC and a ST-SMC. The dynamics of the prosthetic can be represented as:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) + \tau_d = \tau \quad (5.1)$$

which can be written as:

$$\ddot{\theta} = -M^{-1} [C(\theta, \dot{\theta}) + G(\theta) + \tau_d - \tau] \quad (5.2)$$

where τ_d is an external disturbance on the system.

$$\begin{aligned} \tau_d &= \begin{bmatrix} \tau_{d1} \\ \tau_{d2} \\ \tau_{d3} \end{bmatrix} \in \mathbb{R}^{3 \times 1} & M(\theta) &= \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \in \mathbb{R}^{3 \times 3} \\ C(\theta, \dot{\theta}) &= \begin{bmatrix} C_1(\theta, \dot{\theta}) \\ C_2(\theta, \dot{\theta}) \\ C_3(\theta, \dot{\theta}) \end{bmatrix} \in \mathbb{R}^{3 \times 1} \text{ and } & G(\theta) &= \begin{bmatrix} G_1(\theta) \\ G_2(\theta) \\ G_3(\theta) \end{bmatrix} \in \mathbb{R}^{3 \times 1} \end{aligned} \quad (5.3)$$

In the above equation, the prosthetic mass matrix is derived as follows:

$$M_{3 \times 3}(q) = \sum_{i=1}^2 \left(m_i J_{v_i}^T J_{v_i} + J_{\omega_i}^C I_i J_{\omega_i} \right) \quad (5.4)$$

$$\begin{aligned} M_{11} &= m_1 l_1^2 + m_2 (L_1^2 + l_2^2 + 2L_1 l_2 c_2) + m_3 (L_1^2 + L_2^2 l_3^2 + 2L_1 L_2 c_2 + 2L_1 l_3 c_{23} \\ &\quad + 2L_2 l_3 c_3) + I_1 + I_2 + I_3 \\ M_{12} &= m_2 (l_2^2 + L_1 l_2 c_2) + m_3 (L_2^2 + l_3^2 + L_1 L_2 c_2 + L_1 l_3 c_{23} + 2L_2 l_3 c_3) + I_2 + I_3 \\ M_{13} &= m_3 (l_3^2 + L_1 l_3 c_{23} + L_2 l_3 c_3) + I_3, \quad M_{22} = m_2 l_2^2 + m_3 (L_2^2 + l_3^2 + 2L_2 l_3 c_3) + I_2 + I_3 \\ M_{23} &= m_3 (l_3^2 + L_2 l_3 c_3) + I_3, \quad M_{33} = m_3 l_3^2 + I_3 \end{aligned}$$

The coriolis and centrifugal vectors are derived as:

$$C(\theta, \dot{\theta}) = \dot{M}(\theta) \dot{\theta} - \frac{1}{2} \frac{\partial}{\partial \theta} (\dot{\theta}^T M(\theta) \dot{\theta}) \quad (5.5)$$

$$\begin{aligned} C_1 &= -m_2 L_1 l_2 (2\dot{\theta}_1 + \dot{\theta}_2) s_2 \dot{\theta}_2 - m_3 [L_1 L_2 (2\dot{\theta}_1 + \dot{\theta}_2) s_2 + L_1 l_3 (2\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) s_{23} \\ &\quad + L_2 l_3 (2\dot{\theta}_1 + 2\dot{\theta}_2 + \dot{\theta}_3) s_3] \\ C_2 &= -m_2 [L_1 l_2 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) s_2 + L_1 l_2 \dot{\theta}_1 s_2 \dot{\theta}_2] - m_3 [L_1 L_2 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) \\ &\quad + L_1 l_3 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_1 \dot{\theta}_3) s_{23} + L_2 l_3 (\dot{\theta}_1^2 + 2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_1 \dot{\theta}_3 + \dot{\theta}_2^2 + \dot{\theta}_2 \dot{\theta}_3) s_3 \\ &\quad + L_1 L_2 \dot{\theta}_1 s_2 \dot{\theta}_2 + L_1 l_3 \dot{\theta}_1 \dot{\theta}_2 s_{23} + L_2 l_3 (2\dot{\theta}_1 + 2\dot{\theta}_2 + \dot{\theta}_3) s_3] \\ C_3 &= -m_3 [L_1 l_3 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_1 \dot{\theta}_3) s_{23} + L_2 l_3 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_1 \dot{\theta}_3 + \dot{\theta}_2^2 + \dot{\theta}_2 \dot{\theta}_3) s_3 \\ &\quad + L_1 l_3 \dot{\theta}_1 \dot{\theta}_3 s_{23} + L_2 l_3 (\dot{\theta}_1 + \dot{\theta}_2) s_3 \dot{\theta}_3] \end{aligned}$$

The gravity vector is derived as follows:

$$\begin{aligned} G_1(\theta) &= g [c_1 (m_1 l_1 + m_2 L_1 + m_3 L_1) + c_{12} (m_2 l_2 + m_3 L_2) + (m_3 l_3)] \\ G_2(\theta) &= g [(m_2 l_2 + m_3 L_2) c_{12} + m_3 l_3 c_{123}] \\ G_3(\theta) &= g (m_3 l_3 c_{123}) \end{aligned}$$

In the above expressions, m_i is the mass of the links, L_i is the length of the links, l_i is the center of mass location, I_i is the moment of inertia of the links, $c_{ij} = \cos \theta_i + \cos \theta_j$, $s_{ij} = \sin \theta_i + \sin \theta_j$ and g is gravitational acceleration (for i and $j = 1, 2, 3$ at each joint).

Defining the state vector for system dynamics as:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \quad (5.6)$$

Let the state of the system be $x_1 = \theta$ and $x_2 = \dot{x}_1 = \dot{\theta}$ then the state equation is:

$$\begin{aligned} \dot{x}_1 &= \dot{\theta} \\ \dot{x}_2 &= -M^{-1} [C(\theta, \dot{\theta}) + G(\theta) + \tau_d - \tau] \end{aligned} \quad (5.7)$$

Thus, the state equation of the system is:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x) + g(x)\tau + d \end{cases} \quad (5.8)$$

where

$$f(x) = -M^{-1} [C(\theta, \dot{\theta}) + G(\theta)] \quad (5.9)$$

$$g(x) = M^{-1}, d = -M^{-1}\tau_d \quad (5.10)$$

$$\tau = g^{-1}(x)[-f(x) + d] \quad (5.11)$$

Then, the control target is x_1 to track x_{1d} , where τ is the control input matrix, and τ_d represents uncertain external factors.

The control law is:

$$\tau = g^{-1}(x)[-f(x) + d + u] \quad (5.12)$$

The state-space form of the equation becomes

$$\dot{x}_1 = x_2 \quad (5.13)$$

$$\dot{x}_2 = u \quad (5.14)$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u \quad (5.15)$$

$$\dot{X} = AX + Bu \quad (5.16)$$

$$Y = CX + Du \quad (5.17)$$

where, $X \in \mathbb{R}^{6 \times 1}$. The first 3 states represent the position of the joints.

$$\begin{aligned} A &= \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{6 \times 6} & B &= \begin{bmatrix} 0 \\ I \end{bmatrix} \in \mathbb{R}^{6 \times 3} \\ C &= [I] \in \mathbb{R}^{6 \times 6} & D &= [0] \in \mathbb{R}^{6 \times 3} \end{aligned} \quad (5.18)$$

The control input will be described as

$$u = f(x) + g(x)\tau - d \quad (5.19)$$

5.2 SMC Controller Design

SMC is a control methodology comprising two phases. The reaching phase, in which the controller guides the system's states to the sliding surface, and the sliding phase, during which the system sustains its state on the surface. The sliding mode dynamics define the system's behavior, rendering it exceptionally resilient. To ascertain the sliding mode dynamics, the system must satisfy specific requirements and execute differentiation. The control law guarantees that the system attains the sliding surface within a given time frame.

5.2.1 Conventional SMC Controller Design

The control law guarantees that the system attains the sliding surface within a given time frame. The overall control law for sliding mode control involves two parts: equivalent control (τ_{eq}) and switching control (τ_{sw}).

$$\tau = \tau_{eq} + \tau_{sw} \quad (5.20)$$

The equivalent control τ_{eq} is the smooth control action that maintains the system on the sliding surface. It assumes the system is perfectly on the sliding surface $S(t) = 0$. It can be found by solving $\dot{S}(t) = 0$.

$$\tau_{eq} = g^{-1}(x)[-f(x) + d] \quad (5.21)$$

The switching control τ_{sw} is designed to drive the system states to the sliding surface. It uses a discontinuous control law, often based on the sign function.

$$\tau_{sw} = -k_{1n} \text{sign}(s) \quad (5.22)$$

where k_1 is a positive gain matrix and $n = 1, 2, 3$. corresponds to each of the three joints. To ensure robustness against model uncertainties and external disturbances, we choose the gain k to be sufficiently high. To design the controller, first the sliding surface needs to be

defined. Then the prostatic tracking error and its derivatives can be defined as:

$$e(t) = \theta_d(t) - \theta(t) \quad (5.23)$$

$$\dot{e}(t) = \dot{\theta}_d(t) - \dot{\theta}(t) \quad (5.24)$$

$$\ddot{e}(t) = \ddot{\theta}_d(t) - \ddot{\theta}(t) \quad (5.25)$$

where θ_d , $\dot{\theta}_d$ and $\ddot{\theta}_d(t)$ are the desired positions, velocities, and accelerations.

Choosing a sliding surface $S(\tilde{x})$ that represents the desired system behavior where a common choice is $S = S(\tilde{x}) = \lambda \tilde{x}_1 + I\tilde{x}_2 = 0$ where $\tilde{x}_1 = e = \theta_d(t) - \theta(t)$ and λ a positive constant.

$$S = S(\tilde{x}) = \lambda \tilde{x}_1 + I\tilde{x}_2 = 0 \quad (5.26)$$

Derivation equation (5.26) gives a is a second-order system that ensures asymptotic stability for any positive λ .

$$\tilde{x}_2 = -\lambda \tilde{x}_1 \quad (5.27)$$

$$\tilde{x}_1 = -\lambda \tilde{x}_1 \quad (5.28)$$

The error dynamics, as defined by (5.27) and (5.28), are indicative of the system dynamics in a sliding surface, which is typically a well-behaved and stable system for any $\lambda > 0$. The switching surface's λ value is the sole determinant of the system. The system is not influenced by parameter uncertainties during the sliding phase. The control law is intended to maintain $\dot{S} = 0$ in the presence of disturbances.

Then to derive the equivalent control and switching control, taking the derivative of the sliding surface in (5.28).

$$\dot{S} = \tilde{x} = \lambda \tilde{x}_2 + I\dot{\tilde{x}}_2 \quad (5.29)$$

$$\dot{x}_2 = \ddot{e} = \ddot{\theta}_d - f(x) - g(x)\tau - d \quad (5.30)$$

where I is an (3×3) identity matrix, $f(x)$, $g(x)$, d and τ are (3×3) matrix. Substituting (5.30) in to (5.29).

$$\dot{S} = s(\tilde{x}) = \lambda \dot{e} + I(\ddot{\theta}_d - f(x) - g(x)\tau - d) \quad (5.31)$$

$$\tau_{eq} = -g(x)^{-1} (f(x) + \dot{x}_{2d} + \lambda (\dot{x}_{1d} - \dot{x}_1) + d) \quad (5.32)$$

$$s(\tilde{x}) = \lambda_1 \tilde{x}_1 + I\tilde{x}_2 = \lambda_1 (\theta_d - \theta) + (\dot{\theta}_d - \dot{\theta}) \quad (5.33)$$

where $S(\tilde{x}) \in \mathbb{R}^{3 \times 1}$ $\tilde{x}_1$ joint tracking errors, $\tilde{x}_2$ joint tracking velocity errors, λ is a (3×3) diagonal matrix, I is an (3×3) identity matrix.

The switching control given in (5.34) is designed to drive the system states to the sliding surface.

$$\dot{S} = -k_{1n} \text{sign}(s) \quad (5.34)$$

The fundamental principle of Lyapunov's theory is the notion of a Lyapunov function, $V(x)$ which is an energy like function assisting in ascertaining if the system maintains a desirable condition or trajectory over time. A prevalent selection for Lyapunov function is: $V = \frac{1}{2}S^T S$.

The properties of a Lyapunov function is that, $V(x)$ is continuously differentiable $V(x) > 0$ for all $X \neq x_0$ and $V(x_0) = 0$ positive definite. The time derivative of $V(x)$ is denoted as $\frac{d}{dt}(V) = \dot{V}(x)$.

Choosing the Lyapunov function in (5.35) the objective is to ensure $\dot{V}(x) < 0$ which guarantees the attainment and maintenance of the sliding surface.

$$V(s) = \frac{1}{2}S^T S \quad (5.35)$$

Differentiating (5.35) with respect to time results.

$$\frac{d}{dt}(V) = \frac{1}{2} \frac{d}{dt} (S^T S) = S^T \dot{S} < 0 \quad (5.36)$$

For reaching phase $\dot{S} = k_{1n} \text{sign}(s)$ then

$$S^T \dot{S} = S * k_{1n} \text{sign}(s) < 0 \quad (5.37)$$

$$S^T \dot{S} = -k_{1n} |s| < 0 \quad (5.38)$$

where $k_{1n} \in \mathbb{R}^{3 \times 3}$ the diagonal matrix, the gain $K_{1n} > 0$ should be set in such a way that $S^T \dot{S} < 0$ remains valid.

The faster convergence to the sliding surface is guaranteed by the higher value of the gain K_{1n} . Thus, as long as $S(t) \neq 0$ the control law forces the system state to move toward the sliding surface. Once the system reaches the sliding surface, $S = 0$ it enters the sliding phase. During this phase, the system dynamics are governed by the sliding surface equation. The control law ensures that the system stays on the surface and the Lyapunov function remains constant $\dot{V} = 0$. Thus, the system is stable and tracks the desired behavior defined by the sliding surface.

The overall control structure is given as:

$$\tau = \tau_{eq} + \tau_{sw} \quad (5.39)$$

$$\begin{aligned} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} &= \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} c_1 \dot{e}_1 \\ c_2 \dot{e}_2 \\ c_3 \dot{e}_3 \end{bmatrix} + \begin{bmatrix} C_1(\theta, \dot{\theta}) \\ C_2(\theta, \dot{\theta}) \\ C_3(\theta, \dot{\theta}) \end{bmatrix} \begin{bmatrix} G_1(\theta) \\ G_2(\theta) \\ G_3(\theta) \end{bmatrix} - \begin{bmatrix} \tau_{d1} \\ \tau_{d2} \\ \tau_{d3} \end{bmatrix} \\ &+ \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} k_{11} \text{sign}(s) & 0 & 0 \\ 0 & k_{12} \text{sign}(s) & 0 \\ 0 & 0 & k_{13} \text{sign}(s) \end{bmatrix} \end{aligned} \quad (5.40)$$

5.3 Super Twisting Sliding Mode Control (ST-SMC)

The discrete switching nature of conventional SMC, attributed to the relay-like function $\text{sign}(S)$, results in oscillations of the system near the sliding mode surface, causing chattering during the latter phase of trajectory tracking. Despite the implementation of the constant velocity reaching law, the system's chattering persists, resulting in diminished tracking performance. Furthermore, in the presence of an external disturbance to the system, stability cannot be assured for a finite duration. Consequently, the second-order sliding mode super spiral method [61] is utilized to enhance the system's robustness. ST-SMC is a modification of SMC in which the control input is engineered to be continuous, leading to gradual control maneuvers rather than sudden transitions and minimized chattering, especially adjacent to the sliding surface. System resilience to exogenous perturbations. The control rule is now founded on the Super Twisting function instead of the conventional discontinuous sign function, which facilitates a smooth convergence of the state to the sliding surface.

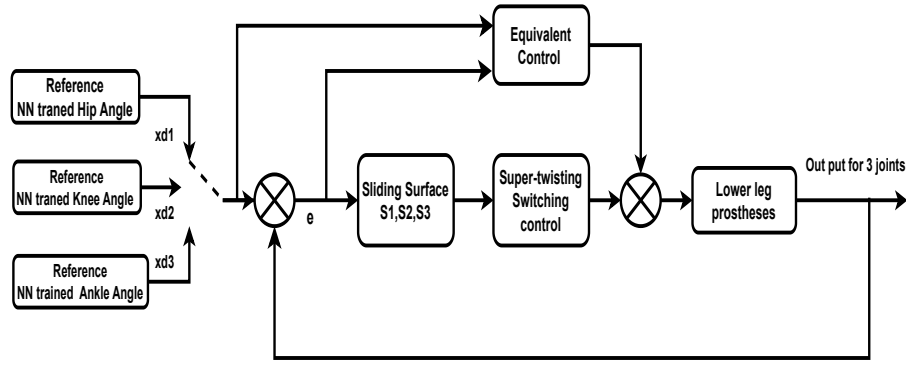


Figure 5.1 ST-SMC Block diagram

$$\begin{aligned} u_{st} &= -k_4 |S|^\delta \text{sign}(S) + u_{aux} \\ \dot{u}_{aux} &= -k_3 \text{sign}(S) \end{aligned} \quad (5.41)$$

The block diagram Fig. 5.1 illustrates how the feedback loop uses the NN-trained joint angles as a reference, with the output joint angles representing the instantaneous measured joint angle. The inner loop displays the ST-SMC controller, which incorporates both equivalent and super-twisting control inputs to enforce the reference angles.

The identical design approach will be adhered to, with the exception that the switching input will be altered. To implement ST-SMC, we define the sliding mode switching surface function as follows:

$$S = x_1 - x_{1d} \quad (5.42)$$

The derivation of (5.42) is performed to obtain.

$$\dot{S} = x_2 - \dot{x}_{1d} \quad (5.43)$$

As the control signal of the prosthesis is absent in the first derivative of the sliding mode switch surface function, the derivative of the (5.43) is computed.

$$\ddot{S} = \dot{x}_2 - \ddot{x}_{1d} = f(x) + g(x)\tau + d - \ddot{x}_{1d} \quad (5.44)$$

If the finite time tracking of the system is considered, let $S = z_1$ and $\dot{S} = z_2$.

Then,

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = f(x) + g(s)\tau + d - \ddot{x}_{1d} \end{cases} \quad (5.45)$$

By forcing the sliding surface to zero $\dot{z}_2 = 0$ the feedback control of the system is:

$$\tau_{eq} = \frac{1}{g(x)} (-f(x) + \ddot{x}_{1d} - d) \quad (5.46)$$

while the final control law will be expressed as $\tau = \tau_{eq} + \tau_{st}$ the second-order ST-SMC algorithm is employed to achieve the robustness of the system.

Let

$$\begin{aligned} u_{st} &= -\alpha |S|^\delta \text{sign}(S) + u_{aux} \\ \dot{u}_{aux} &= -\beta \text{sign}(S) \end{aligned} \quad (5.47)$$

where u_{st} is a super-twisting function, and α and β are the super-twisting function control gains.

The adaptive law can be designed using the following rule:

$$\begin{cases} \dot{\hat{\alpha}} = a|S|^{\frac{1}{2}} + c \text{sign}(S) \dot{u}_{ast} \\ \dot{\hat{\beta}} = 2c|S|^{\frac{1}{2}} + 2b \text{sign}(S) \dot{u}_{ast} \end{cases} \quad (5.48)$$

where a, b , and c are constants. In addition, $P = \begin{bmatrix} a & c \\ c & b \end{bmatrix}$ should be a positive definite matrix. In (5.47) the super twisting input will be:

$$u_{st} = \left(-k_{2n} |s|^\delta \text{sign}(s) - \int_0^t k_{3n} \text{sign}(s(\tau)) d\tau \right) \quad (5.49)$$

where u_{st} is the control input, u_{aux} is the auxiliary input, k_{2n} and k_{3n} are positive control gains must be tuned to achieve the desired control performance and δ is fractional power that controls the rate of convergence.

For the design of super-twisting control law in this paper let's consider:

$$u_{st} = \left(-k_{2n} |s|^{\frac{1}{2}} \text{sign}(s) - \int_0^t k_{3n} \text{sign}(s(\tau)) d\tau \right) \quad (5.50)$$

$$\dot{S} = \frac{1}{g(x)} \left(-k_{2n} |s|^{\frac{1}{2}} \text{sign}(s) - \int_0^t k_{3n} \text{sign}(s(\tau)) d\tau \right) \quad (5.51)$$

Let $\alpha = \frac{1}{g(x)} (-k_{2n})$ and $\beta = \frac{1}{g(x)} (-k_{3n})$ (5.51) becomes:

$$\dot{s} = \left(-\alpha |s|^{\frac{1}{2}} \text{sign}(s) - \int_0^t \beta \text{sign}(s(\tau)) d\tau \right) \quad (5.52)$$

From (5.52) the auxiliary part of the super twisting input will be:

$$u_{\text{aux}} = \int_0^t \beta \text{sign}(s(\tau)) d\tau \quad (5.53)$$

Taking the derivation of (5.53) with respect to s.

$$\dot{u}_{\text{aux}} = -\beta \text{sign}(s) \quad (5.54)$$

The super twisting law for the system will be:

$$\dot{u} = -\alpha |s|^{\frac{1}{2}} \text{sign}(s) + u_{\text{aux}} \quad (5.55)$$

where u_{st} is adjusted continuously, reducing the chattering effect. The auxiliary variable u_{aux} provides a feedback mechanism that ensures smoother control action. The total control law of the system can be formulated as follows:

$$u = u_{eq} + u_{st} \quad (5.56)$$

where $u_{st} = \left(-\alpha |s|^{\frac{1}{2}} \text{sign}(s) - \int_0^t \beta \text{sign}(s(\tau)) d\tau \right)$

These finalize the ST-SMC controller design. The proof of the stability of the ST-SMC controller is described in the Appendix.

5.4 Dc Motor Modeling

The basic principle behind the working of a DC motor is that a current-carrying conductor placed in the magnetic field shows some force due to electromagnetic induction. A DC motor serves as the drive mechanism to control the movement of the prosthetic joint. This enables accurate and regulated movement in the leg. Electric DC motors are often used as actuators in minor load applications, including two-link and three-link leg models. Fig. 5.2 illustrates the model for the dc motor.

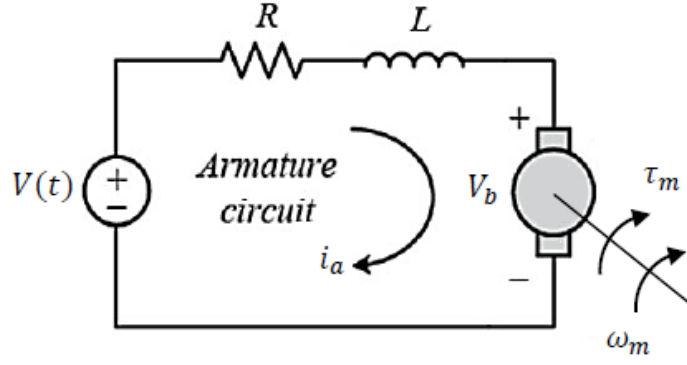
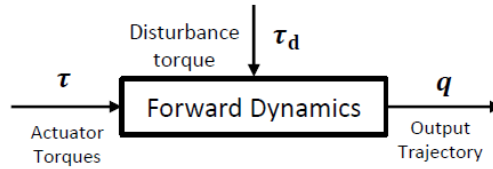


Figure 5.2 ST-SMC Block diagram

In forward dynamics, the output trajectory of the prosthetic leg is determined based on the applied actuator forces.

Figure 5.3 Applied actuator torque τ .

General dynamics of actuator is given as:

$$\tau + \tau_d = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + \mathbf{g}(q) \quad (5.57)$$

The dynamics equations will be integrated to determine the trajectory. The mass matrix is positive definite; hence, it is invertible in all configurations.

$$\ddot{q} = M^{-1}(q) \{ \tau + \tau_d - C(q, \dot{q})\dot{q} - \mathbf{g}(q) \} \quad (5.58)$$

Where, $i_a, R, L, V(t)$ and V_b are Dc motor armature current, armature resistance, armature inductance, armature terminal voltage and back electromotive force (emf) respectively, ω_m , and τ_m are motor speed and motor torque.

DC Motors operation principle of operation is as follows: A current carrying conductor in a magnetic field experiences a Force

$$F = i \times \phi \quad (5.59)$$

i is the current in the conductor. Then the torque generated in the motor is given as:

$$\tau_m = K_1 \phi i_a \quad (5.60)$$

where i_a is the armature current, ϕ is the magnetic field flux. Based on Lenz's Law the back-emf voltage is described as

$$V_b = K_2 \phi \omega_m \quad (5.61)$$

where ω_m is the angular velocity of the rotor, ϕ is the magnetic field flux. For a permanent magnet DC motor ϕ is constant thus $\tau_m = K_m i_a$ and $V_b = K_m \omega_m$ where K_m is the DC motor torque or velocity constant. Finally the electrical model of the PMDC motor using KVL will be expressed as:

$$L \frac{di_a}{dt} + Ri_a + V_b = V(t) \text{ or } (Ls + R)i_a = V - V_b \quad (5.62)$$

$$i_a = \frac{V - V_b}{(Ls + R)}$$

The electro-mechanical model of the PMDC motor substituting Eq.(5.62) into Eq.(5.60) is described as:

$$\tau_m = K_m i_a = \tau_m = K_m \frac{V - K_m \omega_m}{(Ls + R)} \quad (5.63)$$

Usually $\frac{L}{R} \ll 1$ ($\cong 10^{-2} - 10^{-3}$) then up to 100 – 1000 Hz bandwidth neglect induction results in

$$\tau_m = \frac{K_m}{R} (V - K_m \omega_m) \quad (5.64)$$

The Motor characteristic torque-velocity curve is linear. To avoid overheating i_a is clamped to $i_{\max}$ by current limiters. The stall torque τ_s is the maximum output torque of the motor, when the velocity of the motor is stalled.

$$V_s = \frac{R \tau_s}{K_m} \quad (5.65)$$

The maximum speed of the motor $\omega_{\max}$ is obtained when no load torque is applied to the motor. Back-emf acts like a disturbance on the output torque, the more the velocity the less the output torque. Electric motors provide low torque but have high velocity so that gear box is used to increase the torque while reducing the speed. Ideal gearbox with ratio η will have $\theta_m = \eta \theta$ and $\omega_m = \eta \omega$ while $\tau_m = \frac{1}{\eta} \tau$

By use principle of virtual work $\tau_m \cdot \theta_m = \tau \cdot \theta$ (motor - gearbox - load dynamics notation) where the subscript m is for motor side, with no subscript for load side. Suppose inertia I_m, I and viscous friction b_m, b Motor side dynamics and load side dynamics results in

$$\tau_m = I_m \ddot{\theta}_m + b_m \dot{\theta}_m + \frac{1}{\eta} \tau, \quad (5.66)$$

$$\tau = I \ddot{\theta} + b \dot{\theta}$$

Overall dynamics is described as:

$$\tau_m = I_m \ddot{\theta}_m + b_m \dot{\theta}_m + \frac{1}{\eta} (I \ddot{\theta} + b \dot{\theta}) \quad (5.67)$$

By substituting : $\dot{\theta}_m = \eta \dot{\theta}$ and $\ddot{\theta}_m = \eta \ddot{\theta}$ in Eq.(5.67) and considering $\tau_m = \frac{1}{\eta} \tau$ for motor side

$$\tau_m = \left(I_m + \frac{1}{\eta^2} I \right) \ddot{\theta}_m + \left(b_m + \frac{1}{\eta^2} b \right) \dot{\theta}_m \quad (5.68)$$

And at Load side

$$\tau = (\eta^2 I_m + I) \ddot{\theta} + (\eta^2 b_m + b) \dot{\theta} \quad (5.69)$$

For motor-gearbox mechanism with three DOFs prsostasis dynamics:

$$\tau = I_e \ddot{\theta} + b_e \dot{\theta} \quad (5.70)$$

The effect of motor inertia and damping is highlighted by a factor of η^2 , in which the effective inertia and viscous friction are $I_e = (\eta^2 I_m + I)$ and $b_e = (\eta^2 b_m + b)$ If the the general model of the robot is considered $\tau - b\dot{q} = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + \mathbf{g}(q)$ Then adding the motor - gearbox dynamics results in

$$\tau - b_e \dot{q} = M_e \ddot{q} + C(q, \dot{q})\dot{q} + \mathbf{g}(q) \quad (5.71)$$

For three degree of freedom $M_e(q)$ and b_e are described as:

$$M_e(q) = \left\{ M(q) + \begin{bmatrix} \eta_1^2 I_{m_1} & \dots & 0 \\ \vdots & \ddots & 0 \\ 0 & \dots & \eta_n^2 I_{m_n} \end{bmatrix} \right\} \text{ and } b_e = \begin{bmatrix} b_1 + \eta^2 b_{m_1} \\ \vdots \\ b_n + \eta^2 b_{m_n} \end{bmatrix} \quad (5.72)$$

The energy consumed by a joint over a period of time is the integral of power:

$$E = \int_{t_1}^{t_2} P(t) dt = \int_{t_1}^{t_2} \tau(t) \cdot \dot{\theta}(t) dt$$

where t_1 and t_2 are the start and end times of the motion. The angle angular velocity is described in the data set (SIAT-LLMD) [43]. The comparison on energy consumption between the SMC and ST-SMC is discussed in the result section.

Chapter 6

Result and Discussion

6.1 Model Validations

The plant model, which captures the dynamics of human lower limb movements, is constructed using the Euler-Lagrange formulation equation (3.3). This mathematical framework used for the system shown in Fig. 4.5, consisting of three lower limb segments (thigh, lower leg, and foot) joined by three joints (hip, knee, and ankle). Each segment of the lower leg is represented as a stiff bar with distinct properties. These properties include length, mass, and the proximal center of gravity. These segments are oriented at angular locations relative to the vertical axis. This modeling framework is based on a fixed coordinate system with its origin located at the hip joint [64]. The three-link leg model defined as a thigh, a shank, and a foot structure is relevant in assuring the validity of the overall model's biomechanics and dynamics representation of a human being. When dealing with the model validation, joint angles, velocities, and active muscle forces should typically be within human locomotion range. Generally, the validation procedure starts with component-level unit tests, which include establishing the lengths, masses, and joint limits of the links that are consistent with the structures of human legs. Experimental parameters such as joint torques, ground reaction forces, and trajectory movement of the end-effector (foot) are directly compared with the outputs of motion capture systems and force platforms to check whether human patterns are replicated by the artificial limbs. In addition, a sensitivity analysis is also performed to evaluate the effects of the alteration on the optimal settings and other parameters selected on the stability and consistency of the system. This procedure will guarantee that the model is reliable for the evaluation of the controller and biomechanical tests.

For model validation a 27-year-old men with a weight of $74kg$ and $1.74m$ height is considered. For another candidate, a Matlab program could be used to accept the Height (H) and Mass(M) from the user and the corresponding parameters can be calculated. For this simulation, a second and third candidate female with a weight $46kg$ and $1.56m$ height and a male with a weight $100kg$ and $1.90m$ height is called by a Matlab code to

calculate the required parameters. The code is presented in Appendix B.1. Using this the performance of the SMC and ST-SMC can be computed for parameter variation. The reference motion of both human legs may be seen as a singular pattern with distinct phase delays. Consequently, only the unilateral parameters of a single right leg (including the hip, knee, and ankle) were shown in Table 6.1.

Body	THIGH	LOWER LEG	FOOT
Mass(kg)	$m_1 = 0.1 * M$ = 7.4	$m_2 = 0.0465 * M$ = 3.44	$m_3 = 0.0145 * M$ = 1.073
Length(m)	$L_1 = 0.284 * H$ = 0.494	$L_2 = 0.246 * H$ = 0.428	$L_3 = 0.152 * H$ = 0.265
Proximal center of gravity(m)	$l_1 = 0.433 * L_1$ = 0.214	$l_2 = 0.433 * L_2$ = 0.185	$l_3 = 0.5 * L_3$ = 0.053
Radius of gyration(m)	$k_1 = 0.540 * L_1$ = 0.267	$k_2 = 0.528 * L_2$ = 0.226	$k_3 = 0.528 * L_3$ = 0.028
Moment of inertia(kg.m ²)	$J_1 = m_1 * k_1^2$ = 5.27	$J_2 = m_2 * k_2^2$ = 3.78	$J_3 = m_3 * k_3^2$ = 0.058
Viscous damping coefficient (N.m.s)	$b_1 = 0.1$	$b_2 = 0.1$	$b_3 = 0.1$

Table 6.1 Parameter for subject 1

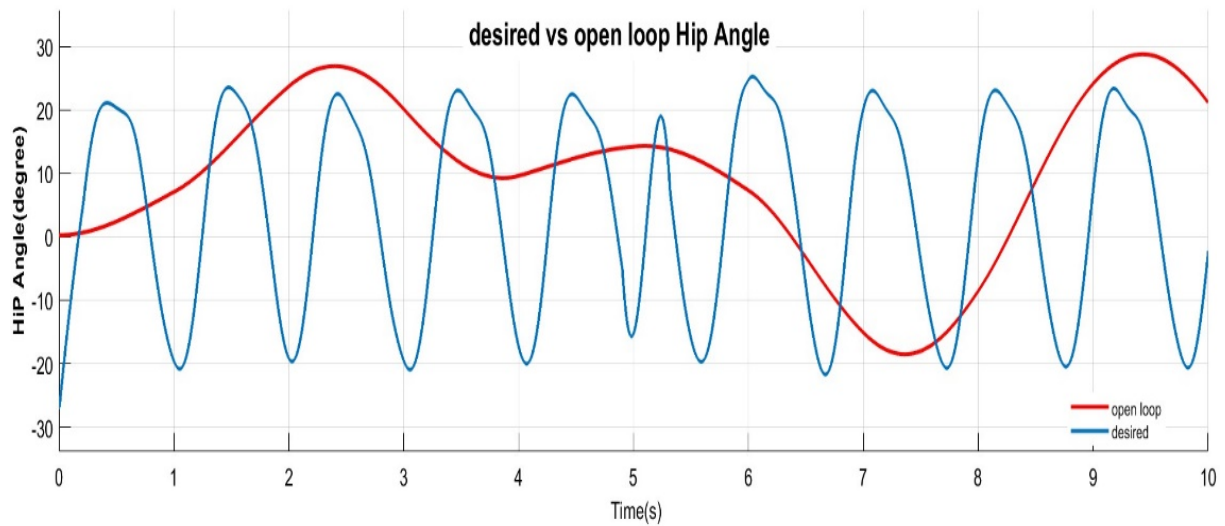


Figure 6.1 Plant Hip angle open loop response for model validation.

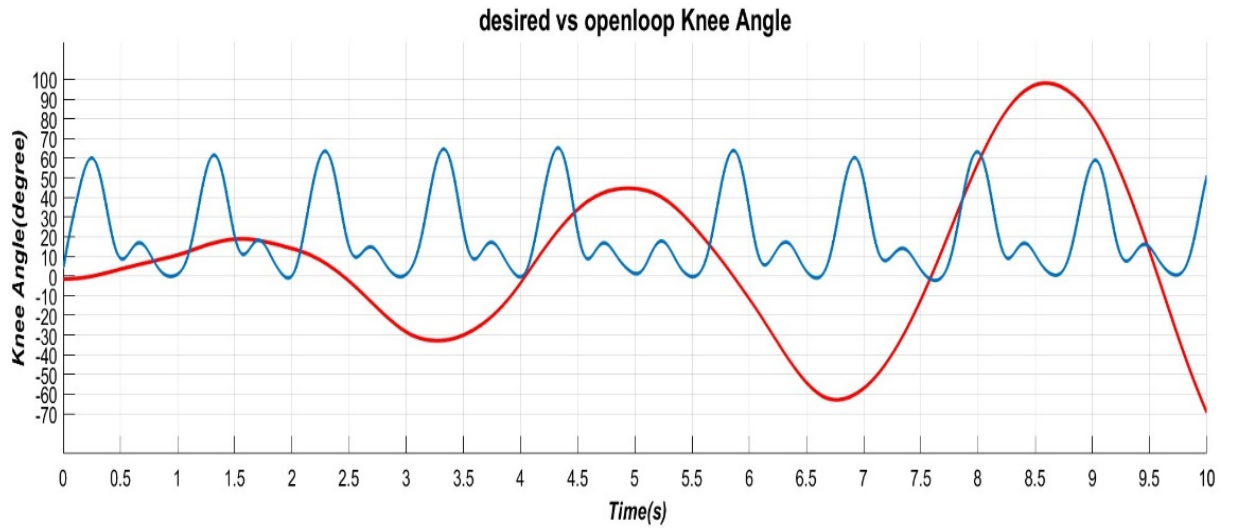


Figure 6.2 Plant Knee angle open loop response for model validation.

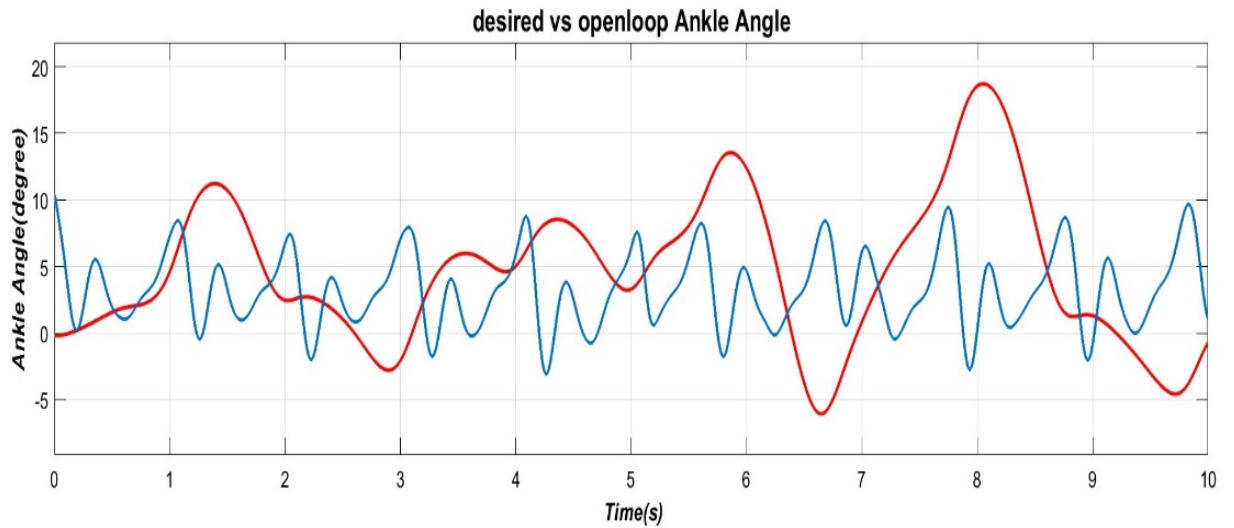


Figure 6.3 Plant Ankle angle open loop response for model validation.

Control parameter ra	Model Joints		
	Hip Joint	Knee Joint	Ankle Joint
Torque Applied (Nm)	510	630	860
Output Angle Range (deg)	-20.5Upto29.5	-60Upto100	-5.5Upto18.5

Table 6.2 Leg model open loop response for model validation.

As can be seen in Table 6.1 for the applied torque range within the human dynamical capability, the model gives a reasonable range of motions at each joint.

The three-link leg model proposed in this paper, Fig. 4.5 demonstrates kinematic and dynamic responses well within acceptable bounds, as seen in Table 6.2 when compared to experimental walking trajectory data. Therefore, it successfully models joint trajectories and the necessary torque requirements over most of the gait cycle, which demonstrates its

potential application in modeling human gait mechanics. This model is therefore suitable for further works like prosthetic control using SMS and STSMC since it can accurately predict lower limb movement.

6.1.1 sEMG Signal Processing

The raw sEMG signals responsible for each joint movement are filtered, rectified, and smoothed to obtain the necessary features. A low-pass filter was utilized to remove noise without losing salient signal features. For the selected muscles, root mean square (RMS) features were extracted from sEMG signals. In Fig. 6.4 raw sEMGVastusMedialis is shown, then it is filtered by low-pass and high-pass filters as shown in Fig. 6.5a and Fig. 6.5b. Then, the signal is rectified and linearized as shown in Fig. 6.6a and Fig. 6.6b to obtain a signal where MAV feature extraction is performed. Fig. 6.7 shows the linearized and the raw sEMGVastusMedialis, with the same manner the for the remaining eight signal processing is performed. Each signal has its feature correlated with flexion-extension movements at each joint. The filtering parameters for each signal are listed in Table 6.3.

Muscle	Fc-low	Fc-high	Sampling rate
Tensor fascialata	150	1000	2500
Rectus femoris	150	1000	2500
Semimembranosus	5	1000	2500
Medial gastrocnemius	20	800	1000
Vastus medialis	20	600	1000
Lateral gastrocnemius	20	600	1000
Upper tibialis anterior	40	400	1000
Lower tibialis anterior	40	400	1000
Soleus	150	100	1000

Table 6.3 sEMG signals filtering parameters

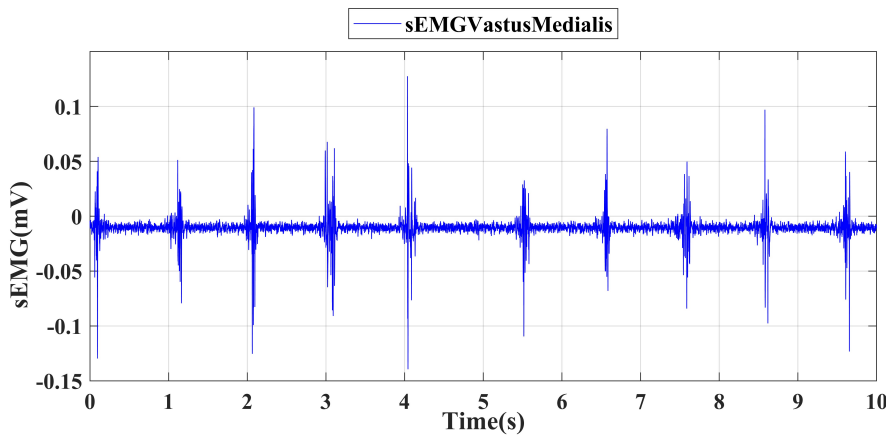
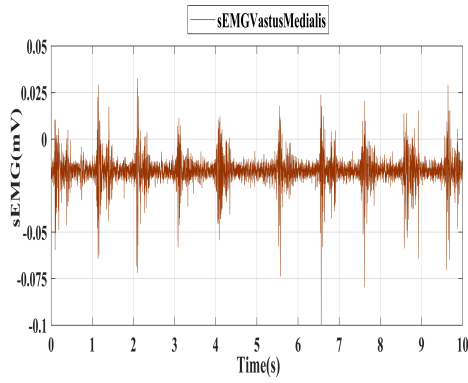
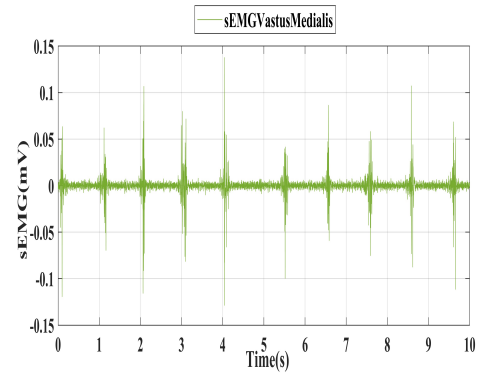


Figure 6.4 Raw sEMG VastusMedialis

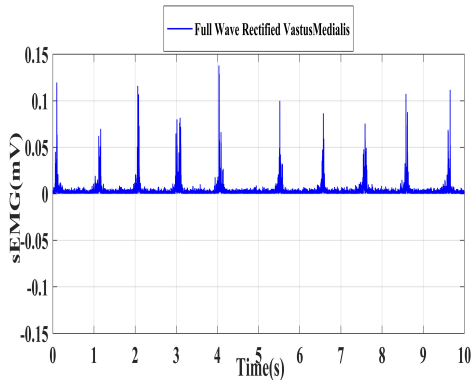


(a) Low-pass filtered Vastusmedialis.

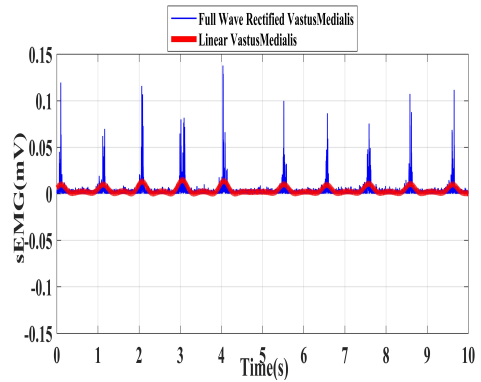


(b) High-pass filtered Vastusmedialis.

Figure 6.5 Shows low-pass filtered and High-pass filtered Vastusmedialis from raw data.



(a) Full wave rectified Vastus medialis.



(b) Full wave rectified vs Linear Vastus medialis.

Figure 6.6 Shows the full wave rectified and the Linear Vastus medialis after further processing.

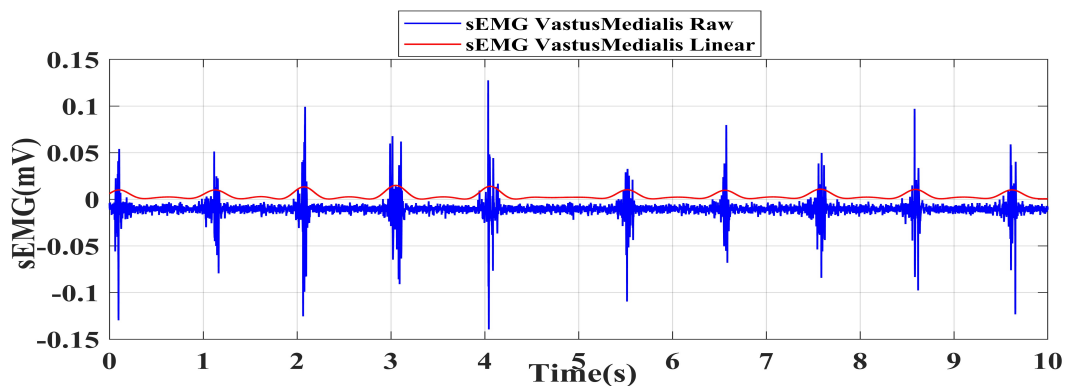
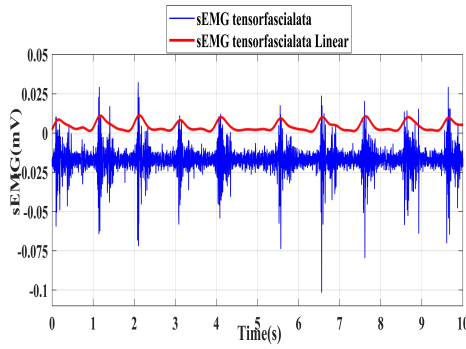
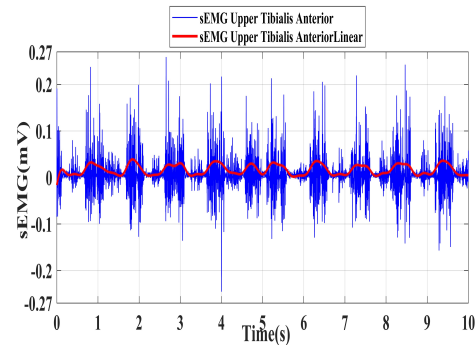


Figure 6.7 Shows how the raw sEMG Vastusmedialis data is converted into a linear signal to be used as an input to NN.

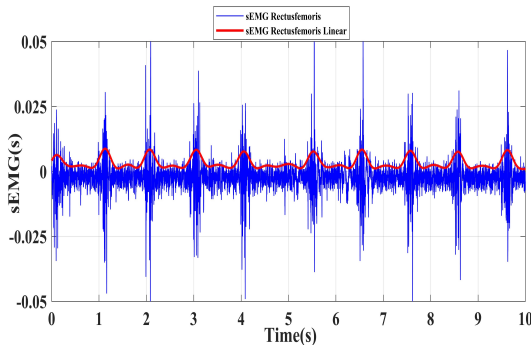


(a) Shows raw and linear Tensorfascialata signal.

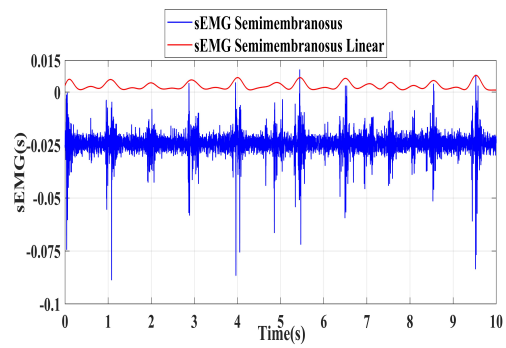


(b) Shows raw and linear upper TibialisAnterior.

Figure 6.8 Shows how the raw Tensorfascialata and Upper TibialisAnterior sEMG data converted into a linear signal to be used as an input to NN.

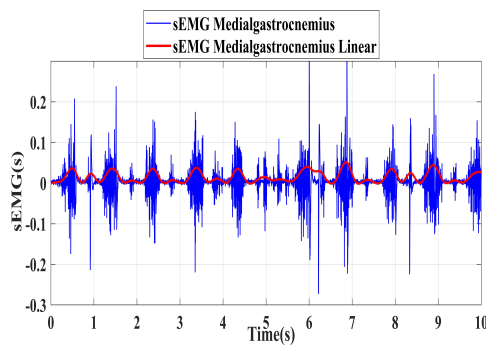


(a) Shows raw and linear lower RectusFemoris signal.

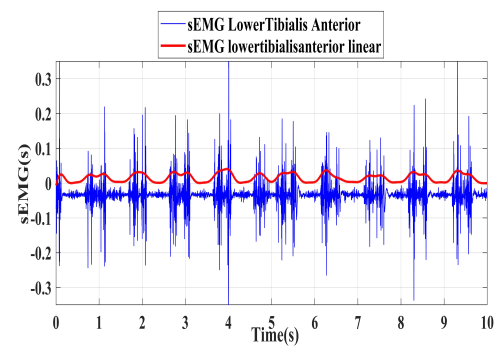


(b) Shows raw and linear lower Semimembrosis signal.

Figure 6.9 Shows how the raw RectusFemoris and Semimembrosis sEMG data converted into a linear signal to be used as an input to NN.



(a) Shows raw and linear MedIalgastrocnemius signal.



(b) Shows raw and linear lower TibialisAnterior signal.

Figure 6.10 Shows how the raw MedIalgastrocnemius and TibialisAnterior sEMG data are converted into a linear signal to be used as an input to NN.

As observed in Fig. 6.8a, Fig. 6.8b, Fig. 6.9a, Fig. 6.9b, Fig. 6.10a and Fig. 6.10b the filter was successful in absorbing high-frequency and low-frequency noise without interfering with vital components of the signal. The SNR is used to measure the quality of

the processed signal. For the lower leg muscle during moderate contraction, the signal power is measured to be $10\mu\text{V}$ up to $50\mu\text{V}$ RMS, whereas noise power is $5\mu\text{V}$ up to $20\mu\text{V}$ RMS. This suggests that raw sEMG signals from lower leg muscles often have an SNR of around 6dB or lower before filtering. The filtered signal's SNR improved significantly after applying the filter with the listed preprocessing techniques, reaching 20dB up to 30dB for the nine muscles.

6.2 Joint Angle Smotthing Using Savitzky-Golay filter

For smoothing the measured joint angles in this paper, Savitzky-Golay filter in the MATLAB software can be tuned and adjusted by the polynomial order, window length, and derivative order, among other parameters, which is presented in Table 6.4.

Control parameter ra	Model Joints		
	Hip Joint	Knee Joint	Ankle Joint
Filter parameters	5	7	9
Window Size (N)	3	3	3
Polynomial Order (k)	0	0	0
Derivative order(D)	5	7	9

Table 6.4 Savitzky-Golay filter parameter for Smoothing

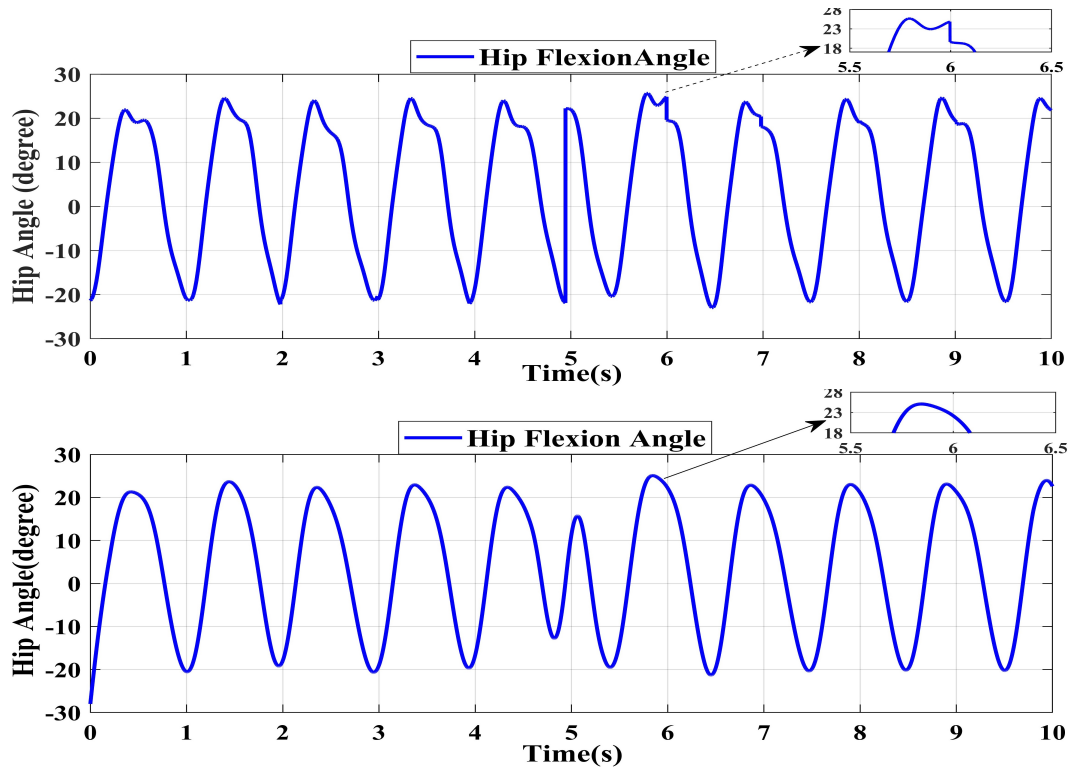


Figure 6.11 Shows Hip angle Unfiltered vs Filtered.

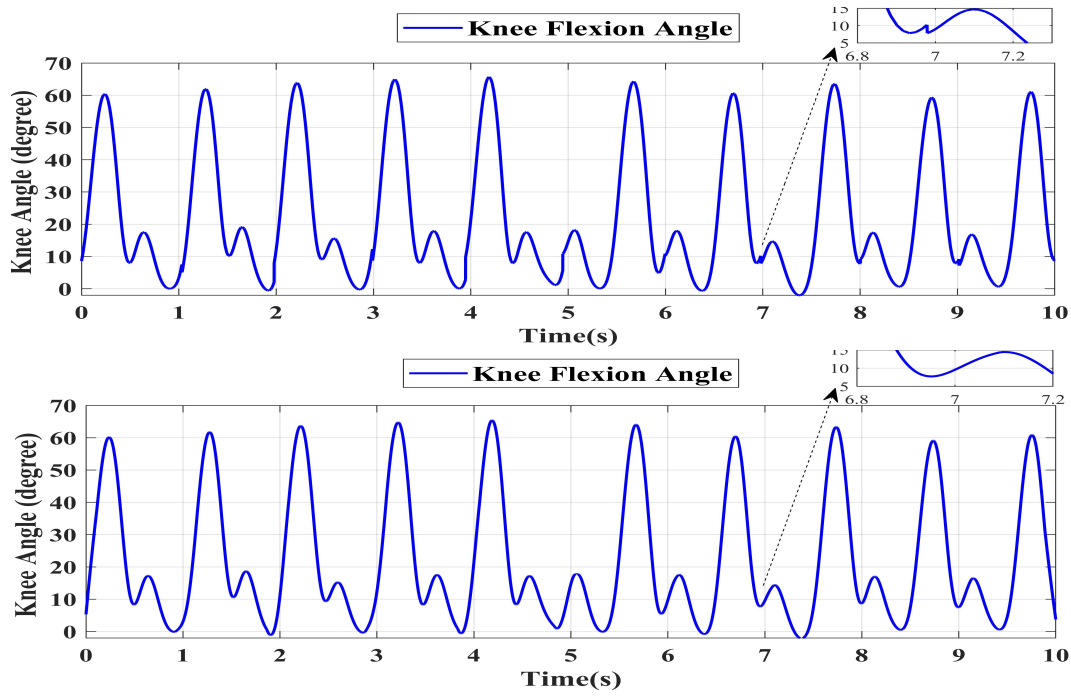


Figure 6.12 Shows Knee angle Unfiltered vs Filtered.

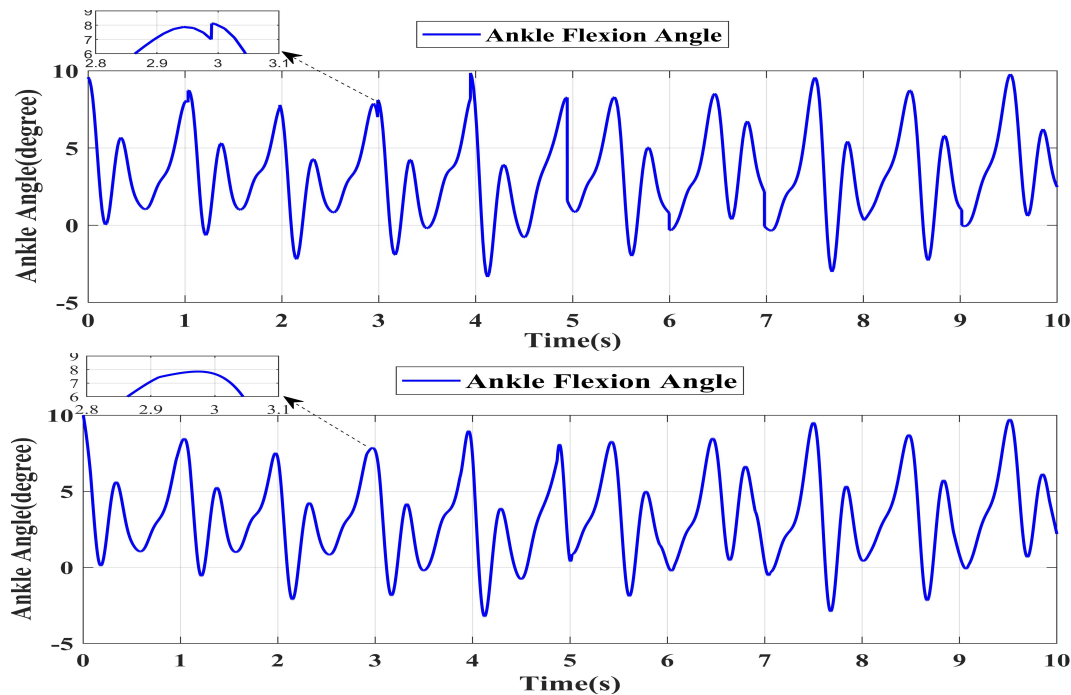


Figure 6.13 Shows Ankle angle Unfiltered vs Filtered.

As seen from the results that are shown in Fig. 6.11, Fig. 6.12 and Fig. 6.13 the filter effectively eliminated high-frequency disturbances without disrupting essential components of the signal. The filter has also yielded fair estimations of the signal's slopes and peaks with only slight alteration at the fringes of the signal. SNR measures the ratio of the power of the signal to the power of noise. SNR for the unfiltered signal was 10.56dB,

10.76dB and 11.25dB. After applying the filter, SNR improved to 20.15dB, 20.32dB and 19.5dB showing improved signal quality. This demonstrates how the Savitzky-Golay filter enhances SNR, quantifying its effectiveness in reducing noise while preserving the underlying joint angle signal.

6.2.1 FFNN Training

In this paper, NN performance analysis regression value close to 1, MSE values with low values are ideal. Regression value evaluates fit quality, while MSE quantifies prediction accuracy. Using these metrics together provides a thorough understanding of NN performance and ensures reliable predictions.

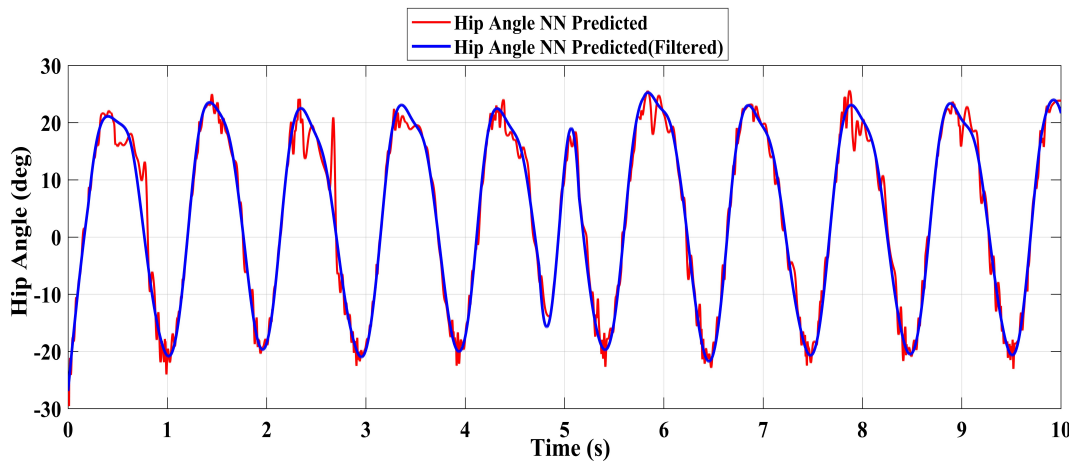


Figure 6.14 NN trained Hip angle.

NN Hip Angle Regression

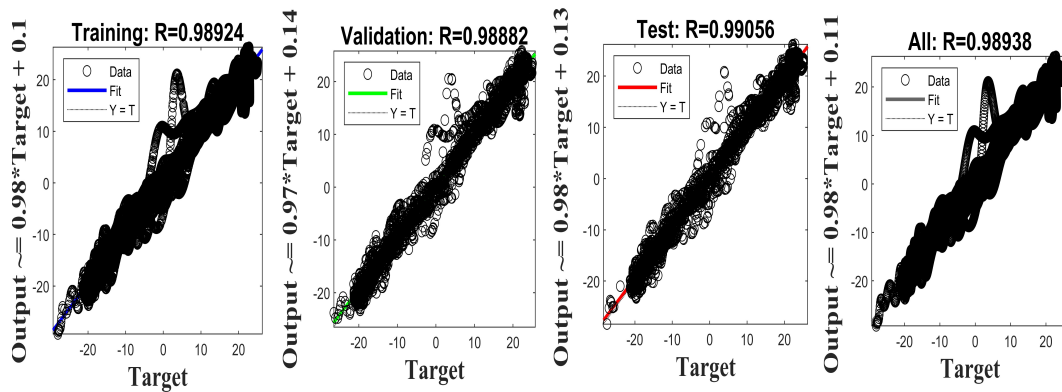


Figure 6.15 Regression value for Hip angle training.

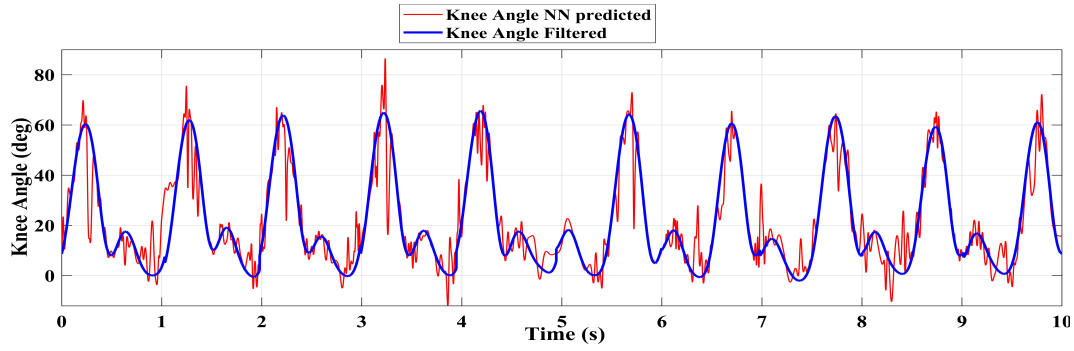


Figure 6.16 NN trained Knee angle.

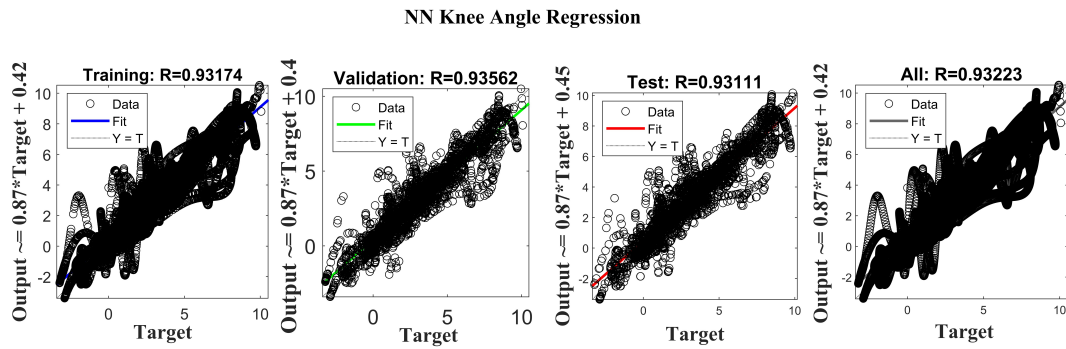


Figure 6.17 Regrasion value for knee angle training.

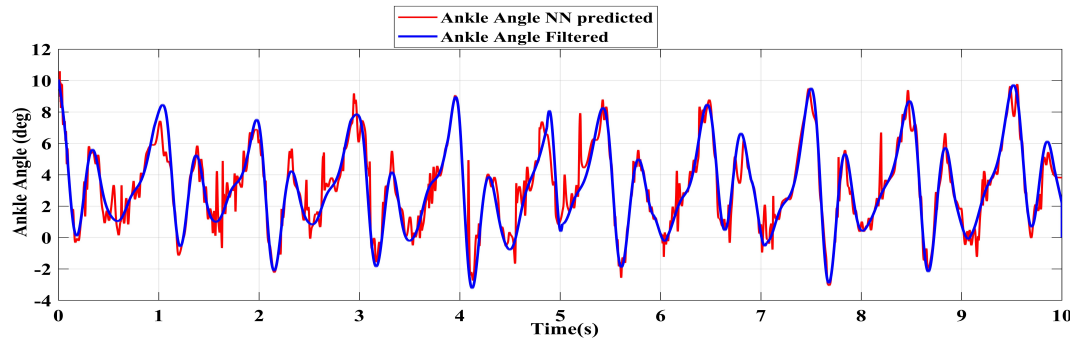


Figure 6.18 NN trained Ankle angle.

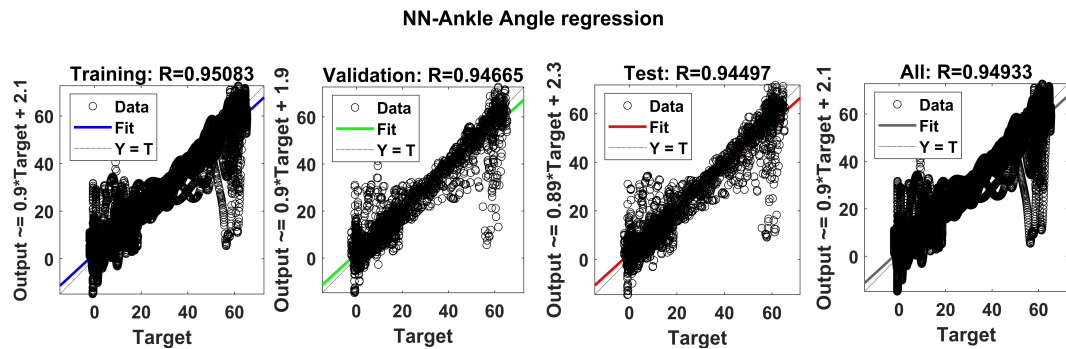


Figure 6.19 Regrasion value for Ankle angle training.

As shown in Fig. 6.14, Fig. 6.16 and Fig. 6.18 the neural network predicted the joint angles using sEMG signals as inputs. Fig. 6.15, Fig. 6.17 and Fig. 6.19 shows R^2 values 0.989, 0.932 and 0.949, which means that a strong correlation prevailed between predicted and actual joint angles.

6.3 controller design

6.3.1 Walking, DNS and UPS trajectory tracking

Waking Trajectory Tracking

The parameter range for a Super-Twisting Sliding Mode Controller (ST-SMC), particularly the gains k_1 and k_2 , is contingent upon the attributes of the controlled system, such as system dynamics, disturbance magnitudes, and the desired convergence rate. Nonetheless, broad guidelines may be offered to assist in initializing the parameters prior to fine-tuning via simulation and testing.

Controller	Parameter	Hip Joint	Knee Joint	Ankle Joint
SMC	K1	60	50	60
	K2	85	75	75
ST-SMC	β	30	25	35
	α	70	50	60

Table 6.5 Controller design parameters

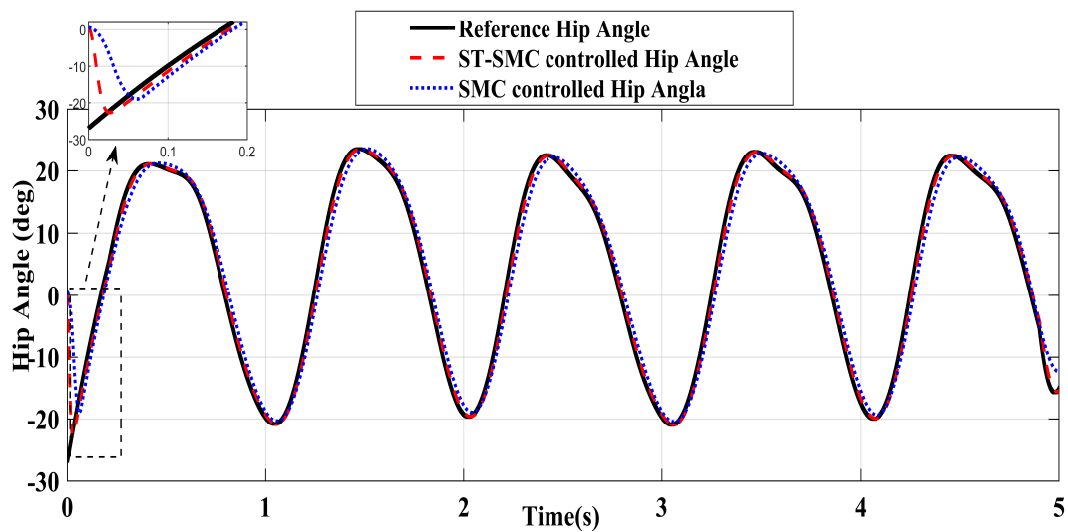
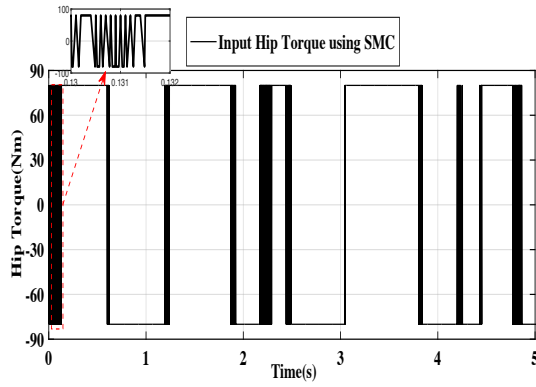
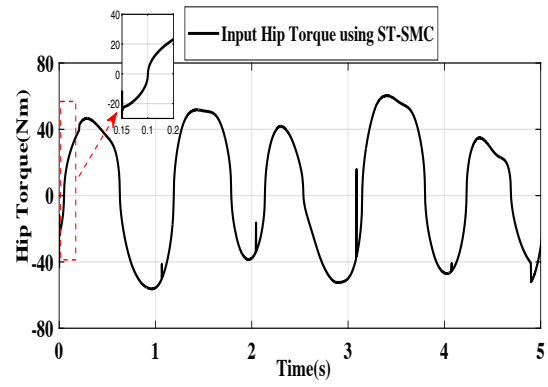


Figure 6.20 STSMC and SMC hip angle.



(a) Input torque at the hip with chattering.



(b) Input torque at the hip without chattering.

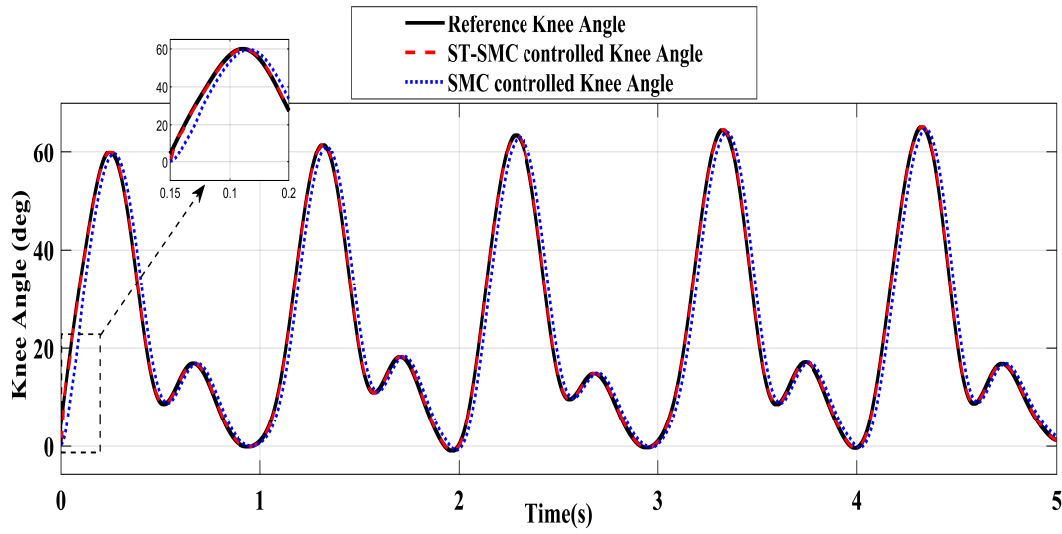
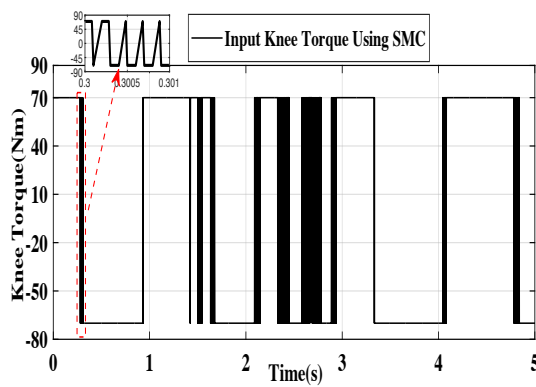
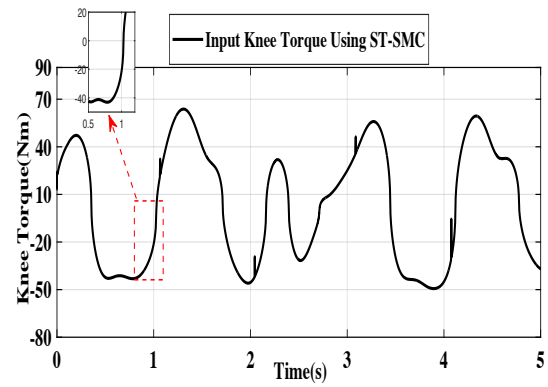


Figure 6.22 STSMC and SMC knee angle.



(a) Input torque at the knee with chattering.



(b) Input torque at the knee without chattering.

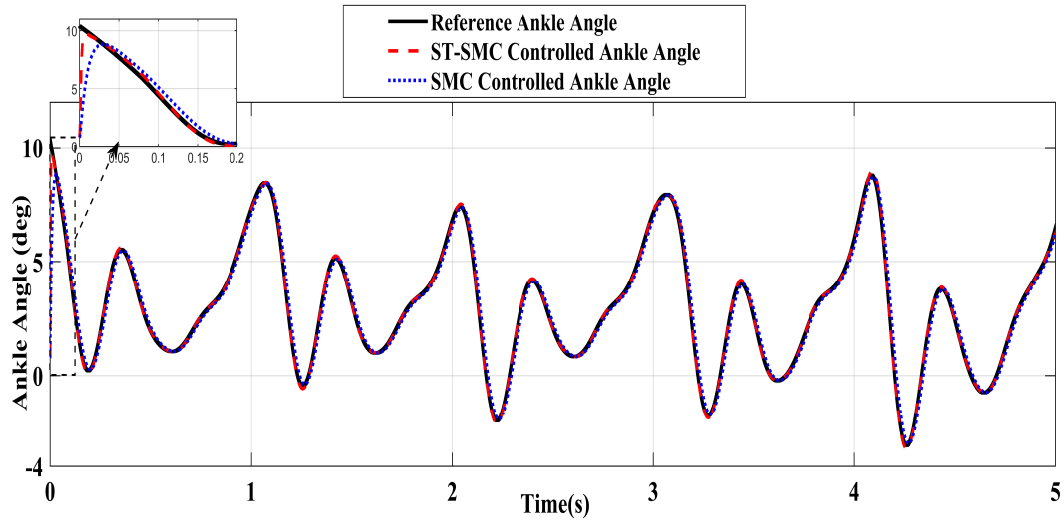
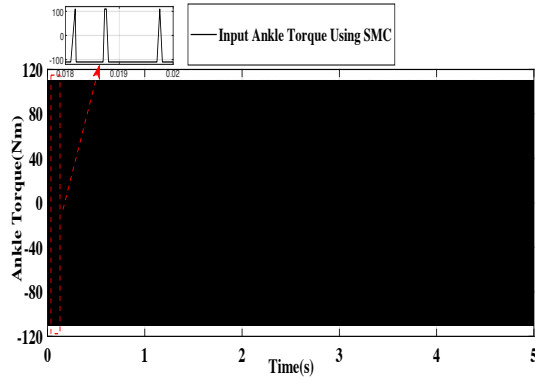
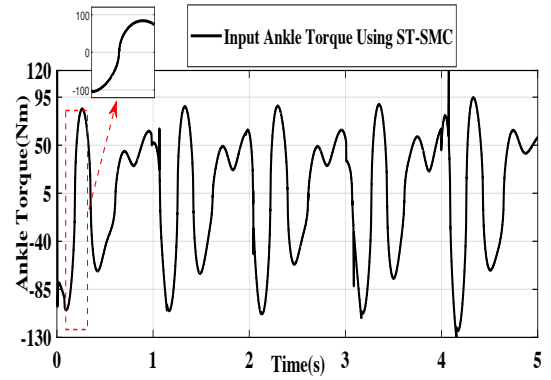


Figure 6.24 STSMC and SMC Ankle angle.



(a) Input torque at the ankle with chattering.



(b) Input torque at the ankle without chattering.

As shown in Fig. 6.20, Fig. 6.22 and Fig. 6.24 both SMC and STSMC experienced low tracking errors, with the STSMC performing best on chattering reduction. As depicted in Fig. 6.21a, Fig. 6.23a and Fig. 6.25a the input torque shows chattering, while in Fig. 6.21b, Fig. 6.23b and Fig. 6.25b the chattering is removed. SMC was able to perform well with comparatively higher control effort in input torque because of discontinuous control actions. While ST-SMC results in reduced control effort from $80Nm$ to $50Nm$ at the hip, from $70Nm$ to $60Nm$ at the knee, and from $90Nm$ to $40Nm$ at the ankle.

Control parameter \ Joints	Hip Joint	Knee Joint	Ankle Joint
Torque Applied (Nm)	80 to 50Nm	70 to 60Nm	90 to 40Nm
Energy consumption (J) SMC and ST-SMC	165to127.5	175to138.2	207to92.2

Table 6.6 Energy consumption of prosthetic for SMC and ST-SMC.

Performance Analysis

For the analysis of the performance, RMS analysis is used. RMS analysis provides a quantitative measure of a system's performance by calculating the average power or magnitude of the error signal over time.

Controller	Hip Joint MSE	Knee Joint MSE	Ankle Joint MSE
SMC	0.6	0.156	0.2
ST-SMC	0.037	0.042	0.054

Table 6.7 Average MSE values for walking trajectory tracking.

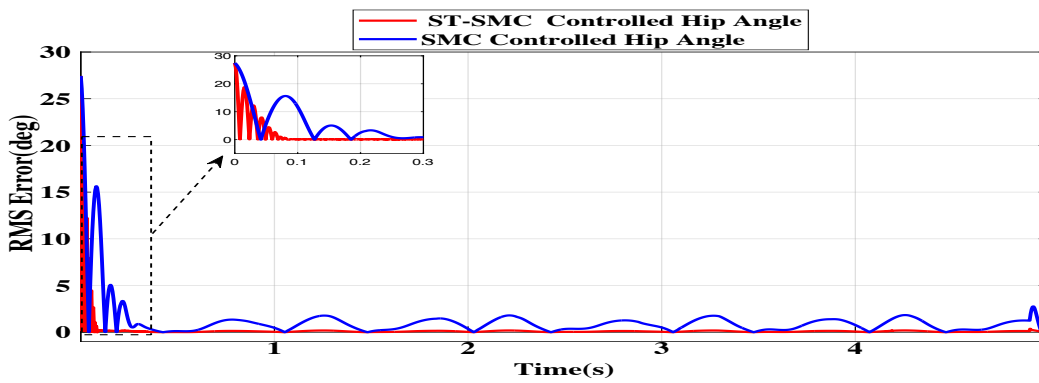


Figure 6.26 RMS value for hip angle ST-SMC and SMC controlled.

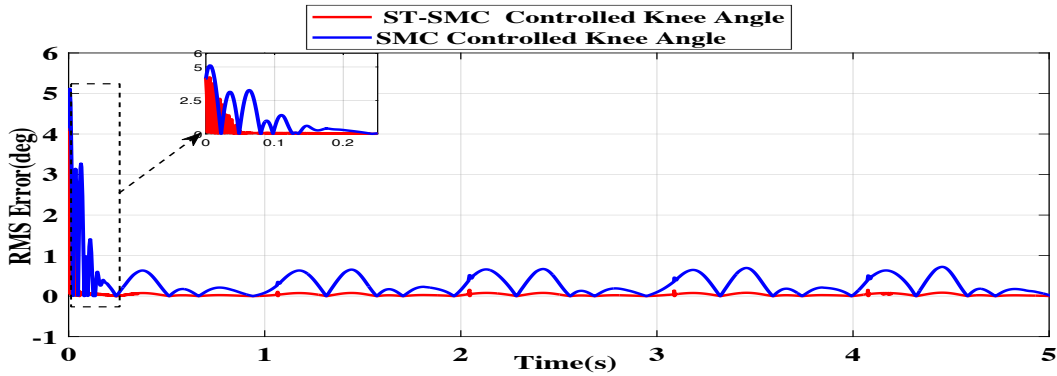


Figure 6.27 RMS value for knee angle ST-SMC and SMC controlled.

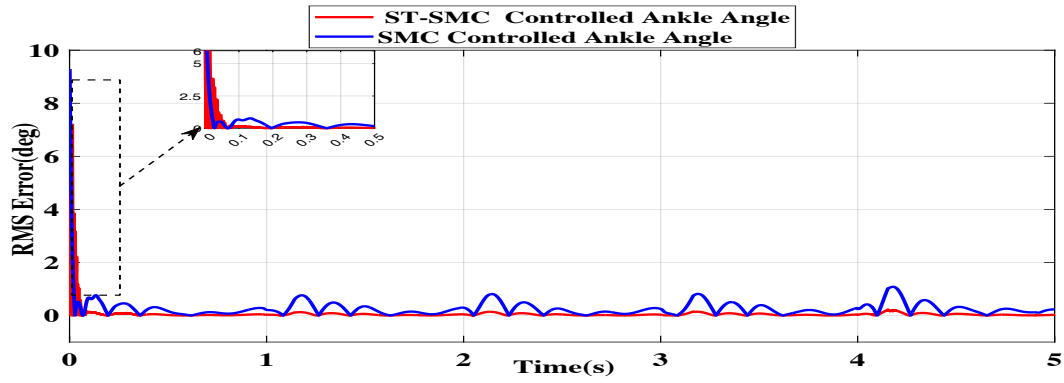


Figure 6.28 RMS value for ankle angle ST-SMC and SMC controlled.

As it is observed in Fig. 6.26, Fig. 6.27 and Fig. 6.28 RMS values for the SMC and ST-SMC are compared, and their values are listed in Table 6.7. For the walking trajectory, it can be concluded that the ST-SMC has a more accurate trajectory-tracking ability than the SMC.

Down-stairs (DNS) Trajectory tracking

In the same manner, discussed above, the ST-SMC controller can trace the UP Stairs (UPS) and Downstairs (DNS) trajectories at each joint.

Controller	Parameter	Hip Joint	Knee Joint	Ankle Joint
SMC	K1	45	68	65
	K2	78	95	70
ST-SMC	β	48	55	64
	α	78	67	70

Table 6.8 Controller design parameters for DNS.

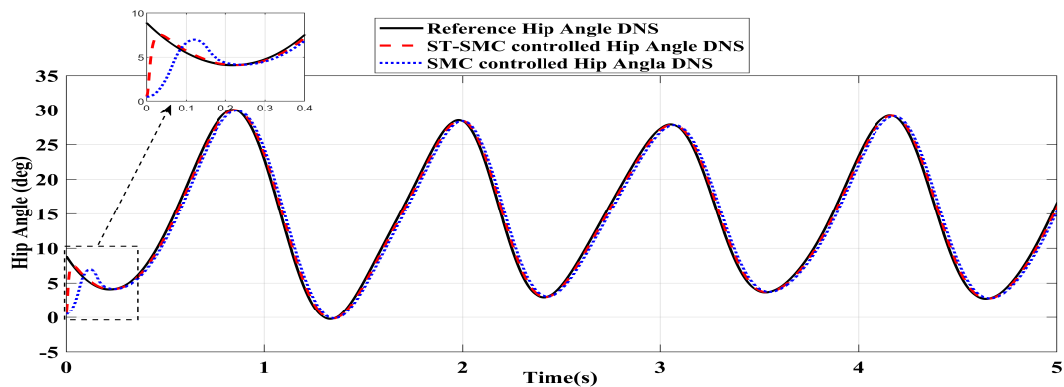


Figure 6.29 DNS Hip joint trajectory ST-SMC and SMC controlled.

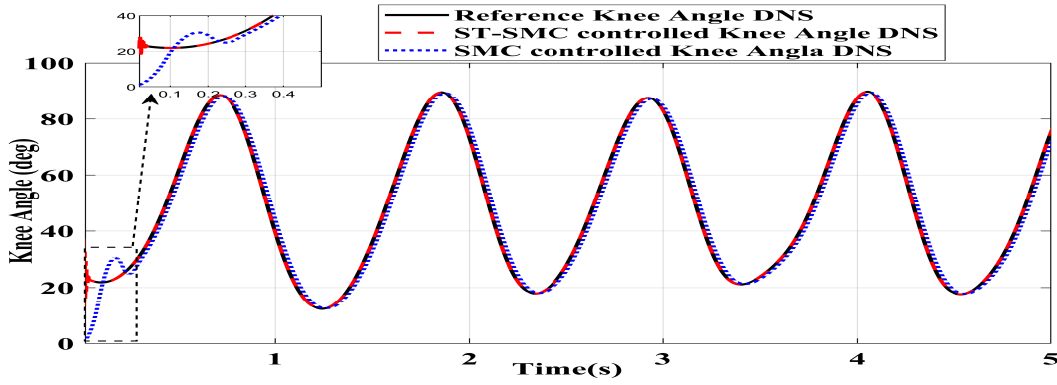


Figure 6.30 DNS Knee joint trajectory ST-SMC and SMC controlled.

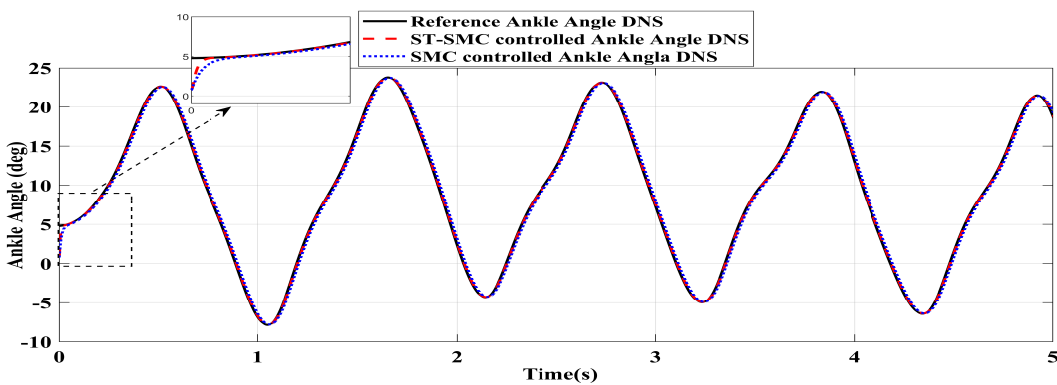


Figure 6.31 DNS Ankle joint trajectory ST-SMC and SMC controlled.

The tracking offered by sliding mode control (SMC) was considerably affected during sharp changes in the vertical trajectories, while ST-SMC achieved superior tracking accuracy as shown in Fig. 6.29, Fig. 6.30 and Fig. 6.31. The MSE for SMC is 0.63° , 0.246° , and 0.22° while for the ST-SMC the MSE value decreased to 0.026° , 0.035° and, 0.043° respectively, for the hip, knee, and ankle joints. Both the controllers showed better robustness with minimal compromise on the tracking accuracy.

UP-stairs (UPS) Trajectory Tracking

For the upstairs (UPS) trajectory, the control parameter and the MSE values are discussed in Table 6.9.

Controller	Parameter	Hip Joint	Knee Joint	Ankle Joint
SMC	K1	65	58	75
	K2	80	85	75
ST-SMC	β	55	75	65
	α	65	70	60

Table 6.9 Controller design parameters for UPS.

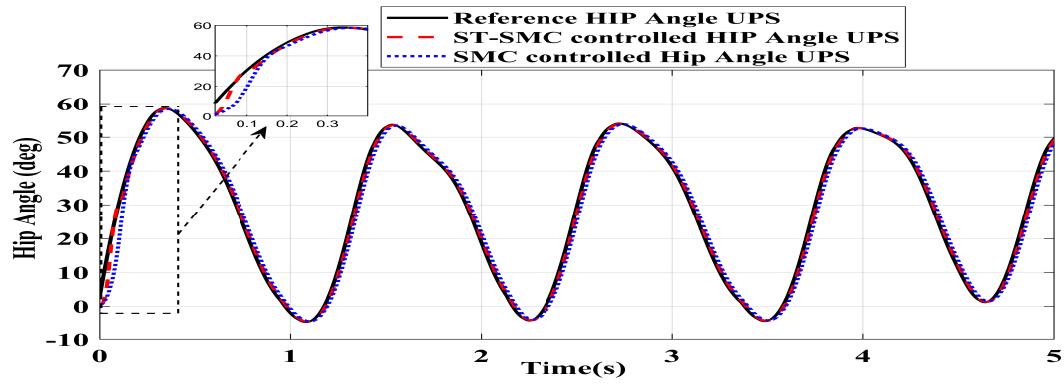


Figure 6.32 UPS hip joint trajectory ST-SMC and SMC controlled.

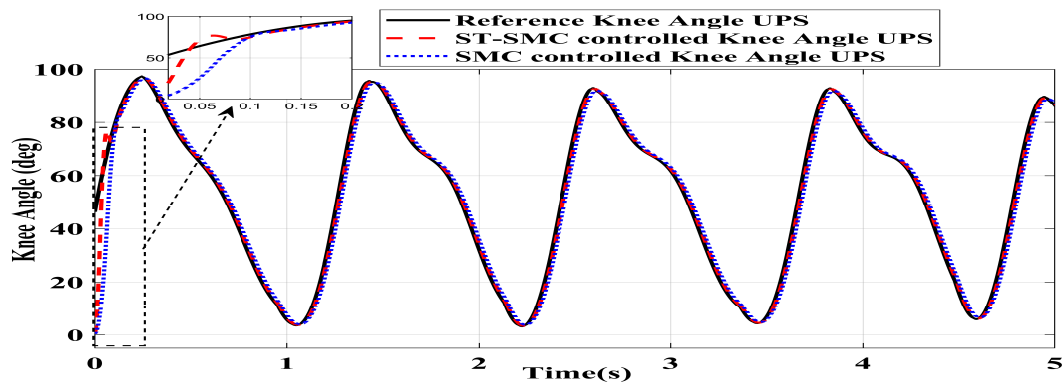


Figure 6.33 UPS knee joint trajectory ST-SMC and SMC controlled.

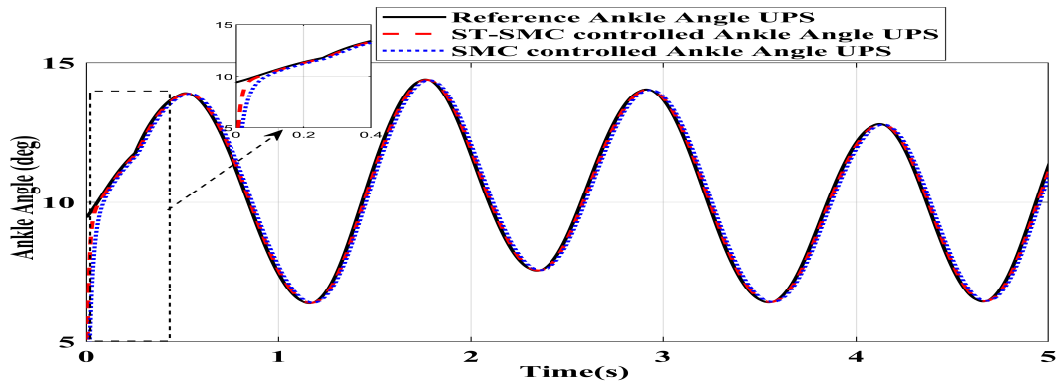


Figure 6.34 UPS ankle joint trajectory ST-SMC and SMC controlled.

As shown in Fig. 6.32, Fig. 6.33 and Fig. 6.34 both the controllers showed better robustness with minimal compromise on the tracking accuracy. The MSE for SMC is 0.532° , 0.346° , and 0.332° while for the ST-SMC the MSE value decreased to 0.016° , 0.015° and 0.023° respectively, for the hip, knee, and ankle joints.

6.3.2 Parameter change Analysis

For analysis of parameter changes on the controller persons with a weight of $46kg$, $74kg$, and $100kg$ height $1.56m$, $1.74m$ and $1.95m$ is considered. For another candidate, a Matlab program could be used to accept the Height (H) and Mass(M) from the user, and the corresponding parameters can be calculated. The MSE result in degree for the corresponding results are shown in Table 6.11.

Subject	Hip Joint MSE	Knee Joint MSE	Ankle Joint MSE
Sub-1	0.05	0.062	0.050
Sub-2	0.04	0.07	0.070
Sub-3	0.05	0.035	0.045

Table 6.10 MSE(deg) analyses for parameter change.

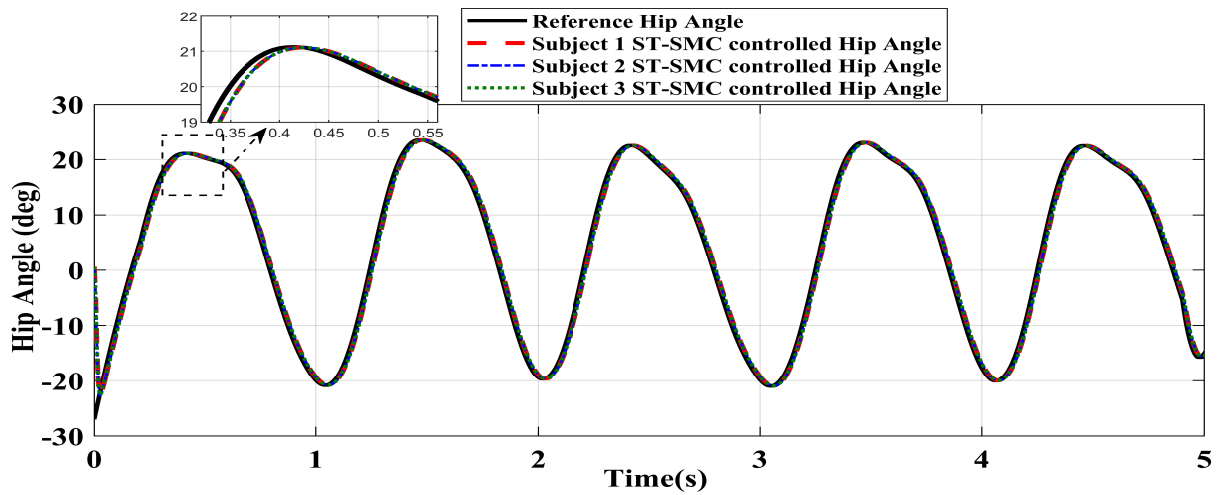


Figure 6.35 STSMC Hip angle for Subject 1, Subject 2, and Subject 3.

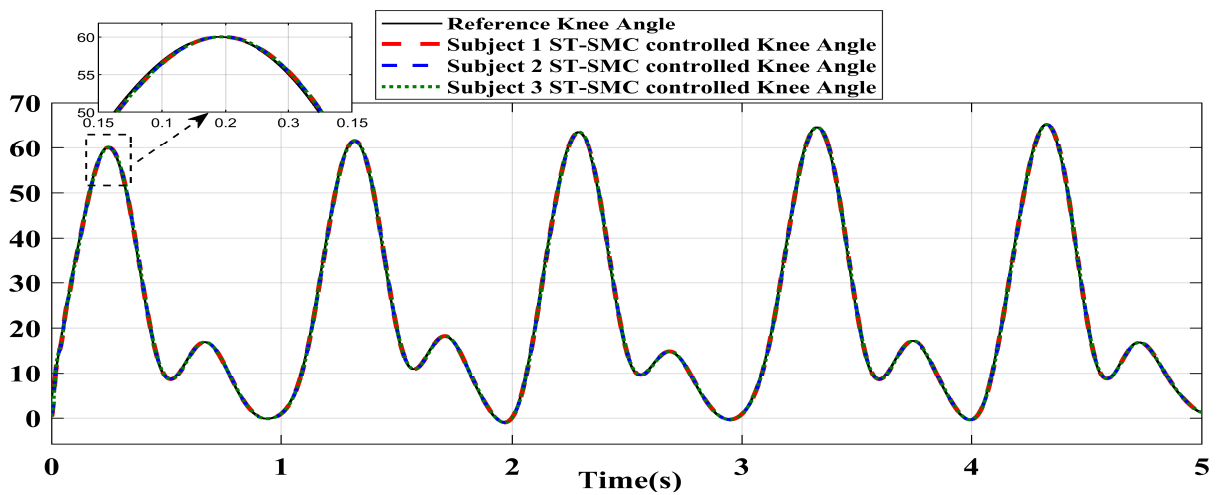


Figure 6.36 STSMC Knee angle for Subject 1, Subject 2, and Subject 3.

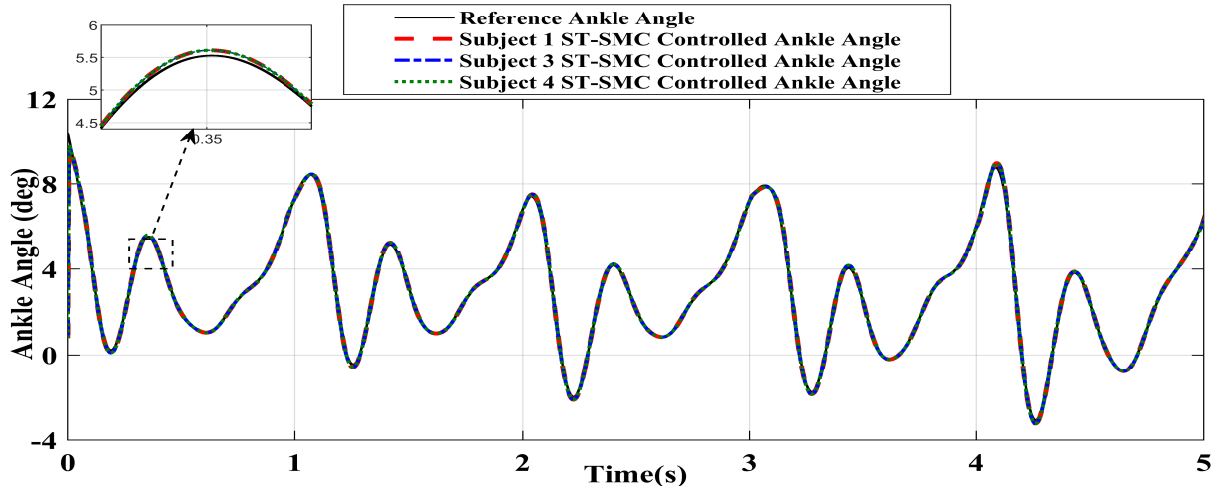


Figure 6.37 STSMC ankle angle for Subject 1, Subject 2, and Subject 3.

From Fig. 6.35, Fig. 6.36 and Fig. 6.39 it can be concluded that both controllers exhibited satisfactory behavior when subjected to parameter changes, while ST-SMC showed to be more efficient against the parameter change.

6.3.3 Disturbance Analysis

Various disturbances in the form of biomechanical disruptions, external environmental obstacles, physiological anomalies, measurement, and sensor noise affect joint motions. This could compromise stability, precision, and resilience in control systems regulating the human gait cycle for applications involving exoskeletons, prosthetics, and rehabilitation robots. The disturbance modeling for the controller design would therefore be sinusoidal disturbances, Gaussian noise, impulse and step disturbances, and time-varying disturbances. With this disturbance, the STSMC controllers must be more capable of dealing with the gait-specific dynamics of both humans and assistance devices. In this paper, sinusoidal signals are used to describe ground reaction forces (GRF). Fig. 6.35, Fig. 6.36 and Fig. 6.39 show the response of ST-SMC controller for (GRF) applied on the walking gait cycle.

Disturbance Nature	Hip Joint MSE	Knee Joint MSE	Ankle Joint MSE
Without disturbance	0.071	0.052	0.049
GRF(t)=20sin(4t) + 10sin(4ft))	0.5	0.45	0.55

Table 6.11 MSE(deg) analyses with and without disturbance.

Both controllers exhibited satisfactory MSE values described in Table 6.11 when subjected to ground reaction force (GRF), while ST-SMC showed to be more efficient against the disturbances as the MSE value is less.

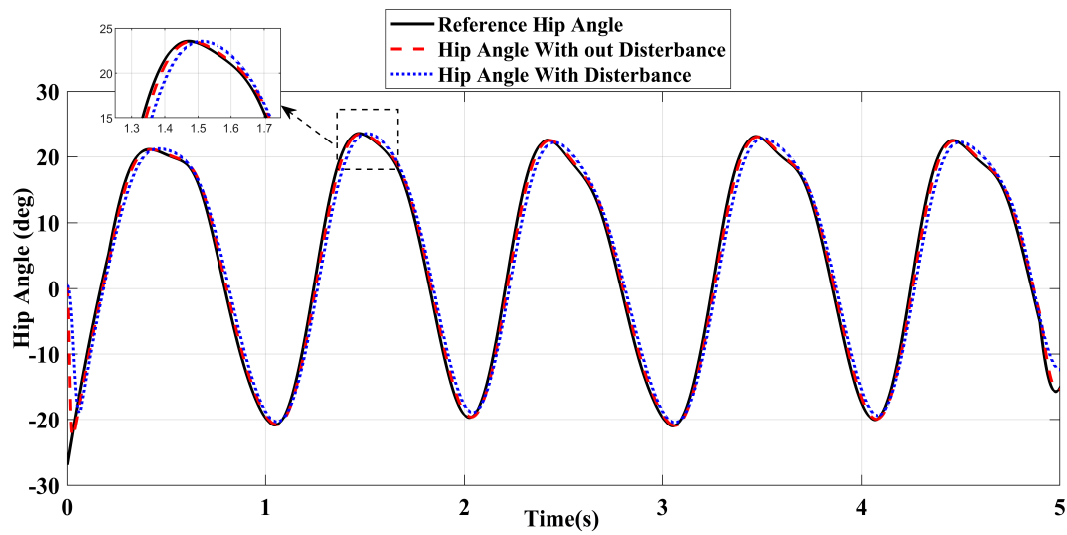


Figure 6.38 ST-SMC controlled Hip angle with and without disturbance.

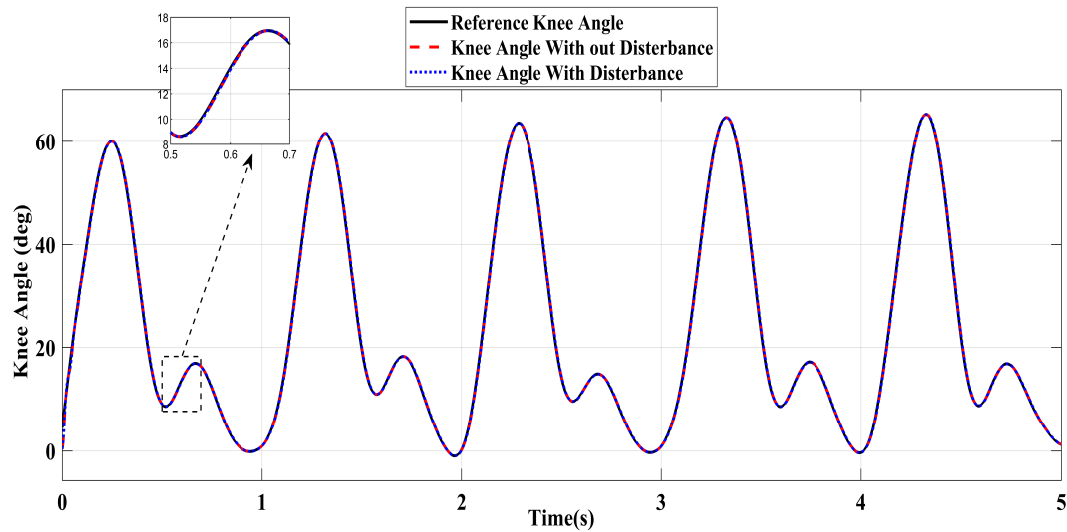


Figure 6.39 ST-SMC controlled Knee angle with and without disturbance.

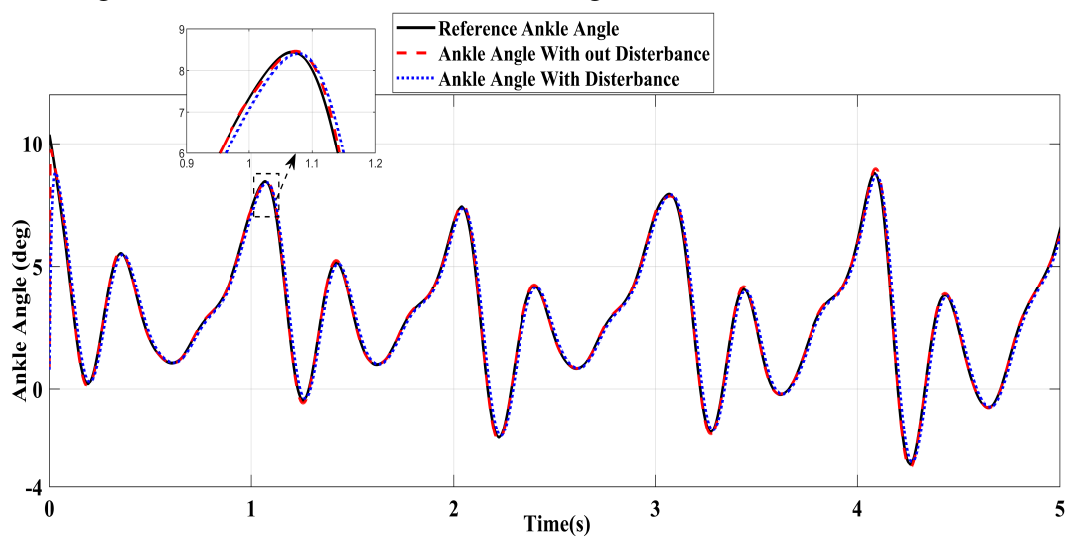


Figure 6.40 ST-SMC controlled Ankle angle with and without disturbance.

Chapter 7

Conclusion and Future Works

To conclude, this work examines using sEMG signals as inputs to the FNN which generates a reference signal for the prosthesis, and then the proposed ST-SMC controller enforces the given trajectory in the presence of external disturbances and system parameter changes. To design the suggested system, first the raw sEMG signal is processed for each muscle group at each joint with its own sampling frequencies, and the linearized signal is used as training data. The FFNN predicts the joint angles for walking, UP-stairs and down-stairs then this output is filtered using the Savitzky-Golay filter. Finally, the proposed ST-SMC controller can enforce the predicted joint angle and compared with conventional SMC controller.

Ultimately, to assess the efficacy of the suggested control method, simulations have been conducted. The simulation findings confirm the efficacy and superiority of the suggested control strategy over sliding mode control (SMC) in attaining high tracking accuracy and disturbance rejection capabilities under nominal conditions, as well as in the presence of parameter variations and disturbances. While the model turned out to be very successful with real-time processing of a signal and accuracy of control, further research may address issues such as enhancement of robustness under various conditions of terrain and improvement of adaptability to individual features of a specific user. This can eventually allow lower-limb prosthetic control to have a more natural, responsive interface, users greater independence and certainty in locomotion.

Future work

Future control techniques for lower limb prosthetics include adding joint degrees of freedom up to nine degrees of freedom and using neuro-musculoskeletal interfaces for more intent recognition. Additionally, integrating deep learning, AI models, intelligent sensors, and haptic feedback on controller designs to make the controller more robust. These advancements aim to enhance motion prediction, stability and balance, making prostheses more efficient and flexible.

Appendix

.1 Proof of Stability

Finite-time convergence and robustness for super twisting control have been proved by geometric approaches or by methods of the homogeneity properties of the algorithm in [65]. Obviously, $s = 0$ and $\dot{S} = 0$ will be obtained when the variables u and u_{aux} converge to zero in t_s . Therefore, stability can be achieved when $t_s \leq t$ and the chattering will be reduced. As a result, we need to demonstrate the convergence of variables u and u_{aux} to zero. The convergence of variables u and u_{aux} was proven in [66] which is recalled in the following theorem.

Matrix $A = \begin{bmatrix} -\frac{\alpha}{2} & \frac{1}{2} \\ -\beta & 0 \end{bmatrix}$ is Hurwitz, all its eigenvalues have negative real parts.

The constant gains are positive, $\alpha > 0$ and $\beta > 0$ for every symmetric and positive definite matrix $Q = Q^T > 0$, therefore, for any positive definite matrix Q , there must be a positive definite symmetric matrix P that satisfies the Lyapunov equation:

$$A^T P + P A = -Q \quad (.1)$$

The algebraic Lyapunov equation (ALE) $A^T P + P A = -Q$ has a unique symmetric and positive definite solution $P = P^T > 0$.

Define the Lyapunov function as $V = \zeta^T P \zeta$, which is a global and strict Lyapunov function for system, and $P = P^T > 0$.

$V_0(\xi)$ Is a continuous positive definite function and is radially unbounded. Therefore, it can be derived by the chain rule.

Let's Define the Lyapunov function as

$$V_0(\xi) = \xi^T P \xi \quad (.2)$$

Where $V_0(\xi)$ a continuous positive is definite function and is radially unbounded. Therefore, $\dot{V}_0$ can be derived by the chain rule derivation below. Assume $x_1 = S$, $x_2 = -\beta \int_0^t \text{sign}(x_1) dt$

Then

$$\xi^T = [\xi_1 \xi_2] = \left[\text{sign}|x_1|^{\frac{1}{2}} x_2 \right] \quad (.3)$$

$$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$C = [1, 0]$$
(.4)

From the derivation of Eq(.3)

$$\dot{\xi} = \begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{2|x_1|^{\frac{1}{2}}} \left[-\hat{\alpha} |x_1|^{\frac{1}{2}} \text{sign}(x_1) + x_2 \right] \\ -\hat{\beta} \text{sign}(x_1) \end{bmatrix} = \frac{1}{|x_1|^{\frac{1}{2}}} (\hat{A}\xi + B)$$

$$\dot{\xi}^T = [\dot{\xi}_1 \dot{\xi}_2] = \left[\frac{1}{2|x_1|^{\frac{1}{2}}} \left[-\hat{\alpha} |x_1|^{\frac{1}{2}} \text{sign}(x_1) + x_2 \right] - \hat{\beta} \text{sign}(x_1) \right]$$
(.5)

Substituting Eq(.3) and the value of A and B, Equation Eq(.5) becomes.

$$\dot{\xi}^T = \frac{1}{|x_1|^{\frac{1}{2}}} (\xi^T \hat{A}^T + B^T)^2$$
(.6)

From the derivation of Eq(.2), we obtain

$$\dot{V}_0(\xi) = \xi^T P \dot{\xi} + \dot{\xi}^T P \xi$$

$$\dot{\xi} = \frac{1}{|x_1|^{\frac{1}{2}}} (\xi^T \hat{A}^T + B^T) P \xi + \xi^T P \frac{1}{|x_1|^{\frac{1}{2}}} (\hat{A}\xi + B)$$
(.7)

Distributing the above Eq(.7)

$$\dot{V}_0(\xi) = \frac{1}{|x_1|^{\frac{1}{2}}} (\xi^T \hat{A}^T P \xi + \xi^T P \hat{A} \xi) + \frac{1}{|x_1|^{\frac{1}{2}}} (B^T P \xi + \xi^T P B)$$
(.8)

Since $\hat{A} = A - \tilde{A}$ Eq (.8) becomes.

$$\dot{V}_0(\xi) = \frac{1}{|x_1|^{\frac{1}{2}}} (\xi^T A^T P \xi + \xi^T P A \xi) - \frac{1}{|x_1|^{\frac{1}{2}}} (\xi^T \tilde{A}^T P \xi + \xi^T P \tilde{A} \xi)$$

$$+ \frac{1}{|x_1|^{\frac{1}{2}}} (B^T P \xi + \xi^T P B)$$
(.9)

$$\text{Let } \dot{V}_0(\xi) = \dot{V}_1(\xi) + \dot{V}_2(\xi)$$
(.10)

Then .9 will be divided in to two parts resulting.

$$\dot{V}_1(\xi) = \frac{1}{|x_1|^{\frac{1}{2}}} (\xi^T A^T P \xi + \xi^T P A \xi) + \frac{1}{|x_1|^{\frac{1}{2}}} (B^T P \xi + \xi^T P B)$$

$$\dot{V}_2(\xi) = \frac{1}{|x_1|^{\frac{1}{2}}} \xi^T (\tilde{A}^T P + P \tilde{A}) \xi$$
(.11)

Given $\tilde{A} = \begin{bmatrix} -\frac{\tilde{\alpha}}{2} & 0 \\ -\tilde{\beta} & 0 \end{bmatrix}$ and $P = \begin{bmatrix} a & c \\ c & b \end{bmatrix}$
positive definite matrix

$$\tilde{A}^T P + P \tilde{A} = \begin{bmatrix} -\tilde{\alpha}a - 2\tilde{\alpha}c - \tilde{\alpha}2c - \tilde{\alpha}b \\ -\frac{\tilde{\alpha}}{2}c - \tilde{\alpha}bc \end{bmatrix} \quad (.12)$$

Substituting the Eq.(12) values on Eq.(11)

$$\begin{aligned} \dot{V}_2(\xi) &= \frac{1}{|x_1|^{\frac{1}{2}}} \left[\text{sign}(x_1) |x_1|^{\frac{1}{2}} x_2 \right] \begin{bmatrix} -\tilde{\alpha}a - 2\tilde{\alpha}c - \tilde{\alpha}2c - \tilde{\alpha}b \\ -\frac{\tilde{\alpha}}{2}c - \tilde{\alpha}bc \end{bmatrix} \begin{bmatrix} \text{sign}(x_1) |x_1|^{\frac{1}{2}} \\ x_2 \end{bmatrix} \\ &= \frac{1}{|x_1|^{\frac{1}{2}}} \left[-\tilde{\alpha}a |x_1| - 2\tilde{\beta}c |x_1| - 2 \left(\frac{\tilde{\alpha}}{2}c + \tilde{\beta}b \right) \text{sign}(x_1) |x_1| x_2 \right] \\ &= -\tilde{\alpha}a |x_1|^{\frac{1}{2}} - 2\tilde{\beta}c |x_1|^{\frac{1}{2}} - 2 \left(\frac{\tilde{\alpha}}{2}c + \tilde{\beta}b \right) \text{sign}(x_1) x_2 \\ \dot{V}_2(\xi) &= -\tilde{\alpha} \left[a |x_1|^{\frac{1}{2}} + c \text{sign}(x_1) x_2 \right] - 2\tilde{\beta} \left[c |x_1|^{\frac{1}{2}} + b \text{sign}(x_1) x_2 \right] \end{aligned} \quad (.13)$$

$$\begin{aligned} \dot{V}_1(\xi) &= \frac{1}{|x_1|^{\frac{1}{2}}} (\xi^T A^T P \xi + \xi^T P A \xi) + \frac{1}{|x_1|^{\frac{1}{2}}} (B^T P \xi + \xi^T P B) \\ &= \frac{1}{|x_1|^{\frac{1}{2}}} (\xi^T A^T P \xi + \xi^T P A \xi + B^T P \xi + \xi^T P B) \\ &\leq \frac{1}{|x_1|^{\frac{1}{2}}} (\xi^T A^T P \xi + \xi^T P A \xi + B^T P \xi + \xi^T P B) \\ &\leq \frac{1}{|x_1|^{\frac{1}{2}}} (\xi^T A^T P \xi + \xi^T P A \xi + \xi^T P B B^T P \xi) \\ \dot{V}_1(\xi) &= \frac{1}{|x_1|^{\frac{1}{2}}} \xi^T (A^T P + P A + P B B^T P) \xi \end{aligned} \quad (.14)$$

Where

$$(A^T P + P A + P B B^T P) = Q \quad (.15)$$

For $A = \begin{bmatrix} -\frac{\alpha}{2} & \frac{1}{2} \\ -\beta & 0 \end{bmatrix}$, $P = \begin{bmatrix} a & c \\ c & b \end{bmatrix}$ and $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Computing Eq.(15) For the above listed values its obtained.

$$\begin{aligned} Q &= \left\{ \begin{bmatrix} -\frac{\alpha}{2} & -\beta \\ \frac{1}{2} & 0 \end{bmatrix} \begin{bmatrix} a & c \\ c & b \end{bmatrix} + \begin{bmatrix} a & c \\ c & b \end{bmatrix} \begin{bmatrix} -\frac{\alpha}{2} & \frac{1}{2} \\ -\beta & 0 \end{bmatrix} + \begin{bmatrix} a & c \\ c & b \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} a & c \\ c & b \end{bmatrix} \right\} \\ &= \begin{bmatrix} \alpha a + 2\beta c - c^2 & \frac{\alpha}{2}c + \beta b - \frac{1}{2}a - bc \\ \frac{\alpha}{2}c + \beta b - \frac{1}{2}a - bc & -b^2 - c \end{bmatrix} \end{aligned} \quad (.16)$$

Since $\dot{V}_0(\tilde{\xi}) = \dot{V}_1(\xi) + \dot{V}_2(\xi)$ Substituting Eq.(16) on this expression its :

$$\begin{aligned} \dot{V}_0(\tilde{\xi}) \leq & -\frac{1}{|x_1|^{\frac{1}{2}}} \xi^T Q \xi + \tilde{\alpha} \left[a |x_1|^{\frac{1}{2}} + \right. \\ & \left. \text{csign}(x_1) x_2 + 2\tilde{\beta} \left[c |x_1|^{\frac{1}{2}} + \text{bsign}(x_1) x_2 \right] \right] \end{aligned} \quad (.17)$$

By further calculation to insure Stability.

$$-\frac{1}{|x_1|^{\frac{1}{2}}} \xi^T Q \xi \leq -\frac{1}{|x_1|^{\frac{1}{2}}} \xi^T \xi = -\frac{1}{|x_1|^{\frac{1}{2}}} \|\xi\|^2 \leq -\|\xi\|_2 \quad (.18)$$

For $V_0(\xi)$, we obtain

$$\lambda_{\min}(P) \|\xi\|_2^2 \leq V_0(\xi) = \xi^T P \xi \leq \lambda_{\max}(P) \|\xi\|_2^2 \quad (.19)$$

Where $\lambda_{\min}(P)$ and $\lambda_{\max}(P)$ the upper and lower eigen values of (P) Then,

$$\left[\frac{V_0(\xi)}{\lambda_{\max}(P)} \right]^{\frac{1}{2}} \leq \|\xi\|_2 \Rightarrow \frac{V_0^{\frac{1}{2}}(\xi)}{\lambda_{\max}^{\frac{1}{2}}(\xi)} \leq \|\xi\|_2 \quad (.20)$$

If $r = \frac{1}{\lambda_{\max}^{\frac{1}{2}}(P)}$, then substituting r in to eq(136)

$$r V_0^{\frac{1}{2}}(\xi) \leq \|\xi\|_2 \Rightarrow \dot{V}_0(\xi) \leq -r V_0^{\frac{1}{2}}(\xi) \quad (.21)$$

Consequently,

$$\dot{V}_0(\xi) \leq -r V_0^{\frac{1}{2}}(\xi) + \tilde{\alpha} \left[a |x_1|^{\frac{1}{2}} + \text{csign}(x_1) x_2 \right] + 2\tilde{\beta} \left[c |x_1|^{\frac{1}{2}} + \text{bsign}(x_1) x_2 \right] \quad (.22)$$

Therefore, if α and β in the above equation are selected to be positive, the SMC can be obtained and becomes asymptotically stable (i.e. sliding variables $\dot{S} = 0$ and $S = 0$ as $t \rightarrow 0$). Also, $\dot{V}_0(\xi) \leq 0$ is guaranteed.

.2 Program for sEMG signal processing

```
sEMG =xlsread('InputsEMG.xlsx','Sheet1');
time = sEMG(:,1);
tensorfascialata=sEMG(:,2);
rectusfemoris=T1_10rm_Bs(:,3);
vastusmedialis=sEMG(:,4);
semimembranosus=sEMG(:,5);
uppertibialisanterior=sEMG(:,6);
lowertibialisanterior=sEMG(:,7);
lateralgastrocnemius=sEMG(:,8);
```

```

medialgastrocnemius=sEMG(:,9);
soleus=sEMG(:,10);
% Lowpass filter 500Hz
Freq = 1500;
NyqFreq = Freq/2;
Lowpass = 500;
Wn = Lowpass/NyqFreq;
[B,A] = butter (4,Wn,'low');
rectusfemoris_low_filtered = filtfilt(B,A,rectusfemoris);
semimembranosus_low_filtered = filtfilt(B,A,semimembranosus);
vastusmedialis_low_filtered = filtfilt(B,A,vastusmedialis);
uppertibialisanterior_low_filterd = filtfilt(B,A,uppertibialisanterior);
lowertibialisanterior_low_filterd=filtfilt(B,A,lowertibialisanterior);
lateralgastrocnemius_low_filterd=filtfilt(B,A,lateralgastrocnemius);
medialgastrocnemius_low_filterd=filtfilt(B,A,medialgastrocnemius);
soleus_low_filterd=filtfilt(B,A, soleus);
tensorfascialata_low_filterd=filtfilt(B,A,tensorfascialata );
% Highpass filter 10Hz
Highpass = 10;
Wo = Highpass/NyqFreq;
[D,C] = butter (4,Wo,'high');

% run filter
high_pass_tensorfascialata_filtdata = filtfilt(D,C,tensorfascialata_low_filterd);
high_pass_BS_BF_1_filtdata = filtfilt(D,C,BS_BF_1_low_filtered);
high_pass_BS_RF_2_filtdata = filtfilt(D,C,BS_RF_2_low_filtered);
high_pass_BS_BF_2_filtdata = filtfilt(D,C,BS_BF_2_low_filtered);
Unrecognized function or variable 'BS_BF_1_low_filtered'.

high_pass_rectusfemoris_filtdata = filtfilt(D,C,rectusfemoris_low_filtered);
high_pass_semimembranosus_filtdata = filtfilt(D,C,semimembranosus_low_filtered);
high_pass_vastusmedialis_filtdata = filtfilt(D,C,vastusmedialis_low_filtered);

high_uppertibialisanterior_filtdata=filtfilt(D,C,uppertibialisanterior_low_filterd);
high_lowertibialisanterior_filtdata = filtfilt(D,C,lowertibialisanterior_low_filterd);
high_lateralgastrocnemius_filtdata = filtfilt(D,C,lateralgastrocnemius_low_filterd);
high_medialgastrocnemius_filtdata = filtfilt(D,C,medialgastrocnemius_low_filterd);
high_soleus_filtdata = filtfilt(D,C,soleus_low_filterd);
Full wave rectification of EMG
filtered_BS_RF_1 = abs(high_pass_BS_RF_1_filtdata);

```

```

filtered_BS_BF_1 = abs(high_pass_BS_BF_1_filtdata);
filtered_BS_RF_2 = abs(high_pass_BS_RF_2_filtdata);
filtered_BS_BF_2 = abs(high_pass_BS_BF_2_filtdata);
%%%
filtered_rectusfemoris = abs( high_pass_rectusfemoris_filtdata);
filtered_tensorfascialata = abs(high_pass_tensorfascialata_filtdata);
filtered_BS_RF_2 = abs(high_pass_BS_RF_2_filtdata);
filtered_BS_BF_2 = abs(high_pass_BS_BF_2_filtdata);
filtered_semimembranosus = abs(high_pass_semimembranosus_filtdata);
filtered_vastusmedialis = abs(high_pass_vastusmedialis_filtdata);
filtered_uppertibialisanterior = abs( high_uppertibialisanterior_filtdata);
filtered_lowertibialisanterior = abs( high_lowertibialisanterior_filtdata);
filtered_lateralgastrocnemius = abs( high_lateralgastrocnemius_filtdata);
filtered_medialgastrocnemius = abs( high_medialgastrocnemius_filtdata);
filtered_soleus = abs( high_soleus_filtdata);
% Linear envelope (2Hz lowpass)
LP = 2;
Wp = LP/NyqFreq;
[F,E] = butter (4,Wp,'low');
linear_rectusfemoris = filtfilt(F,E,filtered_rectusfemoris);
linear_tensorfascialata = filtfilt(F,E,filtered_tensorfascialata);
linear_semimembranosus = filtfilt(F,E, filtered_semimembranosus);
linear_vastusmedialis = filtfilt(F,E, filtered_vastusmedialis);

linear_uppertibialisanterior = filtfilt(F,E, filtered_uppertibialisanterior);
linear_lowertibialisanterior = filtfilt(F,E, filtered_lowertibialisanterior);
linear_lateralgastrocnemius = filtfilt(F,E, filtered_lateralgastrocnemius);
linear_medialgastrocnemius = filtfilt(F,E, filtered_medialgastrocnemius);
linear_filtered_soleus = filtfilt(F,E, filtered_soleus);
set(0, 'defaultFigureToolbar', 'none')
plot(time,linear_filtered_soleus)

```

.3 programs to call another experimental Subject

```

prompt = "What is the mass of the subject? ";
M = input(prompt)
prompt = "What is the Hieght of the subject? ";
H = input(prompt)
Hip                knee                Ankle

```

```

m_1 =0.1*M;          m_2=0.0465*M;          m_ =0.0145*M;
L_1 =0.284*H;        L_2=0.246*H;          L_3=0.152*H;
l_1 =0.433*L_1;      l_2=0.433*L_1;        l_3=0.5*L_3;
k_1 =0.540*L_1;      k_2=0.528*L_2;        k_3=0.528*L_3;
J_1 =m_1*k_1^2;      J_2=m_2*k_2^2;        J_3=m_2*k_2^2;
b_1 =0.1;            b_2=0.1;              b_3 =0.1;

```

.4 Matlab Program for NN Training

```

% sEmgHip - input data.
% ref_HipFlexionAngle_processed - target data.
x = sEmgHip;
t = ref_HipFlexionAngle_processed;
% Choose a Training Function
% 'trainlm' is usually fastest.
trainFcn = 'trainlm'; % Levenberg-Marquardt backpropagation.

% Create a Fitting Network
hiddenLayerSize = 10;
net = fitnet(hiddenLayerSize,trainFcn);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

Train the Network
[net,tr] = train(net,x,t);
% Test the Network
yNN = net(x);
e = gsubtract(t,yNN);
performance = perform(net,t,yNN)
View the Network
Plots
figure, plotperform(tr)
plot(yNN)%hold on
plot(ref_HipFlexionAngle_processed)

```


.5 Simulink model

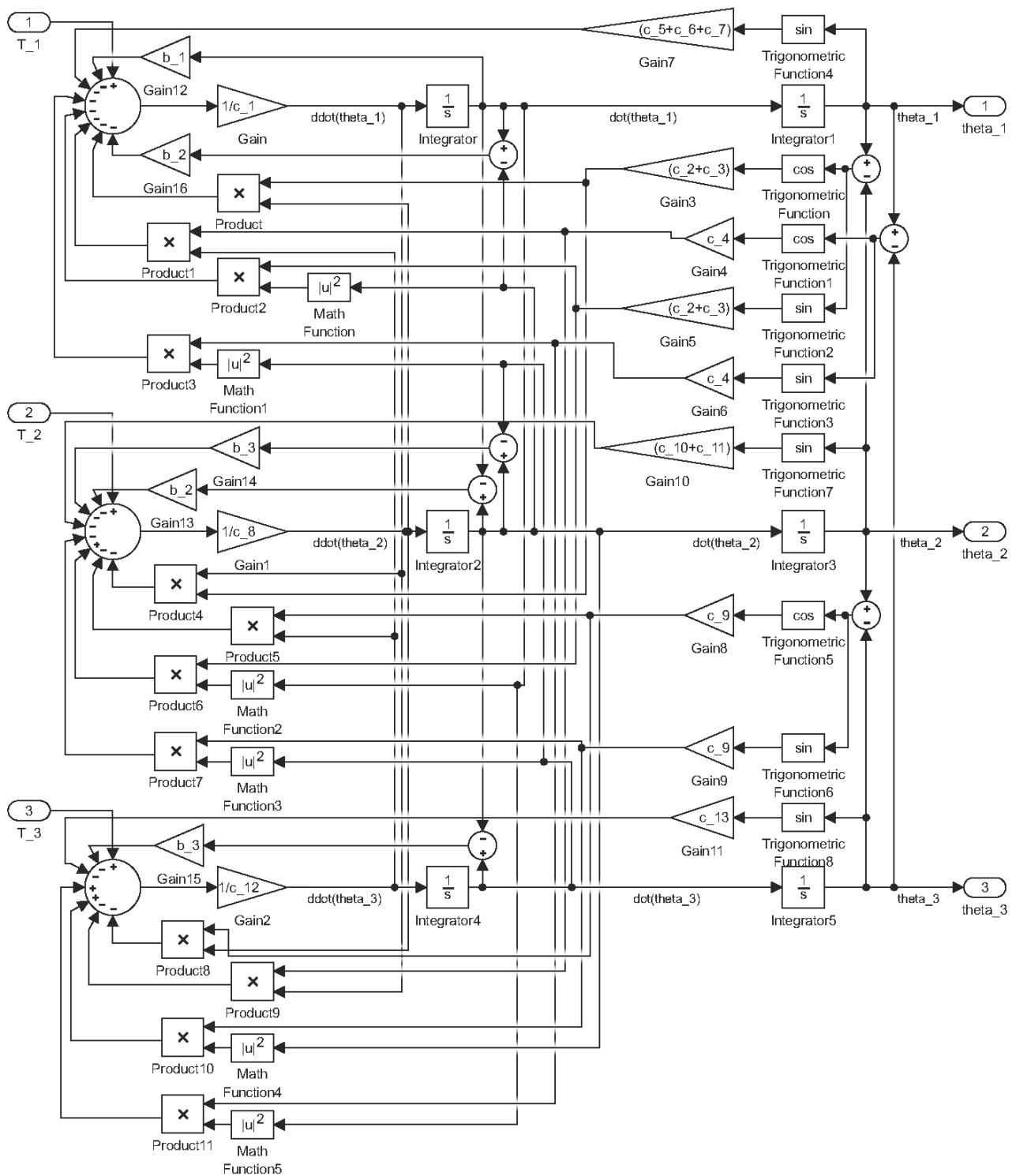


Figure 1 Simulink Model

References

- [1] K. Park, H.-J. Ahn, K.-H. Lee, and C.-H. Lee, “Development and performance verification of a motorized prosthetic leg for stair walking,” *Applied bionics and biomechanics*, vol. 2020, no. 1, p. 8872362, 2020.
- [2] J. Ebnezar and R. John, *Textbook of orthopedics*. JP Medical Ltd, 2016.
- [3] B. Maat, G. Smit, D. Plettenburg, and P. Breedveld, “Passive prosthetic hands and tools: A literature review,” *Prosthetics and orthotics international*, vol. 42, no. 1, pp. 66–74, 2018.
- [4] M. Windrich, M. Grimmer, O. Christ, S. Rinderknecht, and P. Beckerle, “Active lower limb prosthetics: a systematic review of design issues and solutions,” *Biomedical engineering online*, vol. 15, pp. 5–19, 2016.
- [5] A. Nasr, B. Laschowski, and J. McPhee, “Myoelectric control of robotic leg prostheses and exoskeletons: A review,” in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 85444, p. V08AT08A043, American Society of Mechanical Engineers, 2021.
- [6] M. Asif, M. I. Tiwana, U. S. Khan, W. S. Qureshi, J. Iqbal, N. Rashid, and N. Naseer, “Advancements, trends and future prospects of lower limb prosthesis,” *IEEE Access*, vol. 9, pp. 85956–85977, 2021.
- [7] T. Lenzi, M. Cempini, L. Hargrove, and T. Kuiken, “Design, development, and testing of a lightweight hybrid robotic knee prosthesis,” *The International Journal of Robotics Research*, vol. 37, no. 8, pp. 953–976, 2018.
- [8] V. Asanza, E. Peláez, F. Loayza, L. L. Lorente-Leyva, and D. H. Peluffo-Ordóñez, “Identification of lower-limb motor tasks via brain–computer interfaces: a topical overview,” *Sensors*, vol. 22, no. 5, p. 2028, 2022.
- [9] R. Di Gregorio and L. Vocenas, “Identification of gait-cycle phases for prosthesis control,” *Biomimetics*, vol. 6, no. 2, p. 22, 2021.
- [10] N. Roth, A. Küderle, D. Prossel, H. Gassner, B. M. Eskofier, and F. Kluge, “An inertial sensor-based gait analysis pipeline for the assessment of real-world stair ambulation parameters,” *Sensors*, vol. 21, no. 19, p. 6559, 2021.
- [11] E. E. Cabral, G. A. Fregonezi, L. Melo, N. Basoudan, S. Mathur, and W. D. Reid, “Surface electromyography (semg) of extradiaphragm respiratory muscles in healthy subjects: A systematic review,” *Journal of Electromyography and Kinesiology*, vol. 42, pp. 123–135, 2018.
- [12] B. Ahkami, K. Ahmed, A. Thesleff, L. Hargrove, and M. Ortiz-Catalan, “Electromyography-based control of lower limb prostheses: a systematic review,” *IEEE transactions on medical robotics and bionics*, vol. 5, no. 3, pp. 547–562, 2023.

- [13] J. Nieveen, M. Brinton, D. J. Warren, and V. J. Mathews, "A nonlinear latching filter to remove jitter from movement estimates for prostheses," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 12, pp. 2849–2858, 2020.
- [14] T. R. Clites, M. J. Carty, J. B. Ullauri, M. E. Carney, L. M. Mooney, J.-F. Duval, S. S. Srinivasan, and H. M. Herr, "Proprioception from a neurally controlled lower-extremity prosthesis," *Science Translational Medicine*, vol. 10, no. 443, p. eaap8373, 2018.
- [15] S. Shafiu Hasan, M. R. Siddiquee, R. Atri, R. Ramon, J. S. Marquez, and O. Bai, "Prediction of gait intention from pre-movement eeg signals: a feasibility study," *Journal of neuroengineering and rehabilitation*, vol. 17, pp. 1–16, 2020.
- [16] S. Chen, X. Shu, H. Wang, L. Ding, J. Fu, and J. Jia, "The differences between motor attempt and motor imagery in brain-computer interface accuracy and event-related desynchronization of patients with hemiplegia," *Frontiers in neurorobotics*, vol. 15, p. 706630, 2021.
- [17] T. Elery, S. Rezazadeh, E. Reznick, L. Gray, and R. D. Gregg, "Effects of a powered knee-ankle prosthesis on amputee hip compensations: A case series," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 12, pp. 2944–2954, 2020.
- [18] S. Pancholi and A. M. Joshi, "Advanced energy kernel-based feature extraction scheme for improved emg-pr-based prosthesis control against force variation," *IEEE Transactions on Cybernetics*, vol. 52, no. 5, pp. 3819–3828, 2020.
- [19] S. Shi, M. J. Leineweber, and J. Andrysek, "Exploring the tactor configurations of vibrotactile feedback systems for use in lower-limb prostheses," *Journal of Vibration and Acoustics*, vol. 141, no. 5, p. 051009, 2019.
- [20] S. S. Srinivasan, G. Tuckute, J. Zou, S. Gutierrez-Arango, H. Song, R. L. Barry, and H. M. Herr, "Agonist-antagonist myoneural interface amputation preserves proprioceptive sensorimotor neurophysiology in lower limbs," *Science translational medicine*, vol. 12, no. 573, p. eabc5926, 2020.
- [21] S. Raspopovic, G. Valle, and F. M. Petrini, "Sensory feedback for limb prostheses in amputees," *Nature Materials*, vol. 20, no. 7, pp. 925–939, 2021.
- [22] W. Abdallah, R. Boucetta, and S. B. H. Ali, "An adaptive fuzzy-pid control of a lower limb active prosthesis under external perturbations," in *2019 16th International Multi-Conference on Systems, Signals & Devices (SSD)*, pp. 96–101, IEEE, 2019.
- [23] Z. Wu, Y. Chen, Y. Geng, X. Wang, and B. Xuan, "Model-free robust adaptive integral sliding mode impedance control of knee–ankle–toe active transfemoral prosthesis," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 18, no. 3, p. e2378, 2022.
- [24] M. A. Salman and S. K. Kadhim, "Optimal backstepping controller design for prosthetic knee joint," *Journal Européen des Systèmes Automatisés*, vol. 55, no. 1, pp. 49–59, 2022.
- [25] R. Bautista, R. Dubay, J. Carretero, and M. Díaz-Rodríguez, "Pilot study of general predictive control+ integral compensator for powered prosthetic legs," 2015.
- [26] H. Richter, D. Simon, W. A. Smith, and S. Samorezov, "Dynamic modeling, parameter estimation and control of a leg prosthesis test robot," *Applied Mathematical Modelling*, vol. 39, no. 2, pp. 559–573, 2015.
- [27] A. D. Keleş and C. A. Yucesoy, "Development of a neural network based control algorithm for powered ankle prosthesis," *Journal of Biomechanics*, vol. 113, p. 110087, 2020.

- [28] A. Menon, *Neural Prosthetic with In-sensor Shared Control*. PhD thesis, UC Berkeley, 2023.
- [29] A. K. Hamza, *Generating an adaptive and robust walking pattern for prosthetic ankle-foot utilizing a nonlinear autoregressive network with exogenous inputs/Hamza Al Kouzbary*. PhD thesis, Universiti Malaya, 2021.
- [30] A. Loya, S. Deshpande, and A. Purwar, “Machine learning-driven individualized gait rehabilitation: classification, prediction, and mechanism design,” *Journal of Engineering and Science in Medical Diagnostics and Therapy*, vol. 3, no. 2, p. 021105, 2020.
- [31] X. Gao, J. Si, Y. Wen, M. Li, and H. H. Huang, “Knowledge-guided reinforcement learning control for robotic lower limb prosthesis,” in *2020 IEEE international conference on robotics and automation (ICRA)*, pp. 754–760, IEEE, 2020.
- [32] A. H. Shultz and M. Goldfarb, “A unified controller for walking on even and uneven terrain with a powered ankle prosthesis,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 4, pp. 788–797, 2018.
- [33] A. Kawala-Sterniuk, N. Browarska, A. Al-Bakri, M. Pelc, J. Zygarlicki, M. Sidikova, R. Martinek, and E. J. Gorzelanczyk, “Summary of over fifty years with brain-computer interfaces—a review,” *Brain Sciences*, vol. 11, no. 1, p. 43, 2021.
- [34] R. Schulte, “Up to myleg with data: human motor intent recognition using multi-array emg and machine learning,” Master’s thesis, University of Twente, 2019.
- [35] M. Liu, F. Zhang, and H. Huang, “An adaptive classification strategy for reliable locomotion mode recognition,” *Sensors*, vol. 17, no. 9, p. 2020, 2017.
- [36] A. Fleming and H. H. Huang, “Proportional myoelectric control of a powered ankle prosthesis for postural control under expected perturbation: A pilot study,” in *2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*, pp. 899–904, IEEE, 2019.
- [37] P. G. Kulkarni, N. Paudel, S. Magar, M. F. Santilli, S. Kashyap, A. K. Baranwal, P. Zamboni, P. Vasavada, A. Katiyar, and A. V. Singh, “Overcoming challenges and innovations in orthopedic prosthesis design: an interdisciplinary perspective,” *Biomedical Materials & Devices*, vol. 2, no. 1, pp. 58–69, 2024.
- [38] H. Vallery, F. Lachmann, S. van der Helm, A. Pennycott, and G. Smit, “Design and evaluation of the pneumatic leg prosthesis erik to assist elderly amputees with sit-down and stand-up movements,” *Wearable Technologies*, vol. 4, p. e16, 2023.
- [39] M. Goršič, B. Dai, and D. Novak, “Load position and weight classification during carrying gait using wearable inertial and electromyographic sensors,” *Sensors*, vol. 20, no. 17, p. 4963, 2020.
- [40] P. Palumbo, P. Randi, S. Moscato, A. Davalli, and L. Chiari, “Degree of safety against falls provided by 4 different prosthetic knee types in people with transfemoral amputation: a retrospective observational study,” *Physical therapy*, vol. 102, no. 4, p. pzab310, 2022.
- [41] E. Lathouwers, B. Tassignon, A. Maricot, A. Radwan, M. Naeyaert, H. Raeymaekers, P. Van Schuerbeek, S. Sunaert, J. De Mey, and K. De Pauw, “Human-prosthetic interaction (humanit): A study protocol for a clinical trial evaluating brain neuroplasticity and functional performance after lower limb loss,” *Plos one*, vol. 19, no. 3, p. e0299869, 2024.
- [42] M. Asif, M. I. Tiwana, U. S. Khan, W. S. Qureshi, J. Iqbal, N. Rashid, and N. Naseer, “Advancements, trends and future prospects of lower limb prosthesis,” *IEEE Access*, vol. 9, pp. 85956–85977, 2021.

- [43] W. Wei, F. Tan, H. Zhang, H. Mao, M. Fu, O. W. Samuel, and G. Li, “Surface electromyogram, kinematic, and kinetic dataset of lower limb walking for movement intent recognition,” *Scientific Data*, vol. 10, no. 1, p. 358, 2023.
- [44] W. Wei, F. Tan, H. Zhang, H. Mao, M. Fu, O. W. Samuel, and G. Li, “Surface electromyogram, kinematic, and kinetic dataset of lower limb walking for movement intent recognition,” *Scientific Data*, vol. 10, no. 1, p. 358, 2023.
- [45] H. Huang, F. Zhang, L. J. Hargrove, Z. Dou, D. R. Rogers, and K. B. Englehart, “Continuous locomotion-mode identification for prosthetic legs based on neuromuscular–mechanical fusion,” *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 10, pp. 2867–2875, 2011.
- [46] L. González, A. M. López, D. Álvarez, and J. C. Álvarez, “Evaluation of filtering methods for the prediction of human position during walking by means of kinematical models,” in *2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1–6, IEEE, 2021.
- [47] M. M. Madebo, C. M. Abdissa, L. N. Lemma, and D. S. Negash, “Robust tracking control for quadrotor uav with external disturbances and uncertainties using neural network based mrac,” *IEEE Access*, 2024.
- [48] Z. Liu, K. Peng, L. Han, and S. Guan, “Modeling and control of robotic manipulators based on artificial neural networks: a review,” *Iranian Journal of Science and Technology, Transactions of Mechanical Engineering*, vol. 47, no. 4, pp. 1307–1347, 2023.
- [49] M. L. Strydom, A. Banach, J. Roberts, R. Crawford, and A. T. Jaiprakash, “Kinematic model of the human leg using dh parameters,” *IEEE Access*, vol. 8, pp. 191737–191750, 2020.
- [50] M. Cardona, C. E. Garcia Cena, F. Serrano, and R. Saltaren, “Alice: Conceptual development of a lower limb exoskeleton robot driven by an on-board musculoskeletal simulator,” *Sensors*, vol. 20, no. 3, p. 789, 2020.
- [51] J. L. Sarmiento-Ramos, A. F. Meneses-Castro, and P. J. Jaimes-Mantilla, “Dynamic model of lower limb motion in the sagittal plane during the gait cycle,” *Ingeniería*, vol. 29, no. 1, 2024.
- [52] N. Alexander and H. Schwameder, “Effect of sloped walking on lower limb muscle forces,” *Gait & posture*, vol. 47, pp. 62–67, 2016.
- [53] M. G. Pandy and T. P. Andriacchi, “Muscle and joint function in human locomotion,” *Annual review of biomedical engineering*, vol. 12, no. 1, pp. 401–433, 2010.
- [54] D. Stegeman and H. Hermens, “Standards for surface electromyography: The european project surface emg for non-invasive assessment of muscles (seniam),” *Enschede: Roessingh Research and Development*, vol. 10, pp. 8–12, 2007.
- [55] J. Kim, S. Lee, and S. Kwon, “Comparison gait analysis of normal and amputee: Filtering graph data based on joint angle,” *The International Journal of Advanced Smart Convergence*, vol. 12, no. 3, pp. 61–67, 2023.
- [56] K. Igwe, O. Oyedum, A. Aibinu, M. Ajewole, and A. Moses, “Application of artificial neural network modeling techniques to signal strength computation,” *Heliyon*, vol. 7, no. 3, 2021.
- [57] S. Gambhire, D. R. Kishore, P. Londhe, and S. Pawar, “Review of sliding mode based control techniques for control system applications,” *International Journal of dynamics and control*, vol. 9, no. 1, pp. 363–378, 2021.
- [58] Y. Shtessel, C. Edwards, L. Fridman, A. Levant, *et al.*, *Sliding mode control and observation*, vol. 10. Springer, 2014.

- [59] E. A. Teklu and C. M. Abdissa, “Genetic algorithm tuned super twisting sliding mode controller for suspension of maglev train with flexible track,” *IEEE Access*, vol. 11, pp. 30955–30969, 2023.
- [60] E. A. Gedefaw, C. M. Abdissa, and L. N. Lemma, “An improved trajectory tracking control of quadcopter using a novel sliding mode control with fuzzy pid surface,” *PLOS ONE*, vol. 19, 11 2024.
- [61] N. B. Abera, C. M. Abdissa, and L. N. Lemma, “An improved nonsingular adaptive super twisting sliding mode controller for quadcopter,” *Plos one*, vol. 19, no. 10, p. e0309098, 2024.
- [62] B. Derseh, L. Negash, and C. M. Abdissa, “Robust pso tuned fosmc for altitude stabilization and trajectory tracking of agricultural monitoring uav,” *Authorea Preprints*, 2023.
- [63] Y. A. Wendemagegn, W. A. Asfaw, C. M. Abdissa, and L. N. Lemma, “Enhancing trajectory tracking accuracy in three-wheeled mobile robots using backstepping fuzzy sliding mode control,” *Engineering Research Express*, vol. 6, no. 4, p. 045204, 2024.
- [64] J. L. Sarmiento-Ramos, A. F. Meneses-Castro, and P. J. Jaimes-Mantilla, “Dynamic model of lower limb motion in the sagittal plane during the gait cycle,” *Ingeniería*, vol. 29, no. 1, 2024.
- [65] J. A. Moreno and M. Osorio, “Strict lyapunov functions for the super-twisting algorithm,” *IEEE transactions on automatic control*, vol. 57, no. 4, pp. 1035–1040, 2012.
- [66] D. Moreno, S. Pan, L. L. Zeonjuk, R. Islas, E. Osorio, G. Martinez-Guajardo, P. K. Chattaraj, T. Heine, and G. Merino, “B 18 2-: a quasi-planar bowl member of the wankel motor family,” *Chemical Communications*, vol. 50, no. 60, pp. 8140–8143, 2014.