UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG

REAL-TIME FUZZY LOGIC CONTROL OF A SMART MATERIAL ACTUATED, ROBOTIC PROSTHETIC HAND WITH GRIP COMPENSATION

A Master's Dissertation by

Juan-Paul Hynek

School of Mechanical, Industrial and Aeronautical Engineering
The University of the Witwatersrand, Johannesburg
South Africa

25th of February 2021

Supervised by
Prof. Jimoh O. Pedro
Co-Supervised by
Prof. F. Kienhöfer, Ms T. Mangera, Mr A.K. Mohamed

A dissertation presented in fulfilment for the degree of
Master of Science in Mechanical Engineering

Declaration

Title:
Real-Time Fuzzy Logic Control of a Smart Material Actuated, Robotic Prosthetic Hand with Grip Compensation

Author:
Juan-Paul Hynek

Supervisor:
Prof. Jimoh Pedro

Co-Supervisors:
Prof. F. Kienhöfer
Ms T. Mangera
Mr A.K. Mohamed

I Juan-Paul Hynek, a postgraduate student at the University of the Witwatersrand, registered for the degree of Master of Science in Mechanical Engineering in the academic year 2018, declares and submits the following:

- I am aware that plagiarism (the use of someone else's work without their permission and/or without acknowledging the original source) is wrong.
- I confirm that the work submitted for assessment for the above dissertation is my own unaided work except where I have explicitly indicated otherwise.
- I have followed the required conventions in referencing the thoughts and ideas of others.
- The above dissertation has not been submitted before for any degree or assessment at any other university.
- I understand that the University of the Witwatersrand may take disciplinary action against me if there is a belief that this is not my own unaided work or that I have failed to acknowledge the source of the ideas or words.

Signed: .... Date: 25th of February 2021

Signed at: University of the Witatersrand

Abstract

In recent years, several attempts have been made towards the control of a robotic prosthetic-hand for lower-arm amputees. However, these attempts have often been limited as they do not consider the entire coupled system during the development and tuning of the control system, spanning electromyographic signal input to positional and force output of the robotic prosthetic hand. Considering this, a robotic prosthetic-hand model is developed encompassing a back-propagation artificial neural network electromyographic grasp classification model, electromyographic-force regression model, non-linear autoregressive with exogenous excitation shape memory alloy model, anthropomorphic hand mechanism and weighted Mamdani fuzzy logic control system.

The performance of the control system was assessed based on the error in positional and force tracking of the hand in achieving the various grasps and forces as interpreted from the electromyographic signals. Overall, the error in position as well as force evaluated to less than 6% of the setpoint, thereby demonstrating the feasibility of using fuzzy logic in the control of a shape memory alloy actuated robotic prosthetic-hand.

The controllers' success and legitimacy were partly due to the very effective electromyographic grasp classifier which made use of extracted Hudgin's features to yield a postprocessed classification rate of 100%. Additionally, the controllers' ability to track force hinged on the developed electromyographic-force model yielding an acceptable MSE of 0.293N and R^2 of 0.73. Though acceptable, the model yielded a less than expected correlation due to the scattering of electromyographic data points for any singular force recorded. Critical to the angular movement of the robotic prosthetic-hand and consequent position tracking was the highly non-linear shape memory alloy actuator. Characteristics such as pseudo-elasticity, shape memory effect, hysteresis, resistive heating, stress effects and convection were all found to be well captured in the closed-loop non-linear autoregressive with exogenous excitation model, resulting in an RMSE of 0.04. Apart from the aforementioned systems, the anthropomorphic hand itself was modelled with linkages and end-point trajectory mimicking realistic hand movement. The overall system was controlled by a weighted Mamdani fuzzy logic controller. The controller was iteratively tuned and was found to effectively control the robotic prosthetic hand in both movement and force.

The findings and the developed models may be used to develop an embedded system for a robotic prosthetic-hand. Furthermore, the work may be extended to other fields of engineering including smart materials research, bioinformatics, and control systems development.

Keywords: robotic prosthetic hand, grasp classification, electromyographic-force model, shape memory alloy and fuzzy logic control

Dedication

''To my parents and loved ones', without whom this dissertation would not be possible, thank you for your endless support and motivation. My success is a reflection of you"
- Juan-Paul Hynek

Acknowledgements

Firstly, I would like to express endless gratitude to my supervisor Prof. Jimoh Pedro, for his vital guidance, knowledge, support and motivation. He instilled in me a passion for robotics and control as an undergraduate student which fuelled my research path. I appreciate his funding as well as his indirect life lessons, which always left me motivated and encouraged.

I would like to thank my co-supervisors Prof. Frank Kienhöfer, Ms Taahirah Mangera and Mr Abdul-Khaaliq Mohamed, for their continuous support, knowledge and willing guidance throughout my endeavour to complete this dissertation. Their expertise in mechanics, bio-mechanics and electronics aided in the completion of this dissertation.

Furthermore, I thank Prof. Benita Olivier, Mr Jaco Liebenberg and Mrs Monique Keller from the School of Physiotherapy, for access to the Movement Lab, aid in setting up and use of the electromyographic equipment, knowledge on the roles of forearm muscles and support in the palpation of the forearm muscles. Their generosity and insight made electromyographic data collection pleasant and facilitated the development of models presented in Chapters 3 and 4.

Lastly, I thank my parents, siblings, girlfriend and friends for their endless support and motivation throughout my undergraduate and postgraduate studies. I am grateful for their willingness to listen and give advice on the issues I faced personally and academically.

Contents

List of Figures

List of Tables

Nomenclature

Acronym	**Meaning**
ADAM	Adaptive Movement Estimation
ADC	Analogue to Digital Converter
AFC	Active Force Control
ANFIS	Adaptive Neuro-Fuzzy Inference System
ANN	Artificial Neural Network
ANOVA	A One-Way Analysis of Variance
AP	Action Potential
AUC	Area Under Curve
BCC	Body Centred Cubic
BMI	Body Mass Index
BPANN	Back-Propagation Artificial Neural Networks
CE	Contractile Element
C-LSTM	Combination of Convolutional Neural Network and Long Short-Term Memory
CMRR	Common Mode Rejection Ratio
CNN	Convolutional Neural Network
CNS	Central Nervous System
DA	Discriminant Analyses
DC	Direct Current
DIP	Distal Interphalangeal
DoF	Degrees-of-freedom
DT	Decision Tree
EMG	Electromyographic
FCC	Face Centred Cubic
FEA	Finite Element Analysis
FFT	Fast Fourier Transforms
FIS	Fuzzy-Inference System
FL	Fuzzy-Logic
FLC	Fuzzy-Logic Control
FMG	Force Myography
FN	False Negative

FP	False Positive
FSR	Force Sensitive Resistor
GA	Genetic Algorithm
GO	Grasp Object
GPR	Gaussian Process Regression
GR	Grasp Recognition
GUI	Graphical User Interface
Gvis	Grip Visualisation
HCP	Hexagonal Close Packed
HD	High Density
HREC	Human Research Ethics Committee
iEMG	Indwelling Electromyographic
KNN	k-th Nearest Neighbour
LDA	Linear Discriminant Analysis
LM	Levenberg-Marquardt
LSTM	Long Short-Term Memory
MAV	Mean Absolute Value
MAVS	Mean Absolute Value Slope
MCC	Matthews Correlation Coefficient
MCP	Metacarpophalangeal
MCS	Myoelectric Control System
MDF	Median Frequency
MDG	Master Data Glove
MISO	Multi-Input-Single-Output
ML	Multilayer
MLP	Multi-Layer Perceptron
MNF	Ratio of Mean Frequency
MP	Metacarpal
MSE	Mean Squared Error
MU	Motor Unit
MUAP	Motor Unit Action Potential
MVC	Maximum Volumetric Contraction
NARX	Nonlinear Autoregressive Exogenous
Nitinol/NiTi	Nickel-Titanium Alloy
NLR	Non-linear Logistic Regression
NMSE	Normalised Mean Squared Error
NN	Neural Network
NRMSE	Normalized Root Mean Square Error
PCA	Principal Component Analysis

PD	Proportional Derivative
PE	Parallel Elastic
PID	Proportional Integral Derivative
PIP	Proximal Interphalangeal
PL	Power-Line
PLA	Polylactic Acid
PSO	Particle Swarm Optimization
R	Correlation Coefficient
ReLU	Rectified Linear Unit
RMS	Root Mean Square
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
ROC	Receiver Operation Characteristic
RPH	Robotic Prosthetic Hand
RPM	Return Point Memory
RT	Regression Tree
SCG	Scale Conjugate Gradient
SE	Series Elastic
sEMG	Surface Electromyographic
SENIAM	Surface Electromyography for the Non-Invasive Assessment of Muscles
SISO	Signal-Input-Signal-Output
SMA	Shape Memory Alloy
SME	Shape Memory Effect
SNR	Signal-to-Noise Ratio
SSC	Slope Sign Change
SSE	Sum of Squared Estimate of Errors
SVM	State Vector Machine
TN	True Negative
TP	True Positive
WL	Waveform Length
ZC	Zero Crossing

Symbol	Meaning
α	Reference point
β	Reference point
ΔH	Heat transformation constant
$\dot{\sigma}$	Stress rate
$\dot{\varepsilon}$	Strain rate
$\dot{g}$	Rate of heat generation

γ	Reference point
ρ	Density of the SMA wire
σ_f	Detwinning Stress
σ_{lower}	Stress at which pseudo-elasticity occurs in reverse phase transformation
σ_s	Twinned Stress
σ_{upper}	Stress at which pseudo-elasticity occurs in forward phase transformation
Θ	Transformation function
ε	Infinitesimal strain tensor (Green strain)
ξ	Martensite/Martensitic fraction
a_A	Material constants
A_f	Final Austenitic Temperature
a_M	Material constants
A_s	Austenite Starting Temperature
b_A	Material constants
b_M	Material constants
c	Specific heat of the wire
$\ddot{q}_{bar}$	Reference joint acceleration
$\dot{q}_{bar}$	Reference joint velocity
E_A	Austenitic Modulus
E_{gen}	Internal energy generated
E_{in}	Energy entering the control volume
E_M	Martensitic Modulus
E_{out}	Energy leaving the control volume
E_{stored}	Energy stored within the control volume
f	Arbitrarily chosen activation functions for the NARX model structure
g	Arbitrarily chosen activation functions for the NARX model structure
h	Mean convective coefficient for the wire
H_e	Maximum difference between reverse and forward phase transformation
K_D	Proportional gain
K_P	Derivative gain
LU^n	Linguistic value, where n = 1, 2, 3, ...
m	Mass of wire
M_f	Final Martensitic Forming Temperature
M_s	Martensite Starting Temperature
q	Heat flux
q_{bar}	Reference joint trajectory
S	Surface area of the wire
T_∞	Bulk fluid temperature

T_w	Temperature of wire
u	Specific internal energy
$X(t)$	Exogenous Input in a NARX neural network model structure
$Y(t)$	True Output of a nonlinear autoregressive exogenous NN model structure
$Y^*(t)$	Estimated Output of the NARX NN model structure

Chapter 1

Introduction

The use of one's hands whilst completing everyday tasks is undoubtedly invaluable; therefore this dissertation explores the control required in developing a novel Shape-Memory Alloy (SMA)-actuated Robotic Prosthetic Hand (RPH) with grip compensation. To achieve this, the RPH system is broken down into its subsystems namely, Electromyographic (EMG) grasp classification, EMG-force compensation, Fuzzy-Logic (FL) actuator control, SMA actuator modelling and the RPH model. These subsystems are finally integrated seamlessly in both a hardware and software capacity to present a feasible RPH solution. Novel to this dissertation is the development and refinement of control related topics inherent to the subsystems and the complete integration thereof. The unique SMA model compliments the developed control system.

1.1 Research Question

What is the real-time simulated performance associated with the indirect adaptive fuzzy-logic control of a novel shape memory alloy-actuated robotic prosthetic hand with force-based grip compensation?

1.2 Problem Statement

The control of an SMA-actuated RPHs is complex and highly non-linear as a result of actuator hysteresis, pseudoelasticity and SMA effects. These issues are compounded with inadequate EMG grasp classification and force-based grip compensation. This results in overall poor performance, reliability and accessibility of SMA actuated RPHs with grip compensation. The black-box model typically used to account for SMA actuator dynamics fails to present a simple, predictable model which can be used in simulations for applications containing SMA actuators. Signal contamination, feature selection and the type of Myoelectric Control System (MCS) used are all root causes for unsatisfactory EMG grasp classification.

Similarly, the performance of force-based grip compensation suffers as it introduces grasp instability. Once combined with the RPH mechanism, these subsystems are coupled and interact non-linearly, requiring a robust overseeing non-linear control system. In response to these problems, this research dissertation aims to produce novel solutions for each subsystem and the integration thereof. Investigating solutions in the form of

Back-Propagation Artificial Neural Networks (BPANNs) for EMG grasp classification, Artificial Neural Network (ANN) regressors for force-based grip compensation, ANN system identification for SMA actuator modelling, Mamdani Fuzzy-Logic Control (FLC) as an intuitive hierarchical controller and simplified kinematic RPH models (see Figure 1.1).

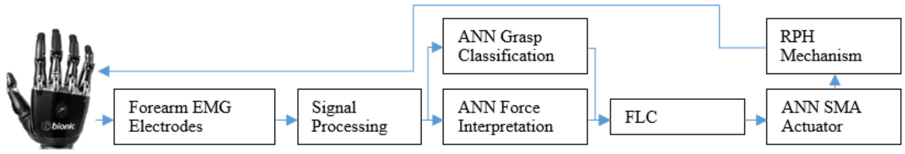


Figure 1.1: Control system of a smart material actuated robotic prosthetic hand with grip compensation.

1.3 Research Background

Non-congenital amputations below the elbow are commonly attributed to severe injury [1], typically caused by motor vehicle and work accidents. Apart from injury-related causes, poor circulation in the form of peripheral vascular disease, and infections that worsen over time due to diabetes, may also result in amputations [2]. The number of lower arm amputees is notably large, leading to a lack of accessibility of prosthetics, particularly in developing countries. Terminal prosthetic devices, common after wars, are either of mechanical or electromechanical design, namely Bowden cable, myoelectric, and switch control [3]. Myoelectric prosthetics are preferred as they offer better dexterity, reliability and function. RPHs like the BeBionics, I-Limb, Michealangelo and Otto Block exist, but not without several drawbacks. Aside from adversities associated with cost, which is addressed in the Touch Hand 3 [4], complexities in the non-linear behaviour of RPHs make it challenging to control. One such subsystem control-related issue is the identification of intended grasps through the acquisition of EMG signals from the forearm.

Noting that electromyography is the process of acquiring signals related to the activity of motor units (MUs) using electrodes, it is easy to see that the technique may be subject to noise and variations due to physiology. This makes the identification of signal and grasp pairs complex and unintuitive, often requiring pre-processing in the form of high-order filters and spectral analysis [5]. Muscle density, body-fat percentage and fatigue are all physiological variations resulting in differences in EMG amplitudes strengthening the need for Maximum Volumetric Contraction (MVC) normalisation. To analyse EMG signals effectively, it is crucial to select the right features to extract, whether it be time-based or frequency-based. Some standard features used in grasp classification include Root Mean Square (RMS), Mean Absolute Value (MAV), Mean Absolute Value Slope (MAVS), Slope Sign Change (SSC), Waveform Length (WL), Zero Crossing (ZC), Fast Fourier Transforms (FFT) and entropy [6, 7]. Similarly, ANNs [8], State Vector Machines (SVMs) [9, 10, 11], Discriminant Analyses (DAs) [12], Fuzzy-Inference Systems (FISs) [13] and Adaptive Neuro-Fuzzy Inference Systems (ANFISs) [14] are typically used in the classification of EMG signals. Literature suggests that ANN-based classifiers with time-based features show the most promise. Another control-related matter is grip compensation within RPHs.

EMG-based grip force compensation originated through the development of hybrid position/force control of manipulators in the 1970's. It involves the use of pressure-sensitive sensors to provide feedback in a closed-loop control system for force-based adjustments dependent on the intended EMG intensity. The relationship between these two parameters, namely force and EMG intensity is proportionally viewed as linear, however opposing views in literature does not validate or disprove this, as considerable data variability exists. Popular regression models frequently used in similar applications include spline interpolation, ANNs, SVMs, Decision Trees (DTs) and Gaussian Process Regression (GPR) [15]. For regressor models to adequately generalise, the boundary environment needs to be carefully selected. Additionally, the single most significant cause for concern with data validity is EMG sensor placement on the target muscle, as its position has a dramatic effect on the acquired EMG signal intensity [15]. Simulating the function of muscles, RPH actuation is responsible mainly for the dexterity, grip force and the movement of RPHs.

First conceptualised in 1919 in a book titled "Ersatzglieder und Arbeitshilfen" (Replacement links and work aids: For war disabled and those injured in accidents); externally powered prosthetic devices gained massive traction. Currently, the most used form of actuation in RPHs are servos and DC motors. Being housed primarily in the socket for most actuators, this form of actuation can achieve human-like grip forces when aided with a gear mechanism. Servo and DC motor actuation are however bulky, power-density deficient and inadequate for situations where a socket is not available. Smart material and SMA actuators provide an alternative solution. SMA actuators initially employed in the aeronautical industry, made use of the thermo-mechanical behaviour of materials such as Nitinol (Nickel-Titanium alloy) to produce a displacement for a given activation temperature. These actuators produce muscle-like movement, but like most thermo-based systems suffer from hysteresis nonlinearities. Predicting the displacement of the actuator is, therefore, complex. SMA actuator system identification models exist but for the most part, only consider the open-loop representation or rely on complex material property descriptions [15], which offer no use during simple closed-loop implementation, simulation and generalisation. Despite these complexities, SMA actuators are power-dense, therefore take up much less space compared to servos and DC motors. Common SMA actuator system identification methods include, FL-based Takagi-Suegeno [16], ANN, state-space [15], Non-linear Autoregressive Exogenous (NARX) [17] and Hammerstein-Weiner [18].

Bearing in mind the mentioned subsystems of RPHs, it is needless to say that an intelligent control system is required to manage the interactions. Hard computing controllers such as Proportional Integral Derivative (PID), although being shown as adequate in individual subsystems [19], do not promote constructive and intuitive supervisor tuning needed for highly non-linear systems. This is where intelligent, soft computing controllers such as Neural Networks (NNs) and FIS excel, all while maintaining or improving on the performance achieved by PID controllers [20]. FIS developed by Lotfi Zadeh is a powerful control technique that aims to construct a fuzzy control surface bounded by real-world limitations by exploiting vague system principles.

1.4 Research Motivation

The need to study and advance research regarding life-changing devices like RPHs is self-evident when considering the impact it can make in the everyday lives of lower-arm amputees. The number of lower-arm amputees increases annually, resulting in the demand for prosthetics, particularly of the robotic variety, to proportionally increase, often placing a strain on production. Additionally, RPHs are tailor-made, with trained control systems unique to each amputee, making lead-times excessively cumbersome. Effort towards mastering anthropomorphic-shaped RPHs, not only opens the door to advancements in prosthetic devices but humanoid robots that can work in industry. Looking at the control and design of RPHs, properties like dexterity (number of the degrees-of-freedom (DoFs)), grip strength, actuation time, audible noise, weight, energy-density and end-point tracking are all functional specifications. From literature, these specifications are never effectively achieved in balance with one another, which tends to result in an RPH with good performance in specific criteria whilst performing poorly in other criteria.. This issue stems primarily from focusing only on individual subsystems that constitute an RPH and the control thereof and not the system as a whole. One such example is designing an RPH without considering control-related inputs. It is not enough to bound the control of an RPH to actuator control, but instead, it is necessary to broaden the scope to include EMG control, force control and overall mechanism control.

On the topic of EMG classification, there are numerous improvements and variations that may constitute a better performance. The task of grasp classification is unique to the type of grasps being investigated and as such RPHs with different grasp capabilities require different classification models. Thus, for a novel RPH, such as the one described in this dissertation, a unique classifier is required. The classifier will build on existing knowledge and further the understanding of EMG grasp classification, mainly related to the chosen pre-processing and processing techniques used. Closely linked to this is the need for a solution to the lack of force-based compensation in RPHs.

EMG force-grip compensation needs to be investigated because the benefits are vast. This includes enabling anthropomorphic RPHs with the ability to be delicate at the same time reliable, to be adaptive and allow more significant user interaction. RPHs currently available allow the user to set the grip strength, however, does not proactively adapt the grip strength dependent on the users' intention. To reduce the rejection rate of RPHs, it is thus necessary to embody these inanimate objects with human-like force responsive characteristics. Mimicking how a hand grips an object and has a smooth and fluidic motion, advancements in RPHs need to be geared towards achieving this. Proposals to use SMAs to achieve better RPH motion, reduce audible noise, increase power-density and maintain grip strength is shown to be successful.

SMAs do possess complexities that are still mostly unexplored with regards to applications involving RPHs. Therefore to advance RPH technology as a whole and potentially use it in the manufacturing of RPHs, strides need to be taken to achieve better SMA performance through control. This includes taking a look at the type of controllers for SMAs, the SMA model (which remains scarce in literature) and the interaction SMAs have in terms of position and force within an RPH mechanism. In terms of controlling SMA actuators, it is hypothesised that FLC will make the task of controlling and tuning the control system more manageable and more intuitive. To investigate the performance of SMAs in RPH applications, it is necessary to have an accurate plant model and the dynamics associated with it.

Existing kinematic RPH models reduce the complexities of a human hand to a sum of joints and links, being mathematically rigorous when introducing external forces. These types of models are crude but crucial as the development of a feasible control system is dependent on accurate system dynamics. Moreover, complex and better RPH models depending on vision tracking and magnetic resonance imaging exist but is not an appealing solution when considering simulation implementation. Therefore simple, accurate, and simulation-suited RPH models needs to be investigated for realistic RPH performance and control.

1.5 Research Objectives

The following chronological list of research objectives is in line with the identified subsystems detailed in sections 3 through 7. These objectives promote the control of a suitable RPH solution.

- Investigate, design and optimise a suitable grasp classifier for identifying and interpreting common everyday grasps.
- Assess and minimise the grasp classification error and benchmark it with similar published classifiers.
- Present a preliminarily design and prototype for a feasible alternative to available Surface Electromyographic (sEMG) systems.
- Investigate and present a regressor-based model, describing the relationship between upper-arm EMG intensity and force.
- Evaluate the performance and minimise the EMG-force regressor model error through validation and tuning.
- The system identifies and presents a cost-minimal model that represents the non-linear contraction/expansion thermal related properties found in SMA actuators.
- Present a simple and accurate RPH kinematic model that can be used in conjunction with SMA actuators to achieve interpreted grasps.
- Explore and validate the use of adaptive FLC of SMA actuators in an RPH model.
- Evaluate the FLC performance and minimise the FLC trajectory tracking error of the RPH mechanism end-points.

1.6 Dissertation Outline

The aim of providing a complete solution to the control-related issues inherent in an SMA-actuated RPH with grip compensation, can only be achieved by breaking down an RPH system and its control into constituent subsystems. Following the literature review, ethics and design development, the subsystems, namely EMG grasp recognition (GR), EMG-force model, SMA actuator model, RPH model and FLC system, compose the body and crux of this dissertation. Each of these sections focuses on individual studies, with corresponding apparatus, approach, methodology, results and discussion. These subsystems are integrated and the overall system performance evaluated in the latter part of Chapter 8. Moreover, the introductory section, subsystem sections and concluding section are composed as shown below:

- Chapter 1 starts by introducing the problem this dissertation intends on investigating and solving. Some background information follows, laying the foundation for motivation and objectives.
- Chapter 2, takes a look at literature, equations and past attempts as it relates to the ideas disseminated in this dissertation.
- Chapter 3 explores the ethical methodology and scope associated with human EMG and force testing and examines RPH design related specifications, constraints and criteria. A model overview concludes this chapter, setting the blueprint on how the subsystems fit together.
- Chapter 4 deals with the complexities associated with EMG signal conditioning and classification. Various classifiers and hyper-parameters are iteratively investigated and summarised, along with the process for achieving classification success.
- Chapter 5 is focused on determining a curve-based relationship between EMG intensity and the force exerted by a human hand. This will be used to provide adaptive force control, actualising the users' force intentions. Various regressor models are put to the test and a chosen solution further optimised.
- Chapter 6 develops a comprehensive SMA model through system identification. Many aspects deemed necessary are accounted for in the model dynamics, namely power input, load, position and power modulation frequency. A new closed-loop, simulation-ready model, is presented and its performance evaluated against identified models.
- Chapter 7 is guided by the control required to achieve the grasps set out by the users EMG signals. To do so, an intelligent controller is developed to minimise the force and positional error affiliated with the actuator and gripping. The interactions and performance of the subsystems are evaluated as a whole with respect to the developed control system.
- Chapter 8 concludes the dissertation and the ideas contained within it. Drawing to conclusion the performances of the subsystems and complete system.

Chapter 2

Design and Concept Development

Ethical clearance, performance and functional specifications are established before commencing with any experimental procedures contained within this dissertation. As per the University's requirements, any research activity involving human participants requires ethics clearance. The application and supporting documents submitted to the Human Research Ethics Committee (HREC) may be found in Appendix 10, with clearance being granted. Performance and functional specifications detail a collection of certain information related to the engineering requirements, constraints and criteria. These specifications are crucial in assessing the performance of the subsystems and system as a whole, see section 3.3, the model overview.

2.1 System Performance and Functional Specifications

Performance and functional specifications define the required achievable system goals and functions, and frequently how it is achieved. These are often broadly customer-specified but subsequently segmented to provide quantitative and qualitative measures by which the solution is accessed. Critical to the control engineering development process, the performance specifications listed in the following subsections are rooted in literature, where the appropriate reference is given. Also, functional specifications define the purpose, process and method of a system. The subsections of section 3.2 deal with the performance and functional specifications for the subsystems detailed in sections 4 through 8.

2.1.1 Electromyographic grasp recognition model

With the development of NN pattern recognition models, it is necessary to have performance specifications in place that govern the training, validation and stop criteria. The risk of converging to a local minimal solution is vastly increased with complexity; thus, it remains imperative to reduce or segment the NN, ensuring a minimal global solution. Additionally, smaller networks inherently reject over training, has better generalisation capabilities and requires less memory and computation time. An acceptable model has performance and functional characteristics resulting in adequate training, testing and validation class classification accuracy. It also should (i) have robust disturbance rejection, (ii) have low false acceptance rates and (iii) be simple such that it dynamically maximises the number of useful parameters. Cognisant of this, the quantitative and qualitative EMG GR performance and functional specifications are as follows:

- The classifier must accommodate six grasp classes excluding rest, namely 3 jaw chuck pinch, cylindrical grasp, hook grasp, lateral pinch, spherical grasp and tip pinch.
- The number of neurons and hidden layers must be optimally minimal to avoid overtraining and maintain generalisation.
- The number of training epochs must remain minimal as it leads to longer computation time and adversely affects testing and validation accuracy.
- The model must be robust, reject noise and artefacts employing pre-processing and maintaining outliers in the dataset.
- Layer weights and biases must be fixed to the same non-zero starting values when investigating NN trials.
- The Normalised Mean Square Error (NMSE) of the NN classifier model must be less than 0.01 [21].
- The combined overall training, testing and validation classification accuracy must exceed 90%, and the accuracy of each class must exceed 85% [21].
- The Area Under Curve (AUC) of the Receiver Operation Characteristic (ROC) must be greater than 0.85 for each classification class [21].
- The trained NN model must be capable of processing in excess of 20, 80ms observations per second to mimic human response time [21].
- The trade-off between precision and recall must yield an F-measure approaching unity.
- The resultant Matthews Correlation Coefficient (MCC) must be positive and approach unity.
- It must be possible to deploy and embed the developed NN classifier model on a microcontroller.

2.1.2 Electromyographic-force model

Regressor models are widely used, interpretable and serve as the basis for many other input-output models. Not surprisingly it is used in signal-input-signal-output (SISO) system identification. Conversely, the absence of publications establishing the relationship between the EMG intensity recorded from the isometric contraction of forearm muscles, and the force generated by the hand prompts the use of statistically-driven performance specifications. Good-fit measures such as the Root Mean Squared Error (RMSE), the sum of squares due to error (SSE), (R^2), residuals and confidence intervals make up the standard statistical tools used to evaluate regressor model performance. Detailing these measures, RMSE is the absolute mean deviation from the predicted response. Likewise, R^2 defines the variance in the dependent variable as a result of independent variable changes, whilst SSE gives a sense of the expected discrepancy between the two. Finally, residuals and confidence intervals describe the uniformity and uncertainty inherent to the regressor model. With this in mind, the quantitative and qualitative EMG-force regressor model performance and functional specifications are as follows:

- The regressor model must linearly describe the relationship between forearm EMG intensity and grip force.
- The data collection process must be visually verifiable to safeguard data validity.
- The number of neurons and hidden layers must be minimal to prevent outlier data from significantly impacting the model training process.
- The RMSE of the model must be below 1.5N, thus representing good generalisation capabilities amid data variability [22].
- The R^2 of the model must be above 0.85, indicating little in data variance around the line of best fit [22].
- The model residuals must be homogeneously scattered around the zero axis, showing unbiased usable data.
- The SSE must be below two, and the data must lie within the 95% confidence interval [22].
- The data collected and the subsequent model must correspond to the operating force range of 0N-9.81N whilst being normalised with respect to EMG intensity.
- The developed NN model must be capable of processing in excess of 20, 80ms observations per second [22].
- It must be possible to deploy and embed the developed NN regressor model on a microcontroller.

2.1.3 Shape memory alloy actuator model

NARX NNs make use of a buffered input and feedback signal in order to forecast a response. The efficacy of NARX NN truly becomes apparent when taking into account time-dependent applications. One such application is system identification of SMA actuators, which are notoriously transient due to heating/cooling cycles and hysteresis. Apart from evaluating the performance specifications of the model using statistical means, i.e., NMSE and R, it is necessary to include time-domain characteristics encompassing speed of response, stability and steady-state error. Also, the autocorrelation of the targets and crosscorrelation of input and target input needs to be within the prescribed confidence intervals, and the significant input and feedback lag identified. With functional specifications, consideration is given to the error encountered in displacement, velocity and acceleration tracking. Shifting to functional specifications, properties required from the SMA actuator include power consumption, peak and continuous force production and heat dissipated during and/or after resistive heating. Thus, the quantitative and qualitative SMA actuator model performance and functional specifications are as follows:

- The model must account for the transient dynamics present in the SMA actuator; these include heat dissipation, resistive heating, hysteresis, pseudo-elasticity and the SMA effect for the given configuration.
- The data range must encompass a broad bandwidth such that the controlled motion is fluid and non-jerky.
- The number of neurons and hidden layers must be optimally minimal.
- The minimum significant lags for both input and feedback delay signals must be obtained.
- The model must be ready to implement in a simulation environment to ensure a closed-loop form.
- The NMSE of the SMA actuator model must be below 0.01 [23].
- The R of the model must be greater than 0.95, representing a strong direct correlation between model response and actual data [23].
- The NN SMA actuator model must be capable of processing in excess of 20, 80ms observations per second [23].
- It must be possible to deploy and embed a BlackBox equivalent SMA actuator model on a microcontroller.

2.1.4 Robotic prosthetic hand model

Anthropomorphic RPHs aim to achieve the dexterity and appeal of normal hands whilst reducing the complexity associated with them. To do this, considerable attention is given to the action of the joints and linkages and the DoFs they represent. In a 10 DoF RPH, which is presented in this dissertation, the action of the distal interphalangeal (DIP) joint of each finger is assumed to bear no significance in the task of gripping. It is therefore fixed in a set position; as such only, the proximal interphalangeal (PIP) and metacarpophalangeal (MCP) joints are mechanically represented and specified. Functionally, these representations need to be constrained to have the same range of motion and grasping potential as a normal hand would have. Also, specifications such as friction, damping and density of the links need to be presented accurately to reflect the dynamics of an RPH. An accurate representation of an anatomically-based RPH results in a realistic control solution that is viable beyond a simulation environment. The performance specifications represent the trajectory tracking and steady-state error stemming from inertial effects. Supplementary to this, the resultant delay in response should be minimal in affecting the controller stability and performance. The type of feedback element and its corresponding dynamics should not impact controller performance. The quantitative and qualitative RPH model performance and functional specifications are as follows:

- The RPH model must consist of 10 DoFs [24].
- The model and design must account for the range of usual motion of a normal hand.
- Density, friction, damping, together with gravitational effects, must be accounted for in the simulation model.
- The RPH must be steady-state stable.
- The RPH must have joint angle and force feedback sensing elements with saturation bounded outputs.
- RPH inertial-based response delay must be considered during controller development.

2.1.5 Fuzzy logic control system

As typical with controller performance geared towards achieving a desired dynamic response over a given operating range, a time and frequency-domain evaluation route are taken. Cognisant of experimental and published values, time-based objectives are set for measures such as maximum overshoot, peak time, time to first zero error, settling time, rise time, delay time, decay ratio and steady-state error. These measures frequent in the analysis of the overdamped, underdamped or critically damped trajectory tracking error. Besides the measures as mentioned earlier, frequency-based objectives such as bandwidth and cutoff rate seek to optimise the range of excitation frequencies beyond which saturation and attenuation occur. Implementing these concepts in an FLC system means intuitive decisions must be made with respect to the: operating range, number of classes, membership functions, rule-base and defuzzification method, leading to a stable, noise attenuating, disturbance rejecting and robust controller. Along these lines, the quantitative and qualitative FLC performance and functional specifications are as follows:

- The stability of the FLC must be guaranteed during the control of the SMA actuator over the specified operating range.
- The FLC must be able to reject disturbance.
- The FLC must be able to attenuate noise.
- The position, velocity and acceleration tracking error of the FLC must be the least possible, and the NMSE must be less than 0.01 [25].
- The FLC must be simple and reduce the classes required so that the coupled rules are narrowed.

- In-line with published performance specifications, for a step input of 30°[26, 27]:
  1. The maximum overshoot must not exceed 5°.
  2. The peak time must not exceed 0.1s.
  3. The time to first zero error must be less than 0.01s.
  4. The response must settle to within 95% of the steady-state value within 1s.
  5. The rise time must be less than 0.08s.
  6. The delay time must not exceed 0.05s.
  7. The decay ratio must not surpass 0.3.
  8. The steady-state error must not be greater than 1.5°.
  9. The bandwidth range and cutoff rate must promote a fast response time and reduce noise susceptibility in the system.

2.2 Control System and Model Overview

The core of this dissertation, as shown in Figure 2.1, the model overview, focuses on the control-related subsystems and how they fit together. Coherence and detailed attention is given to the integration of the subsystems, primarily centralised around points of interest in control theory.

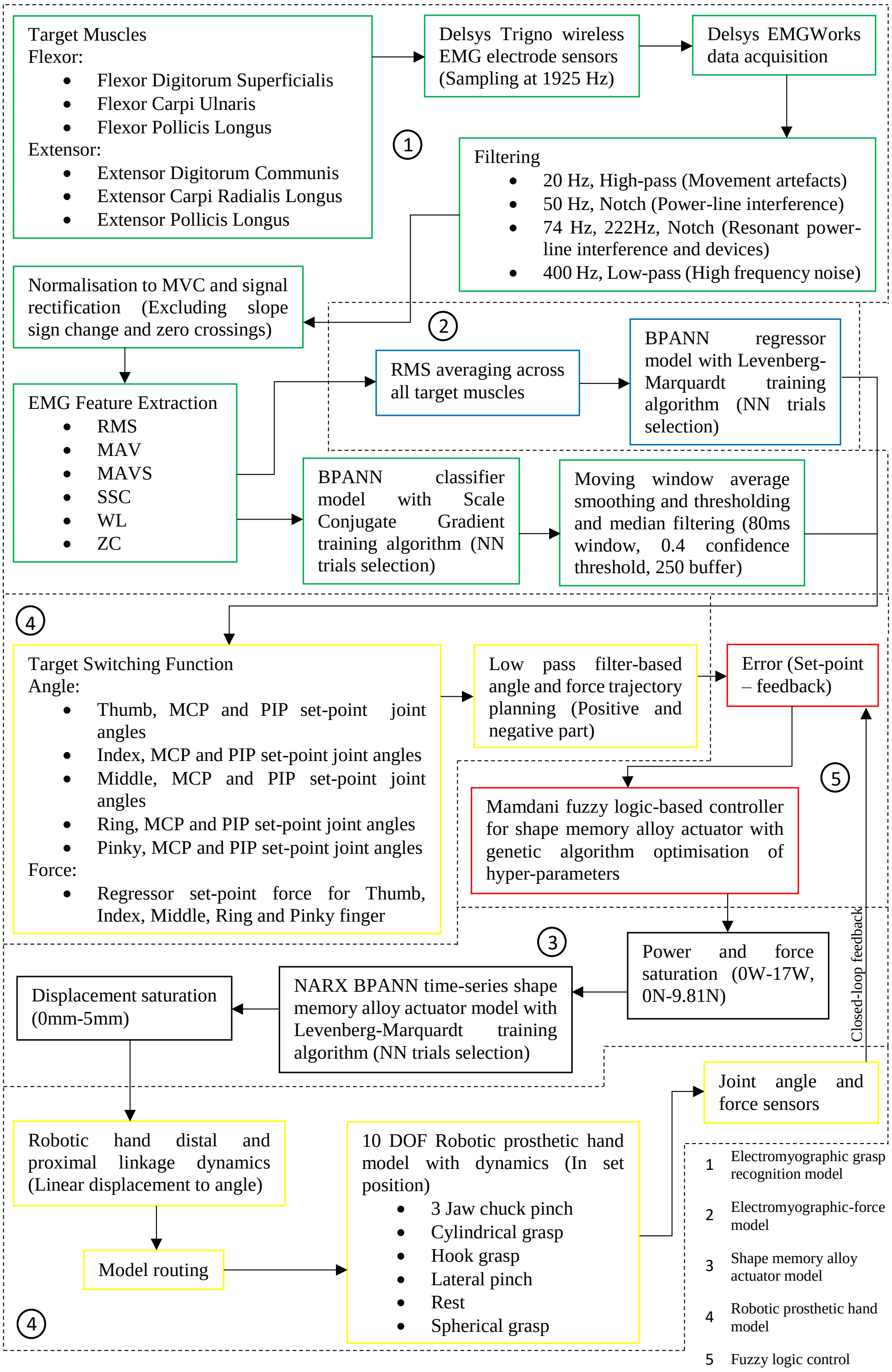


Figure 2.1: Model overview.

Chapter 3

Real-Time Grasp Classification of Multi-Channel Surface Electromyographic Signals

Towards Soft Robotics for Robotic Prosthetic Hand Actuation

In this chapter, a feasible and novel robust sEMG GR model for use in an RPH was developed. The Delsys Trigno setup and grasp apparatus were used to extract data from five non-disabled participants. Using Matlab and Python, the extracted data went through pre-processing, feature extraction using analytical and numerical tools, and pattern recognition processes. The classifiers developed were compared based on, confusion matrices, and ROC plots; and using the performance specifications results, the NN classifier proved to be the most feasible model for further use. This classifier had a 98.83% accuracy and was used to get the boundary for the GR model after smoothing, thresholding, and class interpretation of its outputs.

3.1 Theory and Literature Review

Classification of hand grasps using agonist and antagonist muscle pairs, i.e., flexor and extensor pairs, in the context of forearm muscles, is considered a multifaceted control problem. Initial strides towards myoelectric controlled RPHs made use of calibrated thresholds to perform on/off two-state control, also known as bang-bang control. This control scheme switches between on and off states when certain thresholds are met, thereafter performing a predetermined grasp movement. This has, however, progressed to more modern and disruptive soft computing artificial intelligence-based solutions.

NN proponents of such solutions include shallow, deep, convoluted, long short-term memory, Multi-Layer Perceptron (MLP) and radial basis function NNs. These classifiers may make use of numerous cost optimised algorithms for training, testing and validating. . Furthermore, the algorithms achieve minimal error by selecting appropriate weights and biases. Other non-NN-based classifiers exist and are explained in later sections.

The complexity of EMG GR is broken down into multiple steps, each of which introduces different dynamics, resulting in a large number of combinations which may yield a satisfactory solution. As shown in Figure 3.1, the progression of steps includes data acquisition, pre-processing, feature extraction, classification and post-processing, and is based on the recommendations given by Englehart *et al.* [28]. The following paragraphs introduce theory as well as references in the field of sEMG GR.

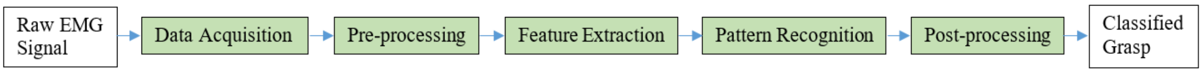


Figure 3.1: GR of sEMG signals for use in an RPH.

3.1.1 Data acquisition

EMG data acquisition, either through the use of surface or indwelling electrodes, follows a strict set of guidelines which remain the same regardless of the muscle being investigated. sEMG is generally preferred over Indwelling Electromyographic (iEMG) since it is non-invasive and does not produce any discomfort. That being said, sEMG come with several drawbacks, including cross-talk and noise. Throughout literature, an overwhelming majority of EMG data acquisition devices used for GR classification purpose are based on sEMG approach.

A notable example of this is presented by Akhmadeev *et al.*, [9], where eight sEMG electrodes, in the form of a Thalmic Myo armband, were used to classify six different hand gestures with a 99% accuracy. The sEMG device was notably unusual in that it sampled at a rate of 200Hz with 8-bit precision, which according to literature [29, 30, 31], is roughly half of the recommended sampling rate. This peculiarity, coupled with the accuracy achieved, seems to suggest that a frequency range of 20Hz to 200Hz accounts for much of the sEMG signal required for GR. On the contrary, it may be debated that the lack of samples used to establish the classifier negatively impacted the generalising capabilities of the classifier.

Further delving into data acquisition, a comparison of six different sEMG data acquisition setups, namely Otto Block, Delsys Trigno, Cometa Wave, Dormo electro- cardiogram and dual and single Thalmic Myo armbands, by Pizzolato *et al.*, [32]. The study found that the best classification accuracy was obtained from the data recorded by a 12-sensor Delsys Trigno setup. This was done using various Ninapro databases, and considered 41 hand movements, retaining a fixed classifier architecture. In conclusion, Pizzolato *et al.*, argue that other low-cost (feasible) alternatives, such as the Thalmic Myo armband, produced data that lead to acceptable performance.

Conversely adopting an iEMG-based approach, Nawab *et al.*, [33], used a database, populated with data collected from quadifilar needle sensors, to classify complex hand movements using a wavelet decomposition technique which yielded feasible performance measures. The wavelet decomposition technique, which is growing in popularity [34], assigns fundamental waves extracted from the iEMG signal to different MUs and further superimposes these to reproduce the original iEMG signal. In this case, the signals are produced by three differential channels and a cannula-derived channel.

Due to various safety concerns surrounding iEMG data acquisition, it was decided that an sEMG route be explored in this dissertation. For sEMG, recommendations are given by the Surface Electromyography for the Non-Invasive Assessment of Muscles (SENIAM) protocol. Apart from containing an extensive repository of data resources, the SENIAM protocol standardises the position and placement of electrodes by using reference motor point locations. The protocol also considers the sensory and pre-processing requirements necessary for EMG data acquisition. This dissertation focuses on forearm muscles with the intended goal of achieving prehensile movements, as shown in Figure 3.2. As identified by Billock [3], these grasps are arguably the most common grasps used, thereby substantiating their selection for further investigation.

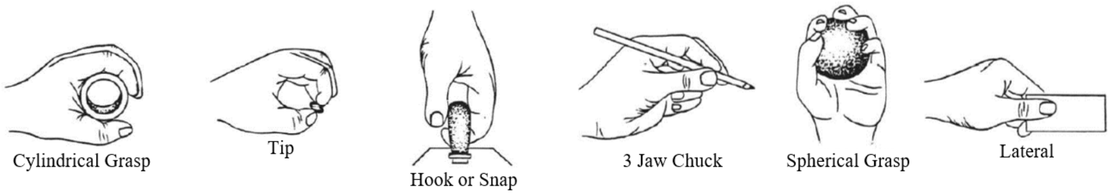


Figure 3.2: Prehensile hand movements associated with common everyday tasks [3]; reproduced with permission.

Comparably with the database introduced by Pizzolato *et al.*, Sun *et al.*, [32, 36], investigated a large number of grasps, totalling 52 grasps, excluding rest. The research made use of two Thalmic Myo armbands in parallel, each with eight data acquisition channels. The first armband was placed close to the elbow, with the corresponding first sensor placed in-line with the radio humeral joint. In addition, the second armband was placed closer to the hand with a staggered 22.5°arrangement. Using this setup, Sun *et al.*, [36] maintain that a generative flow model with a softmax classifier could classify the aforementioned grasps with a 64% accuracy. The workers also postulated that the assumption made between the factorised feature and muscle synergy led to lower than expected results.

With fewer grasps examined, Gondolla *et al.*, [8] managed to achieve a mean classification accuracy of 76% using a cascade ANN classifier for the fist grasp movement. By comparing the results obtained by Sun *et al.*, [36] and Gondolla *et al.*, [8] the evidence projects that fewer grasping actions result in a better average classifier accuracy. This, therefore, leads to the hypothesis that increasing the number of grasps investigated, increases the complexity of the classifier, thereby making it exceedingly challenging to tune and optimise. By the same token, using limited sensors to classify a large number of grasps may be likened to an underactuated system.

Recently advancements in GR has resulted in classification forecasting through anticipatory signals. A notable advantage of such an approach is the reduction in lag. Siu *et al.*, [37] implemented this operating procedure and achieved an accuracy of 76% using a combination of Gaussian mixture and continuous-emission hidden Markov models. The investigation made use of seven Delsys Bagnoli sensors, equidistantly spaced around the circumference of a stretchable band, to forecast the intended grasp, either picking up a cup or a rod. The sensors were placed at 25% ulna length, with the origin at the lateral epicondyle.

Due to the link between grasp and muscle, it is mandatory to give insight and discuss the source of the grasps. Each muscle in the body is made up of many individual muscle fibres, where fine motor control generally requires fewer muscle fibres as opposed to muscles providing strength. This dissertation considers high-density muscles in the superficial compartment of the forearm, making the task of palpation easy to accomplish. Palpation is a preliminary step in sEMG GR and is used to locate a muscle of interest.

To do this, resistance is applied in the opposite direction to the action, which results in a palpable tensing of the target muscle.

The forearm muscles identified as being pivotal in achieving the grasps shown in Figure 3.3 are listed in Table 3.1, with corresponding action, origin and insertion points. Using SENIAM recommendations, palpation and visual cues, the location of the target muscle and its belly may be found. Once the locations are established, an electrode is attached to the surface of the skin directly above. To this end, the target muscles and corresponding belly locations shown in Figure 3.3 and are reflected by label and colour in Table 3.1. Grasping may now take place; however, the correct posture and placement of the forearm must be maintained throughout the recording of the sEMG signals.

Table 3.1: Target muscles for the prehensile hand movements identified [38, 39]

Muscle	Label	Action	Origin	Insertion
Flexor digitorum superficialis	1 Brown	Flexion of PIP joints with coupled MCP and wrist movement	Medial epicondyle of the humerus and portions of the radius and ulna	Anterior base of proximal phalanges of fingers excluding thumb
Flexor carpi ulnaris	2 Purple	Flexion and adduction of the hand at the wrist joint	Medial epicondyle of the humerus, olecranon and posterior of ulna	Pisisform bone, hook of hamate and base of pinky metacarpal
Flexor pollicis longus	3 Pink	Flexion of the thumb phalanges and wrist	Middle of anterior radius surface and adjacent interosseus membrane	Base of distal thumb phalanx
Extensor digitorum	4 Red	Extension of MCP, PIP and DIP joints, and assists wrist extension	Lateral epicondyle	Extensor expansion to middle and distal phalanges of fingers excluding thumb
Extensor carpi radialis longus	5 Green	Extension and abduction of the hand at the wrist joint	Lateral supracondylar ridge	Posterior base of index metacarpal
Extensor pollicis longus	6 Yellow	Extension of IP and MCP joints of thumb	Middle third of posterior ulna and adjacent interosseous membrane	Base of distal phalanx of thumb

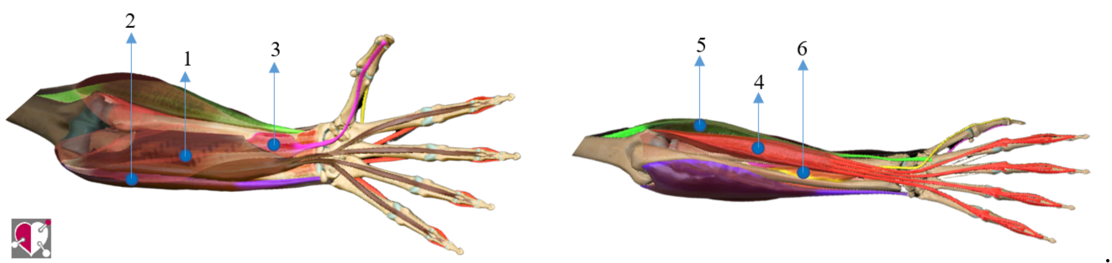


Figure 3.3: Forearm locations of target muscles, left: anterior view, right: posterior view; images courtesy of Anatomy Learning, Dr. R. Blanco Salado

Taking a look at sEMG devices, it is noted that they all follow a similar construction, constituting bipolar electrodes with a carefully selected inter-spacing coupled to an amplification phase that is in series with a filtering phase. Some notable passive and active EMG devices include Delsys, Noraxon and MyoWare, all of which exhibit unique advantages and disadvantages.

Some of the disadvantages include Power-Line (PL) interference, movement artefacts and electrocardiograph interference. On the other hand, differences in physiology lead to varying degrees of cross-talk between adjacent muscles, skin impedance, subcutaneous fat, muscle fatigue, muscle conduction velocity and muscle firing rate. Another pronounced concern apart from the sampling rate is the Signal to Noise Ratio (SNR) inherent in the sEMG device. This factor, in essence, determines the extent to which filtering needs to take place.

3.1.2 Data pre-processing

This second step in GR assesses the quality of the raw sEMG data through the lens of both time and frequency-based characteristics. Nonetheless, before filtering, it is essential to select the data that constitutes the sEMG signal, since many sEMG devices record measurements other than potential difference. A good example is inertia measurements in the x, y and z axes. Due to the sheer size of data being recorded, programs such as MATLAB and Python are more efficient than Excel for processing.

Analysing sEMG data in the time-domain is unintuitive, and sources of noise may not initially be apparent. Therefore, Fourier transforms are useful in transforming the raw data into a frequency-domain representation, where it is easier to pick out sources of noise and their corresponding resonant frequencies. It is well-established in literature that appropriate filtering yields a reduction in amplitude of artefacts from PL interference.

Filtering is an sEMG pre-processing procedure and is introduced by Atzori *et al.*, [10]. Here, a Hampel filter is used to attenuate PL interference and harmonics recorded by the Delsys Trigno sensors. Also, super sampling, through linear interpolation, was used to increase the density of the sEMG signal and offline relabelling was performed with a generalised likelihood ratio algorithm. Vital to the performance of the classifier, the labelling process ensured that visual stimuli were aligned with contraction periods. In combination with low and high-pass filters, Atzori *et al.*, [10] found that the aforementioned pre-processing implementations were sufficient to reduce the effects of white Gaussian noise as well as movement artefacts. Furthermore, De Luca *et al.*, [40] recommended a Butterworth filter with a corner frequency of 20Hz and slope of 12dB/oct. These filter specifications compromised between the attenuation of low-frequency mechanical artefacts and the preservation of the sEMG signal.

As an alternative to filtering, Anand *et al.*, [41] showed that a canonical correlation analysis suppressed heterogeneous white Gaussian noise while retaining the information density present in the original sEMG signal. Herein the noisy sEMG signal was broken into canonical components with subsequent morphological filtering, Gaussian filtering and thresholding.

Another typical inclusion in the pre-processing procedure is sEMG signal rectification. This fundamental concept together with detailed insight is highlighted by Myers *et al.*, [42], who asserts that rectification enhances the sEMG firing rate information of the

EMG signal by shifting the action potential spectrum peak to lower frequencies. Thus, the wide-spread use of rectification is justified, however, may not be explicitly necessary as many features inherently rectify the sEMG signal.

Noise may be categorised into four main types, namely ambient noise, movement artefact, inherent sEMG signal instability and inherent circuit instability [43]. Though filtering deals with the first three types, it fails to remove noise components emanating from the sEMG sensor circuit. This is because the energy associated with this type of noise overlaps the energy of the sEMG signal, thereby making it indistinguishable. To solve this, a wavelet-based denoising method is proposed by Phinyomark *et al.*, [44], which requires the selection of five processing parameters. Phinyomark *et al.*, [44] showed that the method increased classification accuracy between 2% and 50% for an sEMG database considering seven hand movements and eight muscles.
Non-linear signals, such as sEMG signals, often warrant adaptive pre-processing techniques such as empirical mode decomposition. First used by Andrade *et al.*, [45] to filter sEMG signals, empirical mode decomposition uses an iterative approach to estimate intrinsic mode functions. Furthermore, this technique is shown to yield superior results when compared to wavelet-based denoising [46]. However, the computation of intrinsic mode functions is cumbersome due to the iterative manner in which it is estimated. To mitigate long computations and the sensitivity induced by empirical mode decomposition, Zhang *et al.*, [47] suggested a noise-aided derivative called ensemble empirical mode decomposition. This technique removed the effect of mode-mixing and showed that it could adequately remove PL interference, white Gaussian noise and baseline wandering [48].
Though literature suggests various sEMG pre-processing methods which exhibit better classification accuracies than filtering, it should be stated that these techniques are far more complex in implementation and given suitable sEMG sensors become redundant.

3.1.3 Feature extraction

Feature extraction is a method in which time, frequency and time-frequency based characteristics are selected from a signal to define the underlying system or data dynamics. Classification efficiency is vastly reduced without the use of feature extraction due to noise and fluctuations present in the raw sEMG signals. These, in turn, corrupt the training process and leads to the incorrect mapping of relationships. Also, the addition of feature extraction increases the information clustered within the sEMG signal, which in turn increases class separability.

One of the most widely explored time-based feature sets in sEMG classification is the Hudgins' set, consisting of Hudgins' features, namely MAV, MAVS, SSC, WL and ZC [49]. Another commonly used time-based feature in GR is RMS, followed by standard deviation and variance statistical features [49].

Hudgins *et al.*, [48] argued that the combination of Hudgins' features resulted in superior classification, of which it is suggested that it can be improved through the incorporation of time-frequency-domain features.

Due to a large number of features available, it is infeasible to explore all combinations. Thus, dimensionality reduction approaches are often used to remove irrelevant or highly correlated features. Techniques like particle swarm optimisation and genetic algorithm [50] have all shown success in selecting and reducing the number of features required, especially for large datasets. Principal Component Analysis (PCA) has also received a lot of attention and success in increasing the number of effective parameters [51, 52] and remains the default feature reduction technique in the classification learner toolbox provided in MATLAB. More recently, a Grey Wolf Optimiser, proposed by Too *et al.*, [53], showed an improvement in classification performance, feature reduction and computation time when compared to binary particle swarm optimisation and genetic algorithm.

Closely linked to feature extraction is the use of sliding windows, which may be explained as an interval (also known as a window) of specific length iterating each sample over the data set, while computing the intended statistics for the given window length. This procedure increases the information contained within a considered segment. It is, in essence, a collective representation of the data contained within the window length, which may be likened to a low-pass filter with a cutoff frequency dictated by the window length.

Based on the work by Zhai *et al.*, [54] which focused on exploring the advantages of using Convolutional Neural Networks (CNNs) for neuro-EMG-based pattern recognition, it appears that a window and overlap length of 200ms is suitable for GR. Yielding similar performance and again using a CNN, Hartwell *et al.*, [55] validated a shorter window and overlap length of 150ms and 5ms, respectively.

It is however noted that the length of window and overlap, though impacting latency, also has an effect on the overall performance of the classifier. In this dissertation, a shorter window length was explored such that lag induced during later processing steps would result in a latency below 300ms for real-time implementation [56].

Shown in Algorithm 1 is a pseudo-code representing the concept of sliding windows. The pseudo-code starts by importing the raw sEMG data, which consists of voltages typically in the order of 106V. These voltages represent muscular activity over the recorded period. Then using a combination of user-specified window and overlap lengths as well as the total length of the data set, the number of padded zeros are calculated. These are appended to the data array such that out of bounds indexing does not occur. A feature extraction method is subsequently computed for each window. This method reduces the number of samples by several orders of magnitude, inadvertently decreasing computation time.

Algorithm 1: Pseudo-code for GR feature extraction of sEMG data

```
Data: Raw sEMG data
Result: Feature extraction
Load data;
Zero pad data depending on window and overlap length;
for Length of data do
    Perform feature extraction method;
    Append data;
end
Save results;
```

Equations 3.1.1 through 3.1.6 define mathematically the features chosen and investigated in this dissertation. The features were intentionally selected to minimise the need for feature reduction. It is advised that the dimensionality of features be reduced when considering unknown combinations of/or excessive features.

Starting with what is considered a fundamental feature, Equation 3.1.1 represents the RMS over a given window length, where N is the total number of samples in the window length and x_n is the individual samples. Plotting this equation produces a rectified sEMG signal with peaks described by muscle contractions and expansions, and valleys described by periods of inactivity.

$$RMS = \sqrt{\frac{1}{N}\sum_{n=1}^{N} x_n^2} \tag{3.1.1}$$

Equation 3.1.2 is used to calculate the MAV for a given window. By taking the absolute value, Equation 3.1.2 inherently performs full-wave rectification and preserves the negative components of the sEMG signal while reflecting the average sEMG signal.

$$MAV = \frac{1}{N}\sum_{n=1}^{N} |x_n| \tag{3.1.2}$$

Equation 3.1.3 is linked to Equation 3.1.2 such that it considers the first derivative expressed as the slope of Equation 3.1.2. This is useful in determining the rate at which the sEMG mean increases or decreases and may be an indication of the MU firing rate and muscle fibre conduction velocity. The equation makes use of the current and future MAVS values under the assumption of a constant sampling rate.

$$MAVS = MAV_{n+1} - MAV_n \tag{3.1.3}$$

Shown in Equation 3.1.4 is a feature also used outside of feature extraction, in noise attenuation and data pre-processing. sEMG noise which consists of high-frequency superimposed oscillations is identifiable and separable using higher than average SSCs, thereby substantiating its possibility as a pre-processing technique.

The method of SSC feature extraction deals with the number of negative to positive (or vice versa) gradient changes occurring within the window length, in other words, quantifying the number of turning points present in the sEMG signal. This equation results in the sum of a binary array, where zero depicts no SSC, and one proves the existence of one (that is the SSC). It should be noted that the SSC is a whole number which is calculated using non-rectified sEMG data.

The optimum threshold which is contingent on the SNR of the sEMG data is highlighted by Kamavuako *et al.*, [57], suggesting that a threshold factor of zero provides a good trade-off between system performance and generalisation. It is furthermore postulated that an increase in the threshold factor, though reducing background noise by smoothing the signal, only serves to degrade the sEMG signal characteristics. In comparison, Phinyomark *et al.*, [58] found the optimum gain-dependent threshold to be 10mV for both SSC and ZC features. At the same time, Akhmadeev *et al.*, [9] applied a threshold equal to 3% of the maximum signal level.

$$SSC = \sum_{n=2}^{N} f[(x_n - x_{n-1}) \times (x_n - x_{n+1})]$$
$$f(x) = \begin{cases} 1 & \text{if } x \geq \text{threshold} \\ 0 & \text{otherwise} \end{cases} \tag{3.1.4}$$

Equation 3.1.5 outlines an important feature describing signal complexity. The WL illustrates the cumulative jump (that is positive or negative) in the sEMG signal for consecutive time steps. Rapid shorter fluctuations typically are as a result of noise and therefore appears as lower WLs. Also, WL by definition, is a positive number and indicates the waveform amplitude, frequency and duration [49]. All variables used in the equation were previously described.

$$WL = \sum_{n=1}^{N} |x_{n+1} - x_n| \tag{3.1.5}$$

Like Equation 3.1.4, Equation3.1.6 is calculated using rectified sEMG data with the threshold deciding which ZC is counted. Given a discretely sampled sEMG signal, the difference in signs between the current and following sample constitutes a ZC (which is crossing the threshold axis).

Commonly used in speech recognition, ZC gives a coarse depiction of the spectral characteristics of a signal. Also, a higher ZC rate implies the presence of noise and reduced muscle activity. The optimum threshold was discussed in the introduction of Equation 3.1.4. In Equation 3.1.6, sgn and $\cap$ refer to the sign and intersection respectively.

$$ZC = \sum_{n=1}^{N-1} [sgn(x_n - x_{n-1}) \cap |x_n - x_{n+1}| \geq threshold]$$
$$sgn = \begin{cases} 1 & \text{if } x \geq \text{threshold} \\ 0 & \text{otherwise} \end{cases} \tag{3.1.6}$$

3.1.4 Pattern recognition

Pattern recognition in broad terms describes a process where underlying correlations that are mainly non-linear and difficult to spot are deduced from input-output pairs. This may be likened in classical control theory to a form of system identification. Therefore, adequate feature extraction is a prerequisite to good pattern recognition.

Pattern recognition loosely referred to as classification frequently uses iterative training to learn and map a combination of linear and/or non-linear functions to input-output data. In this case, the input data represents the features extracted, while the output data represents the corresponding classes. The output classes are synonymous with labels which bear no numerical significance.

Comparable with the myriad of features is the number of pattern recognition algorithms. Previously mentioned in Section 1.3, these include ANN, SVM, DA, FIS, ANFIS, k-th Nearest Neighbour (KNN) and Decision Tree (DT) classifiers. Each classifier is better suited for specific applications. In the case of sEMG pattern recognition, Linear

Discriminant Analysis (LDA), ANN and SVM classifiers have shown the most potential. Regarding ANNs, Laezza [59] investigated the performance associated with the implementation of CNNs, Recurrent Neural Network (RNN) and a combination of the two, to classify 40 different grasps using data present in the Non-Invasive Adaptive Hand Prosthetics database (Ninapro DB7). Laezza found that an RNN with long short-term memory units performed the best with an accuracy of 91.81%, arguing that this was a result of the memory-like characteristics of the network. It was concluded that deep NNs for myoelectric control have many hurdles to overcome; however, their success as a classifier cannot be discounted.

In the same manner, Yamanoi *et al.*, [60] showed that CNN might be used to reduce the error associated with prosthesis reseating. This operation typically requires retraining of the classifier model, but based on the results demonstrated; this was not a requisite for the developed classifier. Quantitatively the accuracy improved by 12% when compared to the baseline, ultimately culminating in a reseating accuracy of 84% after a reseating drop of 4%.

Concerning ANN classifiers, Gondolla *et al.*, demonstrated that a single-layer BPANN classifier is as effective in GR as its CNN counterpart. Using a unique approach consisting of two BPANNs in series, Gondolla *et al.* effectively reduced network complexity. As a by-product, the tuning of hyper-parameters was made easier. On the other hand, when extended to multiple grasps, the classifier architecture is predicted to be slow and computationally expensive for real-time applications.
Improving on the accuracy attained by Gondolla *et al.*, by a margin of 13%, Ahsan *et al.*, [61] similarly developed a BPANN classifier to distinguish between four rudimentary hand motions specifically up, down, left and right. Also, only one sEMG channel was considered, reducing the size of the feature vector and simplifying fine-tuning opportunities, including the selection of a training algorithm and the number of hidden neurons. Ahsan *et al.*, [61] emphasised the comparison between Levenberg-Marquardt (LM) and Scale Conjugate Gradient (SCG) training algorithms and found the former algorithm was superior. The major limitation to the study was the use of small sEMG datasets, which do not truly reflect the characteristics of the LM training algorithm, since more massive datasets present Jacobian related memory issues.
As previously indicated, a large number of classifiers are validated in GR applications. Summarised in Table 3.2 are a few of the classifiers with corresponding accuracy, training algorithm and reference.

Table 3.2: Comparison of classifiers used in sEMG GR [62].

Classifier	**Accuracy (%)**	**Training Algorithm / Optimisation**
BPANN	89.2	LM
	99.0	Gradient descent
	97.5	Gradient descent
	91.0	SCG
	76.0	SCG
CNN	98.3	Transfer learning
	72.1	Stochastic gradient descent

Continued on next page

Table 3.2 -- *Continued from previous page*

Classifier	Accuracy (%)	Training Algorithm / Optimisation
	90.4	Stochastic gradient descent
DA	98.8	Resillient back-propagation
	92.7	Baysian optimisation
	91.8	Maximization
SVM	92.0	Cost parameter iteration
	92,0	Limited memory Broyden-Fletcher-Goldfarb-Shanno
	93.3	Genetic algorithm
FIS	93.1	User-knowledge
	90.0	User-knowledge
	89.7	Back-propagation
ANFIS	92.0	Back-propagation and least-mean-square
	95.0	Error back-propagation
	72.0	PCA
DT	97.6	Baum-Welch
KNN	84.9	Euclidean distance
	58.0	Euclidean distance
	78.6	Euclidean distance
Extreme learning machine	99.8	Extreme learning machine algorithm
	98.5	Coarse grid-search

ANN is a term which broadly refers to different NNs. Inspired by the brain, ANNs make use of links (synapses) with weights (dendrites) to form the pathways between activation functions (neurons). The lineage of an input signal (action potential) through an ANN shown in Figure 3.4 starts with a multiplicative operation, which essentially scales the input. Thereafter at the neuron junction, the resultants are summed and passed through the activation function alternatively understood as a transfer function. Differences in the activation function dictate the type of NN developed. For instance, the perceptron developed by Rosenblatt [63] is the predecessor to most modern NNs. This consists of a step activation function and a threshold term referred to as the bias which collectively produces a binary output. Returning to the input signal lineage, a bias term is added, and the process is repeated depending on the number of the hidden layers of the NN.

Stemming from the emergence of the back-propagation algorithm; the perceptron was developed into the aforementioned multilayer perceptron (MLP). This algorithm made it possible to update weights given a multi-layer network. The back-propagation algorithm forms the foundation of NN-based classifiers upon which training algorithms are implemented. For example, when the back-propagation algorithm is combined with the gradient descent algorithm, the gradient of the cost function with respect to the output received during the feed-forward step is minimised. Consequently, the weights are updated in the direction of steepest descent, resulting in a classifier with better performance.

Training algorithms often associated with NNs include gradient descent, LM, Polak-

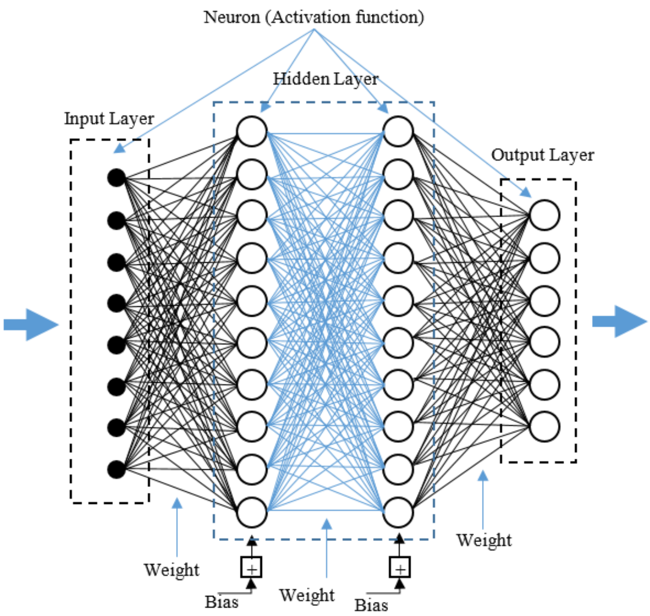


Figure 3.4: Architecture of a multi-layer NN.

Ribiere and Fletcher-Powell conjugate gradient, Quasi-Newton, conjugate gradient and SCG. These algorithms perform differently depending on the application; however, as implicitly noted earlier, LM is known to yield faster convergence and is, therefore, most commonly used. This is substantiated in a review conducted by Nazmi *et al.*, [62], in which the LM training algorithm outperformed the SCG training algorithm given 10, 20 or 30 hidden neurons (single-layer NN). However, the study did not consider the computational affects stemming from large input data, but as an alternative provided adequate training time. Algorithms such as gradient descent produce the same outcome but have an added advantage (in comparison to other training algorithms than LM) of being less likely to produce vanishing outputs due to the small step-sizes taken. Besides slow convergence, a major disadvantage of using such an algorithm for training classifiers is the difficulty brought about in locating a global error minimum.

A solution to this, developed by Møller [64] is the SCG algorithm which accordingly scales orthogonal steps in search of the global minimum. The SCG training algorithm is discussed to provide context consequent to its use in this dissertation. With the conjugate gradient algorithm as the starting point, it was thought that the scaling factor used to determine the step-size could more accurately be described by using a non-symmetric approximation. After testing this theory (adaptive conjugate approach), Møller found that it failed in almost every instance due to the absence of a positive definite global error function Hessian matrix. Møller then proposed that the model-trust region approach from the LM algorithm be combined with the conjugate gradient algorithm. This was done by adding the multiplication of a scalar and the current conjugate point to the approximation. By iteratively adjusting the scalar according to whether the Hessian matrix is a positive definite or not; the indefiniteness of the global error function is regulated. In other words, the scale factor is appropriately shifted such that the updated weights will culminate in a solution that converges.

Equation 3.1.7 [65] defines the proposed scale factor, where the first term is the nonsymmetric approximation of $E''(\tilde{w}_k)\tilde{p}_k$. Additionally, λ_k is a scalar, $\tilde{p}_k$ is the conjugate system, E is the quadratic global error function, $\tilde{w}_k$ is the weights vector and $\sigma_k = \sigma/|\tilde{p}_k|$, where σ is a scalar element in the range $0 < \sigma \leq 10^{-4}$.

$$\tilde{s}_k = \frac{E'(\tilde{w}_k + \sigma_k \tilde{p}_k) - E'(\tilde{w}_k)}{\sigma_k} + \lambda_k \tilde{p}_k \tag{3.1.7}$$

The manner in which the scalar augments the estimation of the global error function is artificial. Møller also proposed a measure denoted as Δ_k, which gauges the approximation strength and is shown in Equation 3.1.8.

$$\Delta_k = \frac{E(\tilde{w}_k) - E(\tilde{w}_k + \alpha_k \tilde{p}_k)}{E(\tilde{w}_k) - E_{qw}(\alpha_k \tilde{p}_k)} \tag{3.1.8}$$

Where α_k is the step-size and E_{qw} is the quadratic global error approximation in the neighbourhood of the weights vector.

Finally, according to the rules shown in Equation 3.1.9, the scalar together with the weights vector, shown in Equation 3.1.10, are updated. δ_k is a recursive product and equates to $\tilde{p}_k^T E''(\tilde{w})\tilde{p}_k$.

Besides the training algorithm, another critical consideration in the development of a NN is the activation function. Activation functions are mathematical constructs which serve to map an input to a corresponding output, typically in the range $-1 \leq output \leq 1$. Apart from the normalising properties, an activation function also needs to be computationally efficient and differentiable due to the scale of the network and requirements for the back-propagation algorithm, respectively.

$$\begin{aligned} &\text{if } \Delta_k > 0.75, \quad \text{then } \lambda_k = \frac{1}{4}\lambda_k \\ &\text{if } \Delta_k < 0.25, \quad \text{then } \lambda_k = \lambda_k + \frac{\delta_k(1-\Delta_k)}{|\tilde{p}_k|^2} \end{aligned} \tag{3.1.9}$$

$$\tilde{w}_{k+1} = \tilde{w}_k + \alpha_k \tilde{p}_k \tag{3.1.10}$$

There are two types of activation functions, namely linear and non-linear. Linear activation functions, which are more advantageous to use than step activation functions, present two major disadvantages. Firstly, back-propagation offers no meaning since the derivative of the linear function is a constant, which consequently hinders the comparison and selection of weights. Secondly, the network is limited to a single-layer because a combination of linear functions remains a linear function. Conversely, a combination of non-linear functions results in a function with a higher degree of non-linearity. Furthermore, non-linear activation functions are differentiable and are known to more accurately describe real-world problems.

Shown in Figure 3.5 are some common activation functions (also referred to as kernels) used in ANNs. Analysis of the log-sigmoid activation function reveals characteristics including a smooth gradient, bounded positive output and vanishing gradient. The tan-sigmoid activation function shares the same characteristics, however, is not positive bounded, but instead, zero-centred. The Rectified Linear Unit (ReLU) activation function is the most computationally efficient of the kernels presented in Figure 3.5 and although seeming linear, possesses a derivative function. The selection of initial weights must consider the fact that the derivative of the ReLU activation function breaks down approaching zero and prevents learning. The softmax activation function found in the output layer of the BPANN classifier developed in this dissertation is discussed in Section 3.4.4.2.

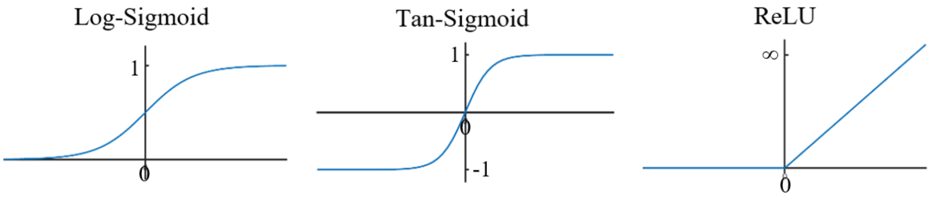


Figure 3.5: Standard NN activation functions

Despite the rules of thumb concerning the optimum selection of the number of neurons and layers in an NN-based classifier that exist, there remains no definitive answer to this question. Thus, a trial and error approach is often utilised and followed up with an optimisation technique for hyper-parameter selection.

Sharing sentiments of a trial and error approach, NN trials suggested by Heath [65] consists of the sequential adjustment for the number of neurons and layers. The performance of the NN is assessed for each iteration given fixed initial conditions and sufficient training epochs. By following this approach, not only does it indicate the neuron-layer configuration with the most potential, but it also gives insight into the degree of non-linearity linking inputs and outputs.

Optimisation techniques such as particle swarm optimisation and genetic algorithm have shown improved success when compared to the random selection of the number of neurons and layers [50]. However, as the network size and ensuing number of parameters increase, the effectiveness of the optimisation technique drops leading to slower convergence. This is attributable to the stochastic behaviour associated with the selection of the number of neurons and layers and is less prevalent in the selection of other hyper-parameters. Hence, these techniques are far better suited for the optimisation of parameters, including the learning rate and regularisation, to name a few.

By virtue of simplicity and the number of parameters, a NN trial approach is adopted in this dissertation for the selection of the number of neurons and layers. The pseudo-code governing the outline of this approach is shown in Algorithm 2. At first, the hyper-parameters are set and kept constant for all trials. Then the number of neurons and layers are changed within the bounds defined, and the NN is trained. Once complete, the performance is recorded for comparison and selection purposes.

Guided by the need to compare the performance of GR classifiers fairly, it is essential to establish the mathematical framework related to the performance specifications pointed out in Section 2.1.1.

The most basic form of comparison between classifiers may be achieved using the accuracy, is the ratio between correct classifications and the total number of classifications. Although showing deceptive properties depending on the distribution of data and application, accuracy remains arguably the most widely accepted performance measure for classifiers. Equation 3.1.11 defines the accuracy of a classifier using constructs extracted from a visual performance metric known as a confusion matrix. Herein, the terms TP, FN, TN and FP are the true positive, false negative, true negative and false

Algorithm 2: Psuedo-code for NN trials; establishing the optimum network architecture

```
Data: Feature and class matrices
Result: NN performance
Load data;
Initialise NN;
Initialise weights and biases;
Set training algorithm;
Set activation function kernel;
Set stop criteria;
for Number of iterations do
    Update number of layers;
    Update number of neurons;
    Train NN;
    Compute and append performance;
end
Save results;
```

positive, respectively. Accuracy is defined within the range $0 \leq accuracy \leq 100$, where zero and 100 refer to the worst and best possible classifiers.

$$Accuracy = (\frac{TP + TN}{TP + TN + FP + FN}) \times 100 \tag{3.1.11}$$

Before regarding other classifier performance measures, two important concepts are introduced mathematically in Equations 3.1.12 and 3.1.13. Equation 3.1.12 defines precision, which is described as the proportion of data predicted as being relevant to the data that is considered relevant.

Equation 3.1.13 specifies recall, which is the ability of the classifier to find relevant classifications. The equations above are inversely proportional and frequently demands a trade-off between the two. Additionally, precision and recall are $\in \{0; 100\}$.

$$Precision = (\frac{TP}{TP + FN}) \times 100 \tag{3.1.12}$$

$$Recall = (\frac{TN}{TN + FP}) \times 100 \tag{3.1.13}$$

Succeeding the definition of precision and recall it is possible to express the harmonic mean of a combination for the two, known as the F1 score with Equation 3.1.14. For a balance between precision and recall, the F1 score excels as a performance measure in cases where an uneven class distribution is present in the data.

Furthermore, when extended to a multi-class problem, the F1 score is calculated on a per-class basis, and a macro weighted or micro average is computed for the overall F1 score for the classifier. F1 score is bounded by zero and one, and respectively refers to worst and best trade-off cases between precision and recall.

$$F1\ score = 2 \times (\frac{Precision \times Recall}{Precision + Recall}) \tag{3.1.14}$$

The last performance measure described in Equation 3.1.15 is MCC which shares similar attributes with Pearson's coefficient. The MCC, unlike the F1 score, is insensitive to class imbalance. In other words, even though classes may have different sizes, the MCC prescribes equal significance to each class. The MCC is a $\in \{-1; 1\}$, where negative one, zero and positive one respectively describe an inverse, random and perfect classifiers.

It should be noted that Cohen's Kappa is also considered in this dissertation; however, the mathematical description is not presented.

$$MCC = \frac{(TP)(TN) - (FP)(FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{3.1.15}$$

3.1.5 Data post-processing

Post-processing of sEMG classifications is a step in which misclassifications or the effect thereof is minimised. In real-time applications where different users are concerned, the classifier is guaranteed to produce a varying degree of performance. Put differently; it is not possible to achieve a classifier that will perform the same given different amputees with different levels of impairment and physiology. Moreover, this is extended to numerous factors that include sensor placement, surrounding noise, static and dynamic motion disturbances, muscle fatigue, skin impedance and psychometric condition. Now that the concept of varying misclassifications is set, the role of post-processing becomes clear.

A notable attempt in the real-time post-processing of GR classifications is presented in work by Simon *et al.*, [66], where a decision-based velocity ramp is used to attenuate the speed of the grasping action. This process effectively smoothens the transition between motions by acting like a low-pass filter. Nevertheless, despite the realistic motion that is instilled, the process also introduces "sluggish" behaviour and may be susceptible to sensor drift.

Another approach often seen is a majority vote post-processing method. To this, Englehart and Hudgins identified the potential of the method in reducing the number of misclassifications. As the name suggests, the majority vote method considers a sliding temporal window in which classification "vote" are cast. Subsequently, the votes are added, and the output is assigned to the classification with the most votes. The majority vote method is similarly reflected as a median filter.

A novel ANN-based solution was proposed by Amsuss *et al.*, [67], in which the maximum likelihood and global mean muscle activity were used as features. This post-processing technique leads to an improvement in classification accuracy ranging between 4.8% and 31.6% depending on the included nonstationarities. It should be noted that this method is complex, but when implemented, produces less lag than both the velocity ramps and majority vote discussed prior.

Based off of a simpler, more intuitive idea, Zhang *et al.*, [47], showed that a carefully selected threshold might be used to detect the onset of motion. After which, the classification is locked in place until another motion onset is predicted. This technique was shown to reduce misclassifications by 50%. Also, the post-processing technique proposed in this dissertation and discussed in Section 3.4.5 correlates with some of the aspects of post-processing presented by Zhang *et al.*, [47]

To this end, the thresholding algorithm applied in this dissertation is shown in Algorithm 3. The algorithm starts with the inclusion of a confidence input from the classifier. Depending on the confidence level and the threshold chosen; the classification is assigned to a class which describes the final output. This method of post-processing replicates the effects of onset detection, particularly if a low threshold is set. However, there remains a trade-off between onset detection (which reduces lag) and the number of misclassifications.

Algorithm 3: Pseudo-code for classification post-processing using a threshold

```
Data: Classification confidence
Result: Classified grasp
Input class confidence level;
if Input class confidence is greater than threshold & only one class then
    Output that class;
else
    Output unrecognised class;
end
```

3.2 Experimental Facilities

The apparatus introduced in this section may be deconstructed into two constituents, namely hardware and software. Hardware formed the basis of the interface between human and computer, and comprised of the sEMG recording device, grasping apparatus and processing system. MATLAB and Python programmes were used as software tools for data processing purposes. These included the applicable toolboxes, programs and written code. Over and above, though, not forming part of the apparatus listed in this dissertation, a low-cost sEMG sensor prototype was developed presupposing the non-procurement of an sEMG recording device. The use thereof was not required; however, it may be extended and used for future studies to improve the feasibility of an RPH.

3.2.1 Hardware

3.2.1.1 Electromyographic device

Delsys Trigno setup – Six wireless sensors with 16bit resolution, baseline noise of 750nV, Common Mode Rejection Ratio (CMRR) <80dB, mass of 14g and onboard Butterworth bandpass filter (40/80 dB/dec). The articulation of the apparatus required starts with the single most important piece of equipment, the sEMG recording device. This device is responsible for capturing muscular activity, which is represented as a potential difference. A Delsys Trigno system was made available by the Physiology Department and subsequently used in this study to record the sEMG signals.

An important factor to consider when acquiring data, in any application, is the sampling rate at which the data is recorded. This point is illustrated by considering a sinusoidal signal sampled at the frequency of oscillation, resulting in a flat-line output where the sinusoidal nature of the signal only becomes apparent when sampling four times the frequency of oscillation.

For sEMG GR, it is well documented that the frequency band containing the information of interest lies between 20Hz and 400Hz [68]. Therefore, with an upper-bound of 400Hz and cognisant of the Nyquist criterion, a sampling rate in excess of 800Hz is theorised. This is partly invalidated by Chen *et al.*, [69] in a comprehensive study detailing the effects of sampling rate on the accuracy of an sEMG classifier. The workers analysed several classifiers, features and sampling rates and concluded that the accuracy only noticeably deteriorated below 200Hz. Also, higher sampling rates were observed to yield higher accuracies, but only marginally so, decreasing by as little as 0.83% between 1000Hz and 400Hz. The opinions asserted by Chen *et al.*, [69] were similarly mirrored by Li *et al.*, [70], finding very little consequence in the removal of frequencies above and below, 600HZ and 60Hz, respectively. With the findings above in mind and employing the Delsys Trigno system, sampling wirelessly at a rate of 1925Hz and allowing for a natural range of motion during data acquisition is justified in suitability for GR applications.

The electrodes are comprised of dual bipolar solid Ag/AgCl extrusions with a grounded reference. This choice of Ag/AgCl electrodes minimises skin to electrode resistance, reducing noise and improving resolution capabilities. The composition of the Delsys sEMG sensors follows an outline similar to that touched on in Section 3.1.1 and are housed in a server case for charging and storing purposes. As indicated in the user-manual, the sensors should be placed parallel to the target muscle such that contractions or expansions are representative of individual muscles and include minimal crosstalk.

The Delsys Trigno system is depicted in Figure 3.6, detailing the hardware (server case and 16 wireless sensors) and software (EMGworks). Alternating green and red blinking of an LED, located on the sensor, indicates that it is on. Based on the work presented in previous studies, the Delsys Trigno system output is presumed credible and validated in both accuracy and GR purpose [71]. Also discussed in Section 3.1.1 is the introduction of low-pass and high-pass filters to attenuate the noise external and implicit in the sEMG sensors. Securing the sensors to the skin of the participant is accomplished with the use of double-sided tape, fixed to the underside of the sensor, and sporting tape (known as Elastoplast) placed over the sensor and adjacent skin.

Shown in Figure 3.7 are the anterior and posterior dominant forearm views of the participants involved after the attachment of the sEMG sensors. Five non-disabled participants, all male within the age range of 23-25 years, volunteered for this study and were all right-arm dominant. The number of participants and trials (not to be confused with NN trials) was arrived at after taking into consideration generalisation requirements and the review conducted by Phinyomark and Scheme [73] on big data sEMG classification. In the review, the number of participants ranged between five and 40, while the trials ranged between 360 and 17280. Predicated on this, the trials, referred to as iterations in Section 3.3.1, were appropriately selected and the lack of data avoided. By considering only non-disabled participants, the scope and palpation requirements were reduced. Other apparatus required for the preparation of the skin before adhering the sEMG sensors, included a razor, shaving foam, body scrub and alcohol wipes. By preparing the skin, any oil, hair, residue or dead skin is removed to increase the consistency of good quality data.

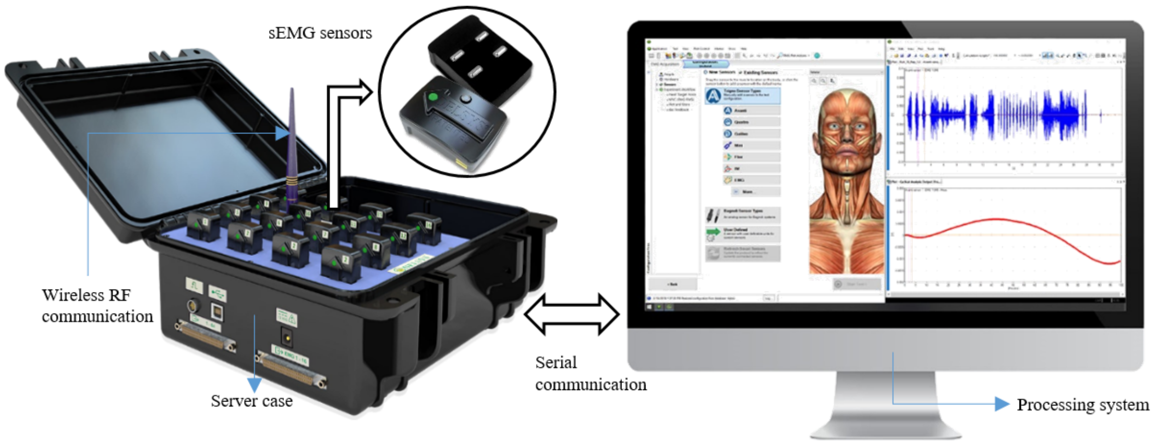


Figure 3.6: Delsys Trigno and EMGworks data acquisition setup; images courtesy of Delsys [72].

Figure 3.7: Forearm placement of sEMG sensors directly above target muscles.

An advisor from the Physiology Department was present during the data acquisition process and primarily assisted in the location of the target muscles. Permission was obtained from the participants to include the images shown in Figure 3.7, which shows the sensor placement and setup. Note that the numbers with reference to the muscle detailed in Table 3.1.

3.2.1.2 Grasp apparatus

Figure 3.8a details the grasping apparatus designed and manufactured solely to standardise the prehensile actions investigated, refer to Figure 3.2. The standardisation of Grasp Objects (GOs) is due to the fact that variations in the dimensions of GOs used during the prehensile hand movements, results in different joint angles, ultimately influencing the level of isometric muscular contraction experienced. With this, the consistency and repeatability of data acquisition, between participants, is ensured.

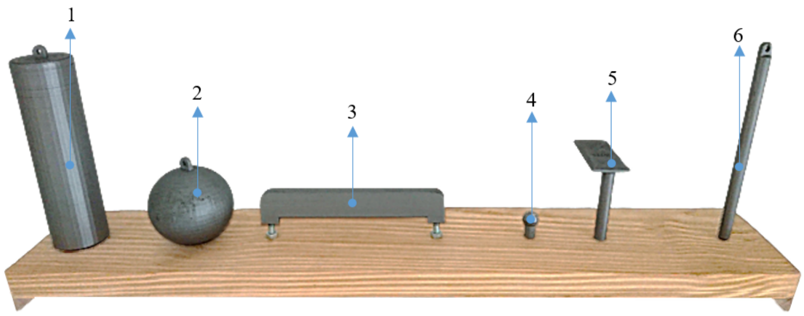
Following the invent of additive manufacturing, in the form of 3D printing, prototyping has become a rapid and straightforward process. It presents an optimal path for the manufacturing of the GOs outlined in this dissertation. The GOs were designed bearing in mind the dimensions of everyday items as well as objects used in previous studies related to GR.

The GOs were designed in Inventor and thereafter discretised into layers, i.e., sliced, using Cura, with a layer height of 0.2mm, infill of 80% and build plate temperature of

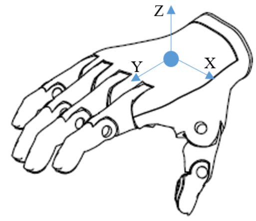
50°C in order to reduce warping. Utilising polylactic acid (PLA) filament with default Creality CR-10 (3D printer) settings, such as feed rate and fan speed, the GOs were printed in roughly 30hrs. The design of the GOs accommodated both left and right-hand dominant participants in either a mounted or suspended position.

Shown sequentially in Figure 3.8a are the manufactured GOs of the prehensile motions, namely cylindrical grasp, spherical grasp, hook grasp, tip pinch, lateral pinch and the 3 jaw chuck pinch.

The set position of the participant's hand during rest, prior and following grasping is highlighted in Figure 3.8b, and its role clarified in Section 3.4.2. The characteristic dimensions of the GOs are given in Table 3.3. By mounting the GOs to a base, the portability was improved, setup time reduced and the support of the participant's elbows made possible, in turn reducing the risk of fatigue-related effects.



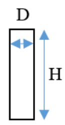
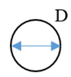
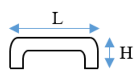
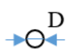
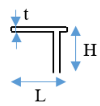
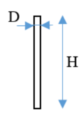
(a) Grasp apparatus



(b) Set position

Figure 3.8: Grasp apparatus and set position of hand during sEMG data acquisition

Table 3.3: Characteristic dimensions of GOs.

Types of Prehension	Dimension (mm)	Image
(1) Cylindrical Grasp	D=42, H=160	
(2) Spherical Grasp	D=60	
(3) Hook Grasp	L=160, H=30	
(4) Tip Pinch	D=15	
(5) Lateral Pinch	L=69, H=66, t=2.5	
(6) 3 Jaw Chuck Pinch	D=10, H=160	

3.2.1.3 Processing system

For machine learning applications such as pattern recognition, it is necessary to obtain a computer superior to most conventionally available computers. Knowing this, the processing system used in the development of the GR model as well as the acquisition and processing of the sEMG data is an HP notebook with 64-bit Windows operating system, 8th generation core i7 CPU at 4GHz and 8GB RAM.

The processing system made use of an integrated intel graphics card, which did not possess the same capabilities as a CUDA-enabled GPU and therefore had no effect on the reduction of the computation time.

3.2.2 Software

3.2.2.1 Electromyographic operating system

EMGworks is a proprietary software package offered by Delsys to work in conjunction with the Delsys Trigno system. When launched and configured, the graphical user interface (GUI) displays and records, in real-time, the data received serially from the analogue to digital converter (ADC) on-board each sEMG sensor. Before being able to display and record sEMG data, some preliminary steps need to be completed.

Firstly, after creating a new project, the software prompts the user to input participant information typically needed for body mass index (BMI) evaluations. After inputting the participant's information, a list of connected Delsys Trigno sensors, as depicted in Figure 3.9, will be made available. The sensors may be renamed and assigned to a specific muscle by dragging and dropping the individual sensor on the corresponding muscle provided in the anatomical model.

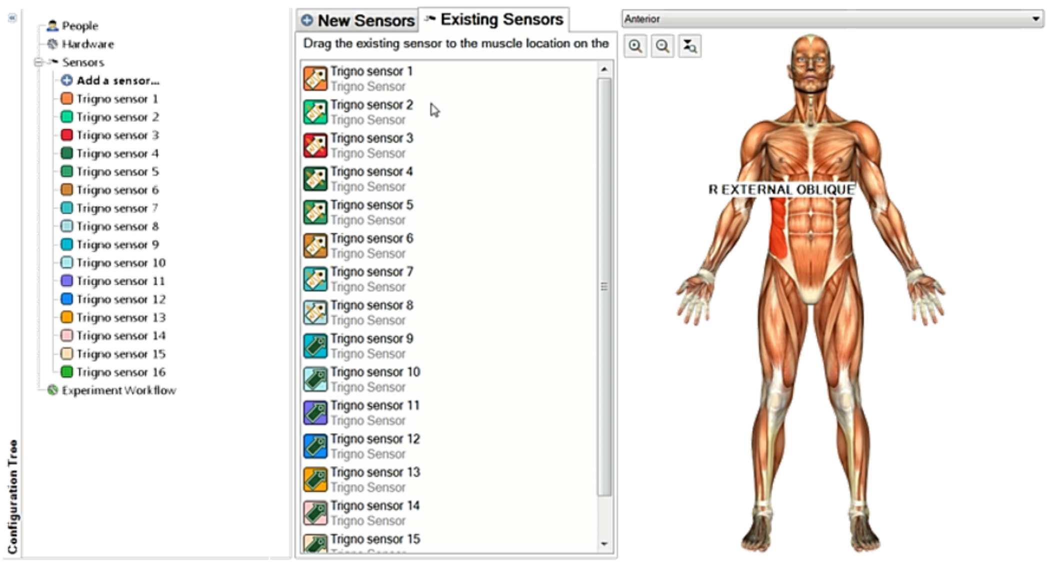


Figure 3.9: Selection and configuration of sEMG sensors in EMGworks.

Additionally, unused sensors may be discarded from the registry of sensors, while the type of data, be that inertial or sEMG potential difference, is selected as an output.
In EMGworks a variety of tasks are made available to the user and particularly include the signal preview, the plot & store and the MVC. Of these, it is advised that a signal preview be performed to ensure the sensors are in working order before commencing

with data acquisition.

The plot & store task is responsible for the real-time display and recording of sEMG data. The record duration, the number of cycles and visual prompts indicating the start and stop of each intended motion defines it. An example of the plot & store task (which comprises the bulk of data acquisition time) is illustrated in Figure 3.10. As per the output of EMGworks, the data is saved as an HPT file and later batch converted to an XLS file. This output data file contains numerous column vectors, where the first column represents time values, followed by the inertial and potential difference. Likewise, each row corresponds to a sample taken.

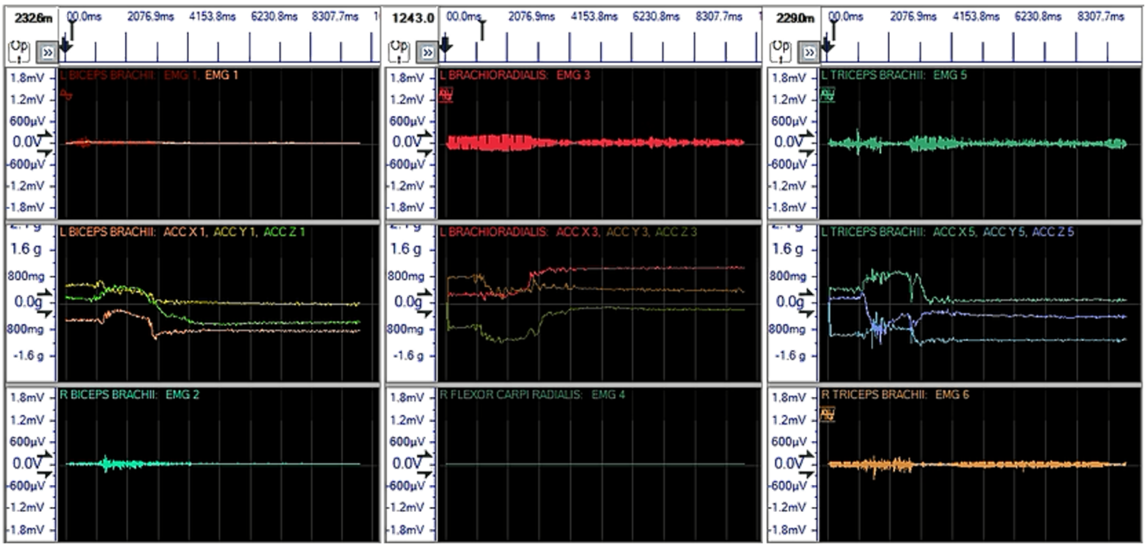


Figure 3.10: EMGworks plot and store task.

3.2.2.2 Programming packages

MATLAB and Python are sophisticated coding languages which, along with most of their add-ons, are open source and easy to manage. The user-friendly process of coding incorporates a myriad of libraries and applications, most of which are geared towards data science applications. Two such applications which have gained much attention in machine learning is Keras and TensorFlow. TensorFlow is similar to toolboxes provided by MATLAB and serves as an alternative machine learning application.

In this chapter, however, Python is not considered for machine learning applications but rather for data manipulation using the Numpy and Pandas libraries. In later chapters the combination of Python and pyQT results in a GUI with principles of which may be extended to the developed sEMG sensor (refer to Appendix 10).

MATLAB and Simulink provide resources particularly for pattern recognition and classification purposes, which permit the rapid development and optimisation of classification models. For this dissertation, the classification learner and pattern recognition NN toolboxes provided by MATLAB will be used.

Shown in Figure 3.11 is the classification learner application, which requires an input-output workspace matrix of data where the input is numerical values and the output labels. Preprogrammed classification algorithms are trained using provided data, resulting in a model with unique performance characteristics, including confusion matrices and ROCs. These characteristics are further used in conjunction with the accuracy of the trained model to select the most feasible model.

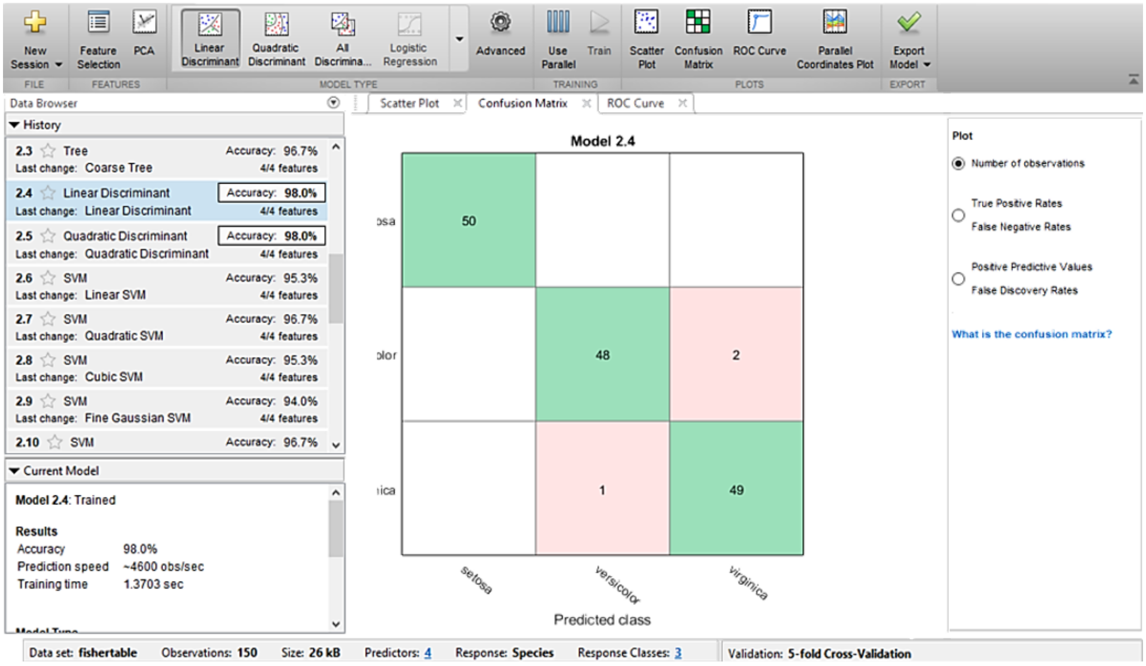


Figure 3.11: MATLAB classification learner application.

Important in big data applications is the holdout validation method, which selects a sample representation of the entire data set and is done in order to evaluate the predictive performance of the fitted model. Other options given in the application include cross-validation and no validation.

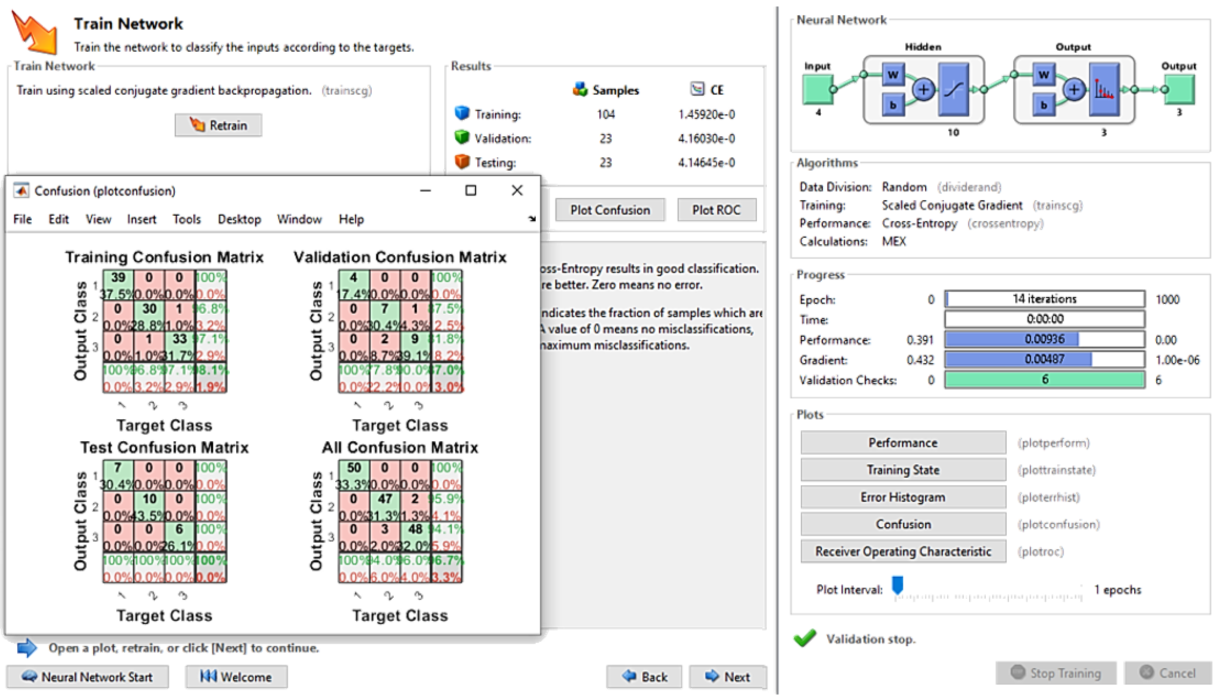


Figure 3.12: MATLAB NN pattern recognition application.

Given a NN dedicated GR solution, the NN pattern recognition toolbox, shown in Figure 3.12, follows a similar input-output data structure to the classification learner, however, differs in the outputs being binary column vectors.

Various NN hyper-parameters affecting the performance of the developed classifier may be adjusted and are only made available through written code. Once the model is trained, options to deploy it in the form of an embedded code or Simulink block are given. In this dissertation, the emphasis is placed on the Simulink block deployment of the selected model for simulation purposes.

3.3 Methodology

In this section, a multi-stage methodology geared towards the classification of grasps from sEMG data is given. The end product of which is a feasible BPANN classifier model that can distinguish between six prehensile motions excluding rest. Each stage is summarized with a flow-diagram before being elaborated on in a stepwise manner.

3.3.1 Data acquisition

The sEMG data acquired from the participants over one week formed the basis for much of the development of the proposed GR classifiers. Initially, mock data was generated using an example dataset available in MATLAB to test the validity of the ensuing methodology and formed part of the first pass attempt.

Numerous intrinsic and extrinsic risk factors were considered throughout the data acquisition process. The range in physical properties between participants posed the largest risk to data integrity and was prevented using MVC normalisation as a standardising procedure. Figure 3.13 shows a summary of the data acquisition methodology. As a prerequisite for GR, it is initially required to obtain both ethical and registrar clearance (see Appendix 10). The ethical and registrar clearance must cover the scope of the work being investigated, and no data collection may commence before receiving them. The process of GR data acquisition starts with the gathering of the relevant apparatus required, after which suitable participants for the study are found. The Delsys Trigno system used in this study is located in the Movement Lab, at the University of the Witwatersrand Education Campus. Prior to the participants' arrival, the system is powered on, and serial communication is established with the processing system using a USB adapter. The data acquisition process also requires grasp apparatus, EMGworks, consumables and Python. The participants in this study are informed, before arriving on the day of testing, to have their dominant forearm cleanly shaven. This minimises injury-related risks and reduces the time needed for setup.

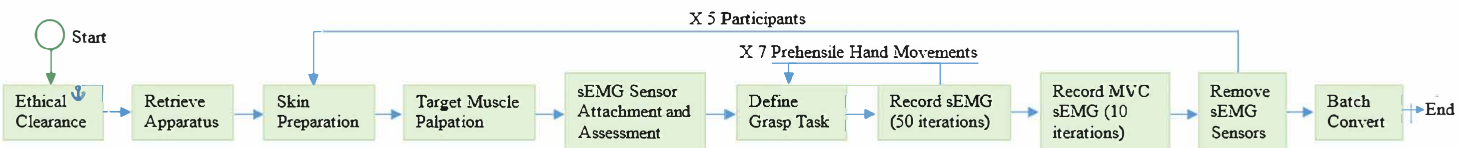


Figure 3.13: Summary of GR the data acquisition methodology.

Below are the detailed steps of the data acquisition process shown in Figure 3.13.

Step 1 With the assistance of a physiology supervisor, the participants' target forearm muscles were correctly located by merging palpable resistive motions with the forearm measurements.

Step 2 Once the locations of the target muscles were determined, the participant's forearm was swabbed with a scrub and further cleaned using an alcohol wipe. This was to ensure that the skin impedance was close to the recommended 60KΩ. A multi-meter was then employed to verify this.

Step 3 Next, the sEMG wireless sensors were attached to the participant's forearm directly above the identified target muscles by using two strips of double-sided tape sandwiched between the bottom ends of the sensor and the skin. For accurate readings, the arrow on the sensor was aligned parallel to the muscle being probed. Then the sensor was pressed firmly into place and reinforced with Elastoplast tape around its perimeter to secure it onto the skin.

Step 4 After providing sEMGworks with the participant's information, the integrity of each sensor was verified by first ensuring it appears in the list of available sensors. This was followed by the use of a signal preview task, indicating the state of the inertial and potential difference measurements graphically.

Step 5 Once the sEMG sensor integrity was validated, the grasping apparatus were introduced, and the principal investigator demonstrated six prehensile hand motions, excluding rest. This was done to acquaint the participants' with the names, actions, visual cues and the recording procedure undertaken for each grasp.

Step 6 Seven plot and store tasks were then set up in sEMGworks, where each task was scheduled to run for 50 x 10s iterations with 5s breaks in between each iteration. A countdown as well as start and stop prompts were used to provide the participants with visual cues.

Step 7 Linked to Step 6, each 10s iteration was structured such that for the first 2s the participant's hand was in a set position (see Figure 3.8b) followed by 6s of the prompted grasp, at roughly 30% MVC, and concluded in a set position for the remaining 2s. Care was taken to minimise fatigue and inconsistencies by advising the participant on posture, arm placement and the strength required for their grasp.

Step 8 After recording and saving 350 iterations, the participant-specific MVC data was acquired through requesting the participant to press firmly against a table with an open hand. This resistive motion was done for 10 x 10s iterations with longer breaks in between to avoid fatigue-related affects

Step 9 After concluding the data collection process, the sEMG sensors were detached from the participant and cleaned. A questionnaire which assessed the level of comfort the participant had during the process and probed on any information regarding prior hand impairments the participant might have had. This was useful to assist in removing erroneous data. Based off of the feedback from the questionnaire, no discomfort or previous impairments were observed or reported.

Step 10 Step 1 to Step 9 were repeated for a total of five participants. The resultant trials were batch converted to XLS files using the Delsys file utility and stored for further analysis.

3.3.2 Data pre-processing

The pre-processing of sEMG data for GR aims to expose the defining signal dynamics by lowering the amplitude of unwanted interferences. This stage of GR is governed by the implementation of software and considers, for the most part, the manipulation of frequency-based signal characteristics.

The sEMG data acquired, in its raw form, is composed of fluctuations in both the positive and negative domain. This is unfavourable in feature extraction and therefore requires rectification as a pre-processing step. Figure 3.14 shows a summary of the data pre-processing methodology. The code developed as part of the pre-processing methodology is given in digital appendices.

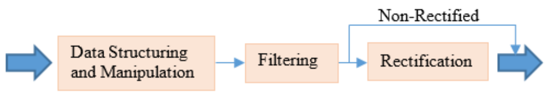


Figure 3.14: Summary of the data pre-processing methodology.

The steps describing the process of data pre-processing shown in Figure 3.14 are as follows:

Step 1 The raw XLS data was read in sequentially from the file directory it was saved in using the Pandas Python library. Inertial data was recorded, but not taken advantage of due to the absence of variance in the signal stemming from the stationary posture maintained. Otherwise stated, the trials were static and the inertial data regarded as meaningless. Hence, columns on the inertial data were removed from the dataset, leaving only the raw sEMG data.

Step 2 The newly structured raw data was imported sequentially into MATLAB, and a filtering process was applied. Using a bandpass filter and notch filters with notch strengths ranging between 35 and 105, PL interference with harmonics, movement artefacts and high-frequency noise were removed.

Step 3 Once filtered, full-wave rectification set the tone for positive definite inner workings of the classifiers developed. A non-rectified copy of the data was kept for SSC and ZC feature extraction purposes. This was done to generate negative index sEMG signals required by the feature.

3.3.3 Feature extraction

Feature extraction is performed solely through the use of software and mathematical tools. Applying the features described in Section 3.1.3 to an sEMG signal leads to an apparent and separable trend. The method of feature extraction is coupled with sliding windows, and Min-Max normalised to the MVC to improve classifier robustness.

In odd cases where the maximum feature values from the MVC data were superseded by the maximum feature values from the sEMG data, the maximum feature value from the sEMG data was used as the normalising divisor. Figure 3.15 shows a summary of the feature extraction methodology. The code developed as part of the feature extraction methodology is given in digital appendices.

Figure 3.15: Summary of the feature extraction methodology.

Step 1 Feature extraction was performed on the rectified and non-rectified sEMG data within the MATLAB environment, and beyond using the conceptsp-resented in Algorithm 1 and Equations 3.1.1 to 3.1.6. A window length of 80ms and an overlap of 20ms was set. Feature extraction results in far fewer data points in comparison to the start of the process. The feature data was categorically stored as XLS files.

Step 2 The resulting feature data was normalised to participant-specific MVC data in MATLAB. This process started with the extraction of the maximum value in the MVC data, for each participant and feature separately. Then using the absolute of this value as the divisor, the corresponding extracted feature data was normalised. This was an important step in ensuring the developed classifier adequately generalised with new data.

Step 3 The normalised feature data for the six target muscles resulted in a total of 36 features (that is six features repeated six times, one for each target muscle). For each grasp repetition and participant, the feature data was appended consecutively into a single matrix totalling 294000 rows and 36 columns.

Step 4 For class labelling purposes 5s of data was discarded from each trial using according time indices, which resulted in 147000 samples. This accounted for the initial and final 2s of each trial with a 1s buffer (in case of a late start or early termination), by utilising a Numpy array and Pandas data frame for batch processing. The feature dataset was saved as an XLSX file with the corresponding output matrix/column being assigned labels indicating the associated grasp.

Step 5 The classification models explored in the MATLAB workspace emphasises the data types as well as the data structure required. For the classification learner, a separate column for each class is required as an output, while the input consists of the feature matrix described in Step 4. The NN pattern recognition input is the same as the input of classification learner, differing in output with a singular column containing the relevant class labels.

3.3.4 Pattern recognition

Two performance recognition paths are covered in this dissertation, resulting in 12 classifiers which are comparatively evaluated among one another and against the performance specifications stated in Section 2.1.1. Figure 3.16 shows a summary of the pattern recognition methodology. The code developed as part of the pattern recognition methodology is presented in digital appendices.

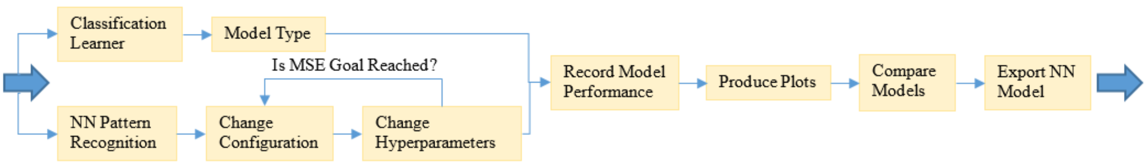


Figure 3.16: Summary of the pattern recognition methodology.

The steps taken during the pattern recognition process presented in Figure 3.16 are given below:

Step 1 An appropriate classifier was selected in the learner application and moved to the history data browser, where it was then possible to train the classifier. Depending on the number and type of classifiers selected and whether the processing computer supports parallel processing and has the relevant toolboxes, the models could be trained parallel to one another. This effectively assigned the cores of the CPU to different tasks and decreased the overall computation time.

Step 2 The training process, stated in Step 1 was performed for the following classifiers, that is coarse, medium and fine DTs, linear and quadratic DAs, linear, quadratic and cubic SVMs, and KNNs (coarse, medium and fine) models. These were computed parallel to one another with 12 parallel pool workers selected as the preference.

Step 3 Then, the accuracy, observation speed and training time of each classifier mentioned in Step 2 were recorded. For each classifier, a snapshot of the confusion matrix and ROC plot was taken in order to discuss the models' performance comparatively. This step concluded the classification learner to the GR approach.

Step 4 With a different input-output data structure to that used in the classification learner (see Step 5), the NN pattern recognition input-output data was imported into the MATLAB workspace.

Step 5 Using a script-based pattern recognition approach in MATLAB, the network size and depth was defined (that is, the number of neurons and layers needed were specified). As a rule of thumb, it was noted that the number of hidden neurons should not exceed twice those of inputs.

Step 6 Next, the hyper-parameters were defined. This includes the training function, data division ratio, type of data division, the performance function, activation function, validation number, number of epochs and training performance goal. These were all set to the default for the first pass model.

Step 7 Forming part of the final model, the NN classifier was optimised using a NN trials strategy. This was done by iteratively varying the number of hidden neurons and layers, which was kept constant once a minimal cost model was found. After that, other hyper-parameters were then individually adjusted, and the best solution kept for further development. This method is similar to genetic algorithm (GA) but does not account for the crossover associated with parent pairs.

Step 8 The process described in step Step 7 resulted in a cost optimised (low MSE) classifier with appropriate weights and biases. As such, the classifier was exported as a Simulink block and represented the final NN pattern recognition model before data post-processing developed in this dissertation. Additionally, the confusion matrix and ROC plots were noted and compared with the other classifiers proposed in this dissertation.

3.3.5 Data post-processing

The final step in GR model construction is data post-processing. Encompassed in the post-processing step is the smoothing, thresholding and class interpretation of the classifier outputs. This step takes place in a simulation environment and outputs classifications directly connected to other subsystems that make up the RPH. Figure 3.17 shows a summary of the data post-processing methodology. The Simulink model developed for post-processing is given in digital appendices.

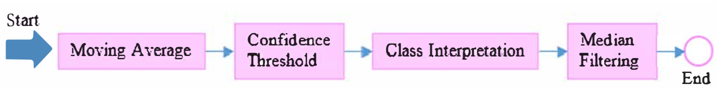


Figure 3.17: Summary of the data post-processing methodology.

The work done in the post-processing phase displayed in Figure 3.17 above is shown in Step Step 1 through Step 5 below:

Step 1 Using a *From Workspace* Simulink block, the feature data, along with time characteristics, are imported into the simulation environment. The block options are edited such that the data is a vector of 36 values repeating every time-step, where each time-step is inherited from the time column in the data. This data formed the input for the NN pattern recognition Simulink block.

Step 2 Upon connecting the input with the NN classifier block, the output of the classifier was probed using a scope. Through probing, it was determined that the classifier output (confidence) fluctuated between classes causing uncertainty. Thus a moving average with a buffer of 100 windows and overlap of 99 windows was used to help reduce the effect of rapid fluctuations. This process cut down on misclassification but added a 60ms delay.

Step 3 After investigating the output of the classifier, it was observed that the model predictions from Step 2 exceeded 40% confidence in cases where the true class was reflected. A threshold method was thus used to facilitate an additional correction for the misclassified inputs.

Step 4 Results from thresholding yielded an output which was interpreted into the corresponding classes, class 1 to 8 (illustrated later in Figure 3.36). These classes represented the 3 jaw chuck pinch, cylindrical grasp, hook grasp, lateral pinch, rest, spherical grasp, tip pinch and no class. The no class defined classifications which resulted in above 40% confidence over multiple classes, which signified classifier confusion.

Step 5 The class outputs from the previous step were further processed by using a median filter with a window length of 251 samples. This additional procedure produced an initial delay on start-up but did not significantly delay classifications made after 251 samples. This window length was found by trial and error and produced markedly better classification. The output from this step formed the boundary for the GR model and was then made the input of the SMA model.

3.3.6 Additional considerations

Additional considerations for the development of an sEMG GR classifier are as follows:

- The methodology, as it relates to this chapter, was performed concurrently with, and linked to, thoughts expressed in Chapter 4.
- The RMS feature extraction data produced in this chapter formed the input to the sEMG force predictive model.
- The uncertainty associated with resolution and truncation globally affected the proposed classifiers, hence it was not accounted for, particularly when comparing the performance measurements of the classifiers.
- It was noted that data acquisition over multiple days might vary dependant on the noise present in the surrounding environment.
- The ambient temperature in the Movement Lab was kept at 20°C, in order to minimise the build-up of sweat on the participants' skin during data acquisition.

3.4 Results and Discussion

This section details the findings produced during the development of a GR model, using raw sEMG data as an input. Following the path outlined in Section 3.3, the results unfold in a similar stage-wise manner. Analysing outputs along the development process led to the decisions and assumptions presented.

A trial and error process, together with a performance evaluation, was used to find the best GR classifier configuration. After acquiring data and subsequently preprocessing it, a few features were extracted and used to train comparative GR models. Using a NN trials approach, successful NN hyperparameter arrangements were discovered and guided towards outperforming other GR models explored in this dissertation, which yielded a deep BPANN GR model that approached the pinnacle in classification accuracy. Attributable to post-processing, a classification accuracy of 98.83% was achieved.

3.4.1 Raw surface electromyographic (sEMG) data

A predominant deciding factor in the quality of the recorded data is the equipment used. To this end, the sEMG data recorded from the participants overflowed with information, in turn containing enough characteristics for GR purposes.

Additionally, limitations were placed on achieving fine motor control since the muscles governing these actions primarily stem from the intrinsic hand. Residing in the superficial muscle-layer, the chosen forearm muscles were not subject to much noise emanating from the muscles radially in-line with it. Thus a wave decomposition step was not needed to extract the underlying signals.

Through observation, it was noted that the grasp apparatus used by the participants during grasping was well suited, dimensioned and catered for the biologically physical variations among participants.

Throughout data collection, 1800 data files (about 10GB of disk space) was collected. Each file represented the sEMG potential difference for an iteration of a grasp per participant, containing approximately 20000 samples over a 10s period for each muscle.

An example of a raw sEMG signal is shown in Figure 3.18. From this, it is clear that grasping occurred over the specified 6s interval, starting at 2s and stopping at 8s (as indicated in Step 4 of Section 3.3.3). It is also noticed that an offset of 25μV is present in the recorded sEMG signal, which may be due to any number of reasons. However, the offset remains persistent throughout the recorded sEMG signal across all muscles and participants, leading to a hypothesised sensor offset. The effects of the offset are global and do not influence class separability, thus may be omitted from further analysis.

The peak amplitude of the sEMG signal, either negative or positive, was observed to scale proportionally with factors such as muscle diameter; therefore, physically adept participants had, on average higher peak sEMG amplitudes. It is stressed that the sEMG signal shown in Figure 3.18 is not indicative of the sEMG signals acquired from the other muscles for a given grasp, seeing as different grasps activate, for different muscles independently. This mutual linking between grasp and muscle appears to be the factor that allows the pattern recognition algorithm to discern macroscopically between different grasps.

Through visual inspection of the sEMG signals, it was observed that similar muscles were active in both 3 jaw chuck grasp and tip pinch. This shows strong evidence that the grasps are similar with respect to extrinsic muscle characteristics and that an intrinsic evaluation may provide more promising separability.

A frequency-domain representation of the raw sEMG signal shown in Figure 3.18 is

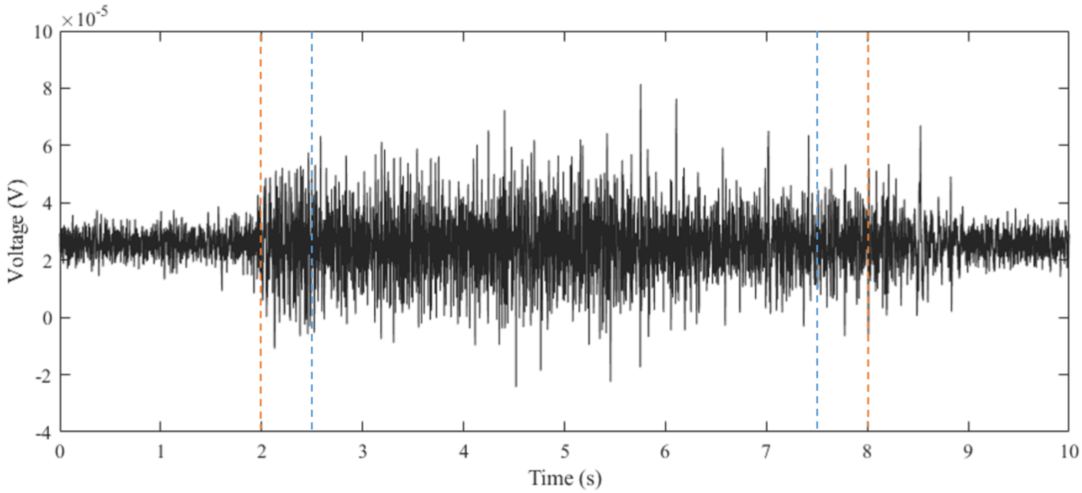


Figure 3.18: Raw flexor digitorum superficialis time-domain sEMG signal for cylindrical grasp.

presented in Figure 3.19. Not apparent at first glance from the time-domain depiction of the raw sEMG signal, is the harsh frequency located at 0Hz (see Figure 3.19). This is indicative of the direct current (DC) bias and may be removed by detrending the data. This was not directly explored in Section 3.4.2 as it did not present clipping issues; however, the peak was removed inconsequentially as part of the filtering process. Resonant frequencies were observed in the sEMG signal, with frequencies within the range of 0Hz to 150Hz. Which also resulted in notable PL interferences at 49.95Hz frequency.

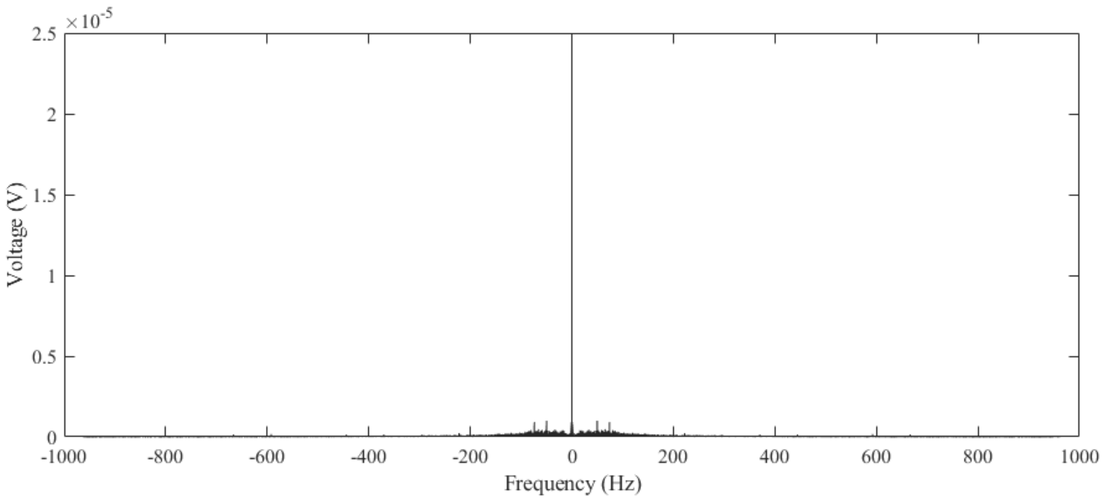


Figure 3.19: Raw flexor digitorum superficialis frequency-domain EMG signal for cylindrical grasp.

Linked to the thoughts expressed in Step 2 of Section 3.3.1, the moderate ambient laboratory conditions during testing resulted in a reduced risk of skin secretions, which in turn reduced the likelihood of skin impedance. This was enforced with regular examination of the skin.

The MVC data from each participant used for the comparison of MU activation potentials between muscles, individuals and tasks, was obtained at the end of grasp testing. This was to reduce fatigue-related skewing of data. In doing so, the MVC sEMG signals representing 100% muscle activation, were reduced in amplitude due to fatigue. The effects of this were noticed and corrected at a later stage using a dataset maximum search. However, it was evident that the alternative (that is acquiring MVC at the start) would have resulted in far worse and inconsistent data.

3.4.2 Pre-processed electromyographic data

The sEMG data acquired from the participants contained high-frequency noise and low-frequency artefacts. The removal of these superimposed interferences from the base sEMG signal is undoubtedly a key to the basis of a superior GR model and is examined and discussed next. The task of separating interference from useful sEMG signals is arguably non-trivial. This is due to the narrow frequency range associated with the critical sEMG signal characteristics coinciding with the frequency range of the interferences.

It is useful to establish a frequency spectrum boundary for the sEMG signal as an initial processing step. This is achieved by using a high and low-pass equivalent band-pass filter; were a subtractive filtering process follows, to remove the harsh individual frequencies and their harmonics. The peak frequencies discovered through a frequency analysis showed a 49.95Hz PL interference with resonant frequencies at 74.08Hz and 222.20Hz. These resonant frequencies are hypothesised PL harmonics augmented in wavelength by the surfaces in the testing facility, however, may equally be attributed to electromagnetic disturbances from other devices present in the vicinity of testing. The associated notch strengths found to provide adequate removal of these peak frequencies increasing peak frequency. This is because a higher notch strength results in a smaller -3dB localised bandwidth and thus drastically attenuates frequencies surrounding the notch frequency. Since GR critical sEMG signals occur in the 20Hz to 150Hz frequency range [30], care was taken to minimise the effect of the notch filters on frequencies surrounding the cusp of the peak frequency. This premise substantiates the initially lower notch strengths applied. Applying unique modifications of the filtering process for each iteration of testing is impractical, and thus it is assumed over the testing period that the interferences remained the same. This assumption allows for the batch processing of all test iterations with the same filter parameters. This assumption presents drawbacks, yet intriguingly, after randomly selecting and performing an FFT on multiple iterations (between participants), it is concluded that the assumption is well-founded and plausible.

Listing 3.1: Code excerpt taken from sEMG signal filtering script

```
%EMG1
EMG1 = EMG1';
EMG1_Bandpass = bandpass(EMG1,Bandpass_Rng,Fs);
emg_notch11=filter(num1,den1,EMG1_Bandpass);
emg_notch21=filter(num2,den2,emg_notch11);
emg_notch31=filter(num3,den3,emg_notch21);
```

Based on the filtering procedure, Figure 3.20 is realised and shows the filtered sEMG signal equivalent of Figure 3.19. Notably, the frequency range that characterises forearm sEMG signals, expressed in previous studies, is valid. This is seen by considering the power spectrum and spectral density of the filtered sEMG signal. Also, the sEMG signal frequency density seems to be consistent with literature following an exponential distribution [68].

The filtering process used in this dissertation leaves room for improvement, an example of which is seen in Figure 3.20, with the unaccounted for 375Hz (augmented fourth-harmonic) peak frequency. This ties back to the assumption stated in the previous paragraph. This limitation may be overcome by using an adaptive filtering technique, which considers each iteration individually and modifies parameters accordingly to remove outlier frequency spikes. Adaptive filtering may prove useful, while similarly retaining a reasonable amount of noise is also essential to promote classifier generalisation (degree of insensitivity towards noise).

The filtered sEMG signals are found to be suitable for feature extraction and comparable to published data [74]. Further analysis of the filtered sEMG signals showed that no distortion took place, and a high signal-to-noise ratio (SNR) was achieved. The SNR was not calculated but visually estimated due to a lack of information surrounding the noise and underlying signal amplitudes.

The last stage of sEMG data pre-processing is full-wave rectification. This basic procedure is fulfilled, as previously addressed, by taking the absolute value of the sEMG signal and is opposed to half-wave rectification, in which the negative parts of the sEMG signal is discarded.

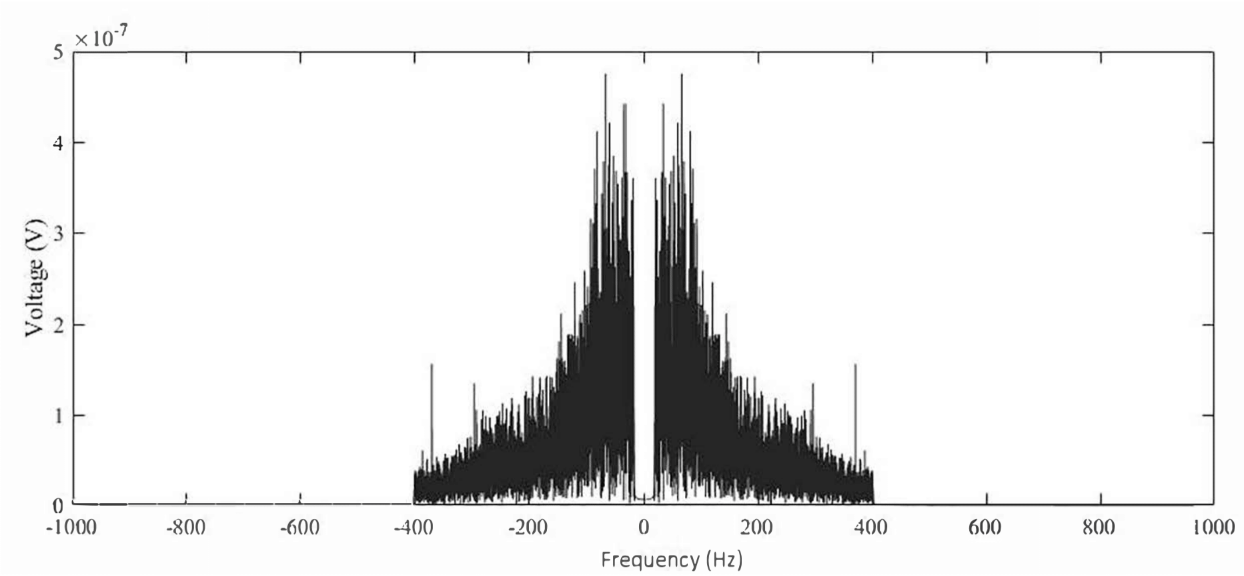


Figure 3.20: Filtered flexor digitorum superficialis sEMG signal for cylindrical grasp.

The aim of rectifying sEMG signals is to produce non-zero features. This concept is explained by aid of Figure 3.21, where the sEMG signal is seen to be strictly positive, and the resulting average over any non-infinitesimal slice never amounts to zero. This is an important necessity for pattern recognition algorithms which rely on features as inputs and is clarified by considering the MAV features. As such, the consequence of computing the MAVs for a non-shifted sEMG signal (without rectification) is a diminished signal that tends towards zero with significant overlap in the feature space. It now remains to extract features from the pre-processed sEMG data.

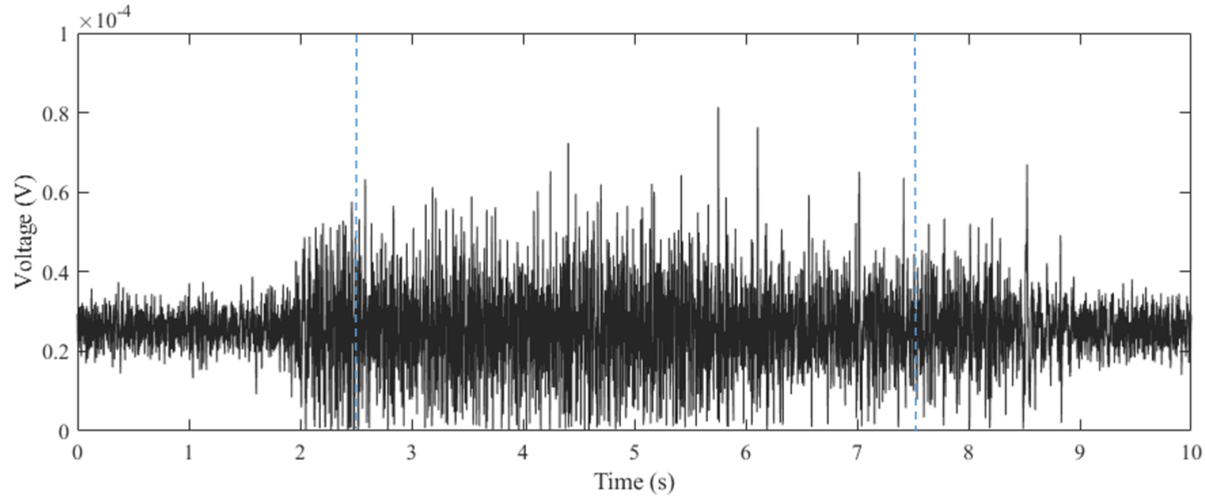


Figure 3.21: Rectified flexor digitorum superficialis sEMG signal for cylindrical grasp.

3.4.3 Feature extraction of electromyographic data

Feature extraction co-exists within many machine learning applications and is highly problem-specific. The chosen features, by definition, represent the dataset. If chosen incorrectly, the features may counteract one another or exaggerate unwanted dynamics, resulting in poor classifier performance.

The Hudgins feature set introduced in Section 3.1.3 is used with additional features to identify markers present in the sEMG signals, which are suggestive of the grasps performed. The addition of RMS and MAVS (originally part of the Hudgins set but omitted in recent years) features, present a two-fold improvement over the recently highlighted Hudgins features.

Figures 3.22 to 3.28 show the results after applying Equations 3.1.1 to 3.1.6, with a sliding window and overlap of 80ms and 20ms, to the rectified sEMG signal shown in Figure 3.21. The white Gaussian noise behaviour of the sEMG signal amplitude, present in Figure 3.21, is reduced after applying RMS feature extraction as shown in Figure 3.22. From this figure, it is also clear that the amplitude difference between the rest state, prior and after grasping, is preserved and the difference amplified. This property aids the classifier in accurately discerning between rest and active states. Also, the rapid transient behaviour of the sEMG signal is reduced through RMS, and the SNR inherently increased. Noticeable differences in the RMS of each muscle were observed between grasps. Participants exhibited slight differences in RMS while on the other hand retaining dominant muscle consistency.

The resulting MAV of the sEMG signal is closely linked in both the mathematical representation and output of the RMS. This sentiment is visible when comparing Figures 3.22 and 3.23, reinforcing the perception that these two features perform the same duty. Though partly true, this does not consider the added benefit of reinforced characteristics, noted in the previous section, especially when a weighted feature approach to classification is taken. Dissecting Figure 3.23, it is apparent that MAV is impervious to rapid peak sEMG fluctuations and carries a deeper level of detail, as depicted by secondary rises and falls superimposed in Figure 3.22. That being said, the output achieved from MAV feature extraction closely resembles RMS feature extraction and the parameters

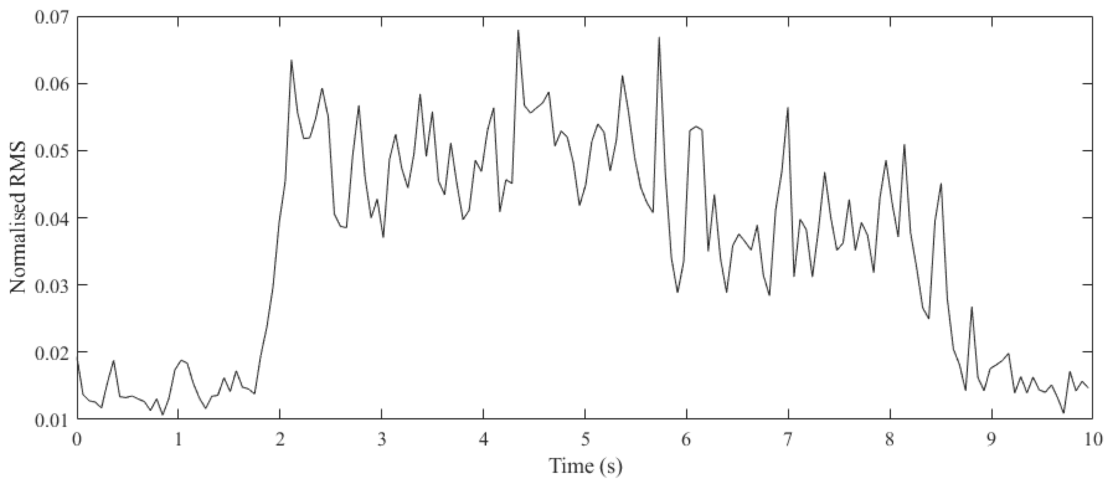


Figure 3.22: Normalised RMS feature of the flexor digitorum superficialis sEMG signal for cylindrical grasp.

derived from it.

There is an observable separability between the MAV acquired from each muscle, for each grasp. Additionally, each participant presented unique coupling between muscles, hinting at the overcompensation of certain muscles due to lack of strength in other muscles.

The MAVS of the sEMG signal is shown in Figure 3.24. From the figure, it is easy to see that not a lot of variation is present in the grasping portion of the sEMG signal. However, this differs slightly between grasps and participants, but ultimately displays the poor ability to expose differentiating underlying signal properties, calculated using two segments. Besides this criticism, the feature is shown to be useful in determining rest versus action states. It may be inferred from Figure 3.24 that MAVS is also a good indicator of steady-state.

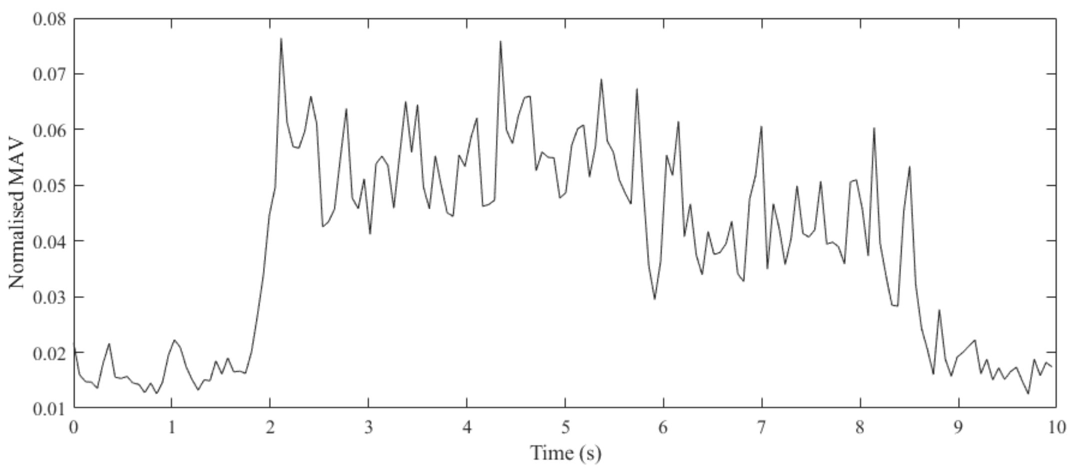


Figure 3.23: Normalised MAV feature of the flexor digitorum superficialis sEMG signal for cylindrical grasp

MAVS is notably not positive definite, thereby influencing the choice of suitable NN activation functions (tan-sigmoid) during classifier training. A commonality among grasps and participants saw an abrupt sharp peak indicating the start of the grasp, followed by considerable fluctuations about zero and ending with another abrupt peak indicating the end of the grasp.

Closely related to ZC is SSC, which is a time-domain representation of frequency-based characteristics. From Figure 3.25, it is observed that this feature is inversely proportional to muscle activity. Mindful of this, a lower SSC represents a grasp differentiating region, while a higher SSC represents rest. Additionally, high SSC could indicate noise [74].

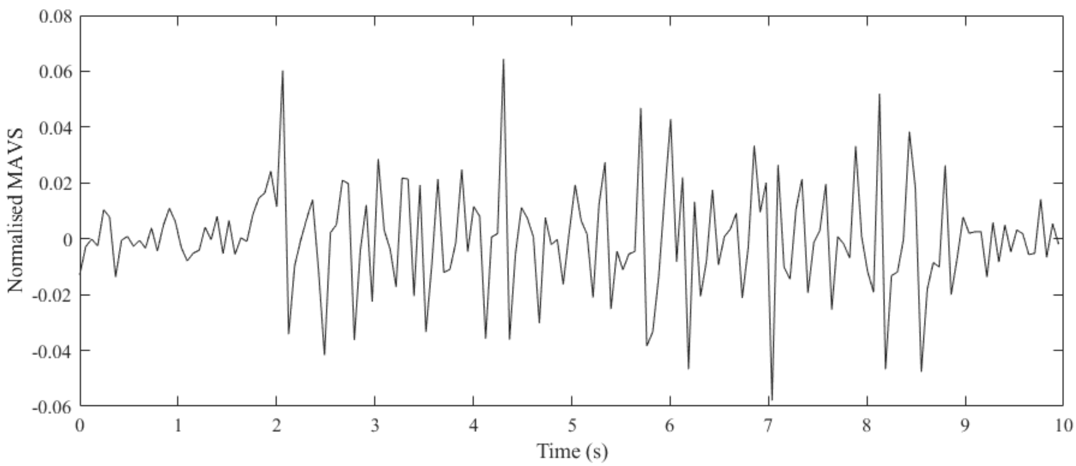


Figure 3.24: Normalised MAVS feature of the flexor digitorum superficialis sEMG signal for cylindrical grasp.

This property, however, changes as the threshold in the defining equation is increased from zero. As the threshold is increased, the noise and rapid sEMG signal fluctuations are attenuated; however, the risk of gating vital signals is increased. Through trial and error, a threshold of 5μV is selected in this study. The resultant output shown in the rest region of Figure 3.25 seems to suggest the threshold is too low, but a stance was taken to err on the side of caution and progress with the chosen threshold. Across the various participants and grasps, the only noticeable changes in SSC was the hierarchy of dominant muscles.

The output of WL feature extraction is similar in trend to that of the RMS and MAV. Though, this feature differs significantly in the tail end depiction. As shown in Figure 3.26, the WL seems to sustain the signal post-contraction in comparison to Figures 3.22 and 3.23. This introduces a trailing offset and could adversely affect the classifier training progress. More importantly, certain peaks from the rectified sEMG signal are accentuated. These represent regions where high-density information is present.

It would be interesting to explore adaptive bandwidths that exploit this phenomenon. It appears from in literature that WL is a principal constituent of feature sets resulting in exceptional classification [74]. Across grasps, inactive and active muscles varied, with a discovered resemblance occurring between 3 jaw chuck and tip pinch.

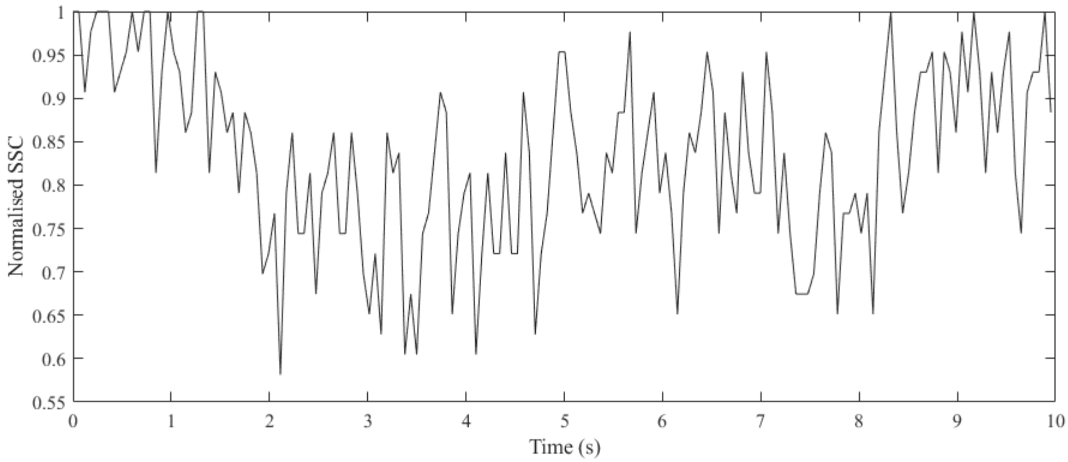


Figure 3.25: Normalised SSC feature of the flexor digitorum superficialis sEMG signal for cylindrical grasp

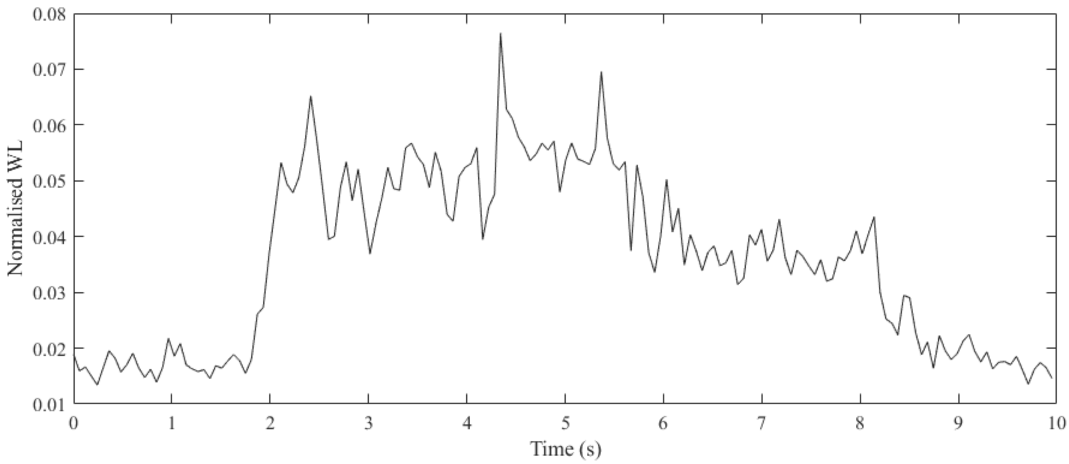


Figure 3.26: Normalised WL feature of the flexor digitorum superficialis sEMG signal for cylindrical grasp.

The inverse relationship between sEMG signal amplitude and ZC is apparent in Figure 3.27 but not strongly correlated. As such a threshold other than zero would remove unwanted portions of the sEMG signal, though still attenuating background noise. There was no discernible difference in ZCs between muscles for a given grasp for certain participants; however, rest is an utter exception, indicating a larger ZC for flexor muscles when compared to extensors. This may be indicative of the set position of the hand during rest being in partial flexion.

Inter-participant variation of ZC, where certain participants exhibit noticeable differences compared to others, is a possible indicator of cross-talk and is observed to correlate with muscularity. ZC feature extraction yielded results that were different between participants with similar SSC muscle hierarchy.

Figures 3.22 to 3.27 are in a normalised state. Doing so allows a fair baseline comparison between participants, muscles and tasks. Suggested protocols for acquiring reference forearm sEMG signals (i.e., MVC) indicate a resistive approach is to be used. This study found participant-specific MVC values to be on average four times higher than

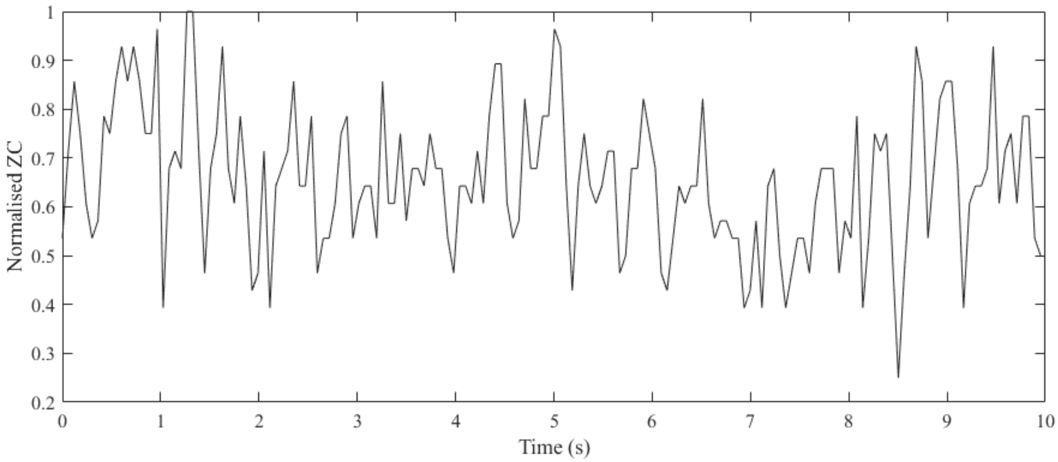


Figure 3.27: Normalised WL feature of the flexor digitorum superficialis sEMG signal for cylindrical grasp.

peaks located in each iteration, which is supported by the literature [5].

Table 3.4 shows the dominant active muscle/muscles during the investigated grasps that fall within the scope of the six muscles investigated in this dissertation. This is based solely on RMS measurements and gives a good indication of the main muscle/muscles responsible for each grasp. Demonstrated in Table 3.4 are the key target muscles, which are namely the extensor digitorum and extensor pollicis longus.

Table 3.4: Dominant muscle/muscles with corresponding MVC.

Grasp Type	Dominant Muscle or Muscles	Dominant Muscle MVC (mV)
3 Jaw Chuck Pinch	Extensor digitorum and extensor pollicis longus	3.35 and 1.98
Cylindrical Grasp	Extensor digitorum and flexor pollicis longus	3.35 and 1.49
Hook Grasp	Flexor carpi ulnaris and extensor pollicis longus	1.16 and 1.98
Lateral Pinch	Flexor carpi ulnaris and extensor digitorum	1.16 and 3.35
Rest	Extensor pollicis longus	1.98
Spherical Grasp	Extensor digitorum and extensor pollicis longus	3.35 and 1.98
Tip Pinch	Extensor digitorum and extensor pollicis longus	3.35 and 1.98

The output produced from feature extraction was trimmed to a 6s region, as outlined in Step 4 of Section 3.3.3. This is done to enable the allocation of classification labels associated with the various grasps. The trimming process is subject to mislabelling as the participants may have started the grasp late or terminated early, in turn, adversely affecting the performance of the classifier. Methods for segmenting the regions of an sEMG signal exist and typically involve SSC or ZC [74].

These segmentation methods were, however, not explored; rather, a 0.5s trimming tolerance was instituted. This effectively reduced the mislabelling associated with the late start or early termination and resulted in a 5s region, defining the specific grasp, as shown in Figure 3.28. Each 5s region was labelled with the corresponding grasp, thereby comprising the input-output matrix of features and labels used to train the classifier. To further clarify, the segmentation of data was not implemented on the raw sEMG signal; but rather on the features that were extracted, hence the MAV feature portrayed in Figure 3.23. The segment shown in Figure 3.28 effectively is retrieved from Figure 3.22.

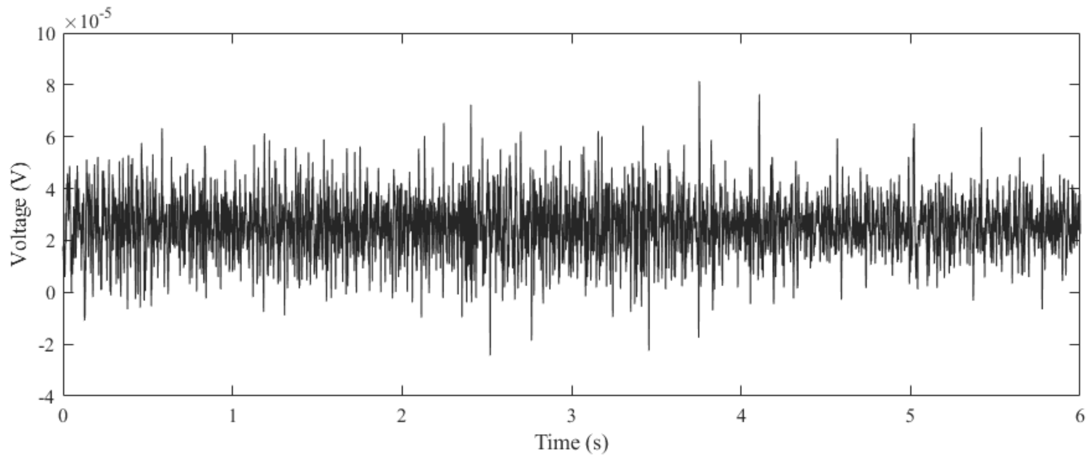


Figure 3.28: Trimmed and normalised RMS feature of the flexor digitorum superficialis sEMG signal for cylindrical grasp.

3.4.4 Grasp recognition models

GR is based on the identification of inputs, in this case, a feature matrix. In order to accomplish this, a model is trained using a governing algorithm that takes into account the "dead truth" (i.e. the labelled outputs) and attempts to learn the underlying patterns not readily apparent. The input feature matrix stems from the feature extraction process and results in a matrix with 147000 data point (rows) and 36 features (six features by six muscles, columns). The outputs consist of either a 147000 (rows) by seven (columns) matrix, for the common classifiers, or 147000 (rows) by one (column) vector, for the BPANN classifier. This difference bears no significance and is simply due to the default data structures MATLAB requires.

To compare the effectiveness and optimise the deep BPANN classifier a range of other common classifiers are trained with the same input features and dead truths. The results from these analyses are presented in the following two sections, where the first considers common classifiers and the second a deep BPANN classifier.

In both sections, the real-time application potential is explored and the system performance evaluated against performance specifications reported on in Section 2.1.1. All models presented are substantiated with accuracies falling within the ranges found in literature (refer to Table 3.2). Feature reduction was not considered but may prove useful in future studies, and if explored, may result in a shorter computing time and faster training convergence.

3.4.4.1 Classification learner models

The classification learner models presented are developed using the MATLAB framework previously shown in 3.11. These models cover a large majority of common classifier models, each demonstrating different working principles. Additionally, the performance of each model is explored using a host of metrics. Notably, the newly discovered inadequacy of Cohen's Kappa is substantiated, and confusion matrices and ROC plots are presented and analysed. The performance of the best classifier in this section is used as the goal for the deep BPANN classifier. The performance related to the training of coarse, medium and fine DTs, linear and quadratic DAs, linear, quadratic and cubic SVMs and coarse, medium and fine KNNs are presented in Table 3.5.

With an average accuracy of 83.63%, SVMs prove to be better at classifying than the other classifiers considered. This feature is, however, lessened when considering both the time required to train and the observation speed attained by the classifier. The training time seems to suggest that a powerful processing system is required which does not lend well to embedded applications such as GR using sEMG signals.

Taking the longest to train, the cubic SVM performs the best with an accuracy of 90.01%, which is confirmed in literature [74]. Moreover, investigating the accuracy difference between the linear, quadratic and cubic SVM models raises an interesting conclusion. A quadratic hyperplane sufficiently maximizes the margin of the hyperdimensional representation of the input feature matrix into distinct classes.

Following in classifier ranking are KNN classifiers, which are the second-best with an average accuracy of 79.35%. Also, the combined training time is the second largest, and the observation speed proves to be inadequate. This is because of the sampling speed of the acquired sEMG signals exceeds the maximum observation speed, which if implemented, would cause signal loss or lag (if buffered). Since KNN classifiers are non-parametric, they are computationally expensive, requiring the storage of training data. This is an added disadvantage hindering a KNN classifier from being used in sEMG GR.

A common phenomenon is observed between the KKN classifiers, that is a finer classifier (k=1; one neighbour). This results in overfitting, a rough classification plane and ultimately poor accuracy. On the other hand, a coarse KNN classifier (k=100) is smooth and underfits the data, resulting in the medium KNN classifier (k=10) being superior.

With an average accuracy of 70%, the DA classifiers are third in performance among the four investigated classifiers and exhibit larger observation speeds and fast training times. A noticeable difference in accuracy between the linear and quadratic DA classifiers suggest the input feature data cannot be acceptably described using a singular covariance matrix, but rather a covariance matrix for each class is needed. In addition, DA classifiers assume the feature data follows a Gaussian distribution. This is a valid assumption; however, the small number of participants involved in this study does not provide an adequate sample size, which possibly accounts for the less than satisfactory accuracy. With the worst average accuracy of 57.41%, DT classifiers are ill-suited for GR and would result in an unreliable RPH. The fact that these classifiers have short training times means that if the accuracy is improved upon, it would be possible to develop RPHs that are uniquely calibrated rapidly. Also, the combined observation speed is the highest for this classifier and would suit computationally weak processing systems. Considered next is the F1 score and MCC for the classifiers, which is derived from Equations 3.1.14 and 3.1.15.

The F1 score heightens the influence of false positives and false negatives and gives a slightly more pessimistic outcome than accuracy, in turn striking a balance between precision and recall. From the results in Table 3.5, the F1 score is highly correlated with accuracy and the MCC, and correspondingly the highest for the cubic SVM classifier. Inconsequentially, the MCC is also the highest for the cubic SVM classifier. The MCC considers a proportion of the four categories (TN,TP,FN and FP) of the confusion matrix and is, therefore a better measure for unbalanced data. The agreement between

F1 score, accuracy and MCC is indicative of a balanced feature data set.

The astonishing newly discovered inadequacy of Cohen's Kappa as a performance measure for multi-class classifiers presented by Delgado and Tibau is substantiated in this dissertation. This is clearly visible as the Cohen Kappa performance measure behaved randomly. Therefore, the performance measure was not further considered, and its meaning disregarded. Graphical representations of the performances for classifiers will be focused on next.

A confusion matrix is an important performance measure frequently used in the assessment of a classifier. Apart from being easy to understand, it gives valuable information and quantifies the number of misclassifications among classes. Figures 3.29a to 3.29d show the confusion matrices associated with the best classifier architectures from the GR classifiers investigated. Two significant regions exist within each confusion matrix, that is the lead diagonal (green) and the off-diagonal (red). Where the former represents the correct classifier predictions and the latter the misclassifications (incorrect predictions). These regions provide vital information on pitfalls for the classifier/s they represent.

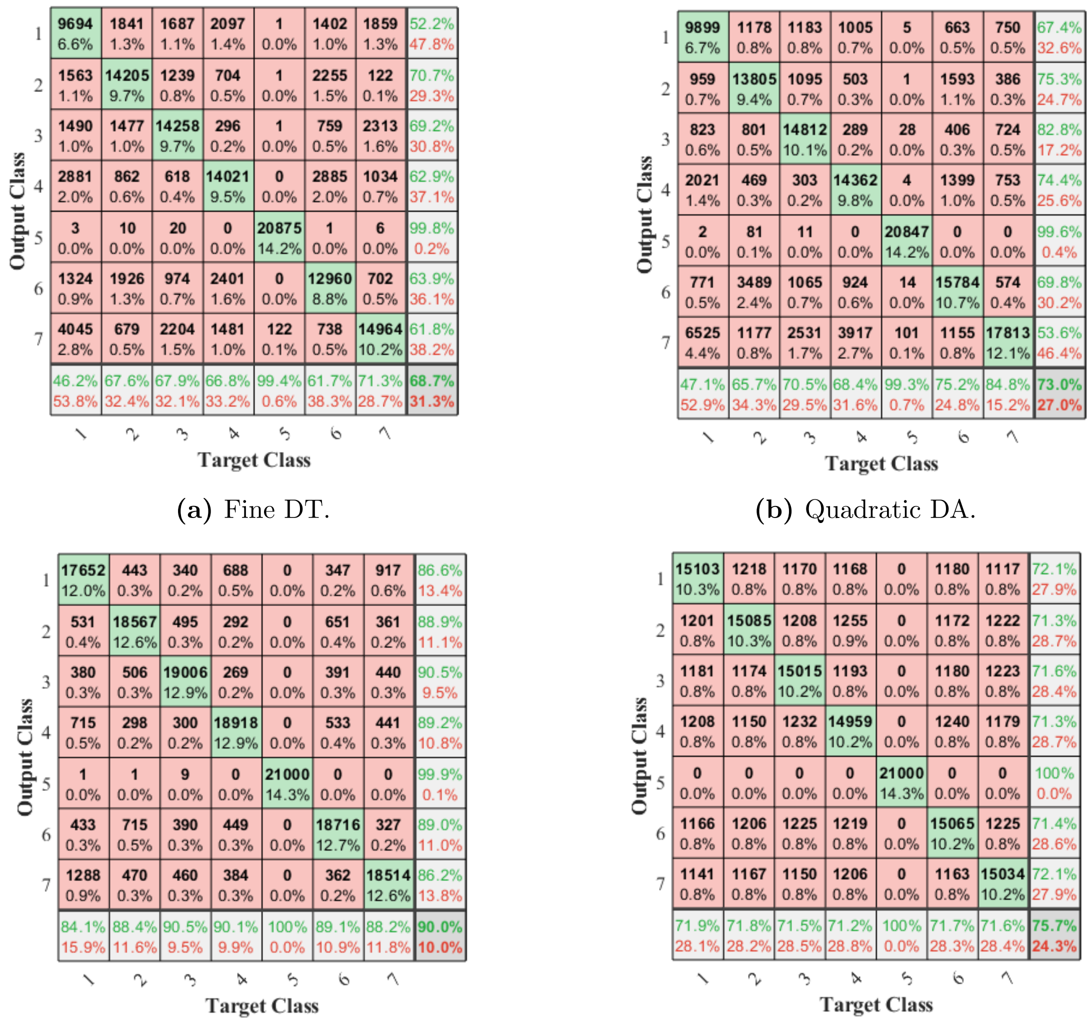
Table 3.5: Performance of classification models.

Model	**Accuracy (%)**	**Kappa**	**F1 Score**	**MCC**	**Observation Speed (Obs/s)**	**Training Time (s)**
Coarse DT	43.23	0.5686	0.3459	0.3232	340000	14.42
Medium DT	60.27	0.3835	0.5942	0.5409	370000	14.71
Fine DT	68.73	0.2178	0.6857	0.6342	590000	23.39
LDA	66.95	0.2591	0.6677	0.6134	160000	11.72
Quadratic DA	73.04	0.0474	0.7424	0.7066	210000	7.55
Linear SVM	72.79	0.1000	0.7261	0.6818	57000	2515.70
Quadratic SVM	88.10	0.5685	0.8941	0.8767	3600	5133.9
Cubic SVM	90.01	0.7984	0.9106	0.9053	4100	8492.6
Coarse KNN	80.94	0.2636	0.8085	0.7900	690	691.45
Medium KNN	81.43	0.4185	0.8132	0.8051	710	731.86
Fine KNN	75.68	0.2096	0.7487	0.7412	690	698.69

Moreover, the number in each block illustrates the number of observations said to take place between the target and output class. The total of these observations equates to the 147000 data points previously discussed. However, due to the uniformly selected holdout validation of 25% d (see Section 3.3.4), a fraction of the observations are not depicted and kept aside for validation purposes.

More importantly, the confusion matrices indicate the worst and best-classified classes. The best-classified class is unanimously class five, which is rest and is substantiated with the low off-diagonal observations. This is expected since the sEMG signals associated with rest are minimal, across all muscles, in comparison to the other grasps, therefore easily discernible. An interesting point gathered from the worst-classified class, or rather class pair, apparent in all the confusion matrices, is a discovered confusion between 3 jaw chuck grasp and tip pinch. This is evident considering the intersections of classes one and seven, which account for the largest misclassifications.

Taking a step back to the root cause of this, it is obvious that the grasps are similar and therefore require similar muscle actions. It is therefore suggested that future studies considering both grasps, consolidate efforts to only one grasp seeing as the grasps are seemingly coupled in origin. The classes referred to in the confusion matrices are sequentially 3 jaw chuck pinch, cylindrical grasp, hook grasp, lateral pinch, rest, spherical grasp and tip pinch.



(a) Fine DT. (b) Quadratic DA. (c) Cubic SVM. (d) Fine KNN

Figure 3.29: Optimum threshold confusion matrices of the best performing classifiers.

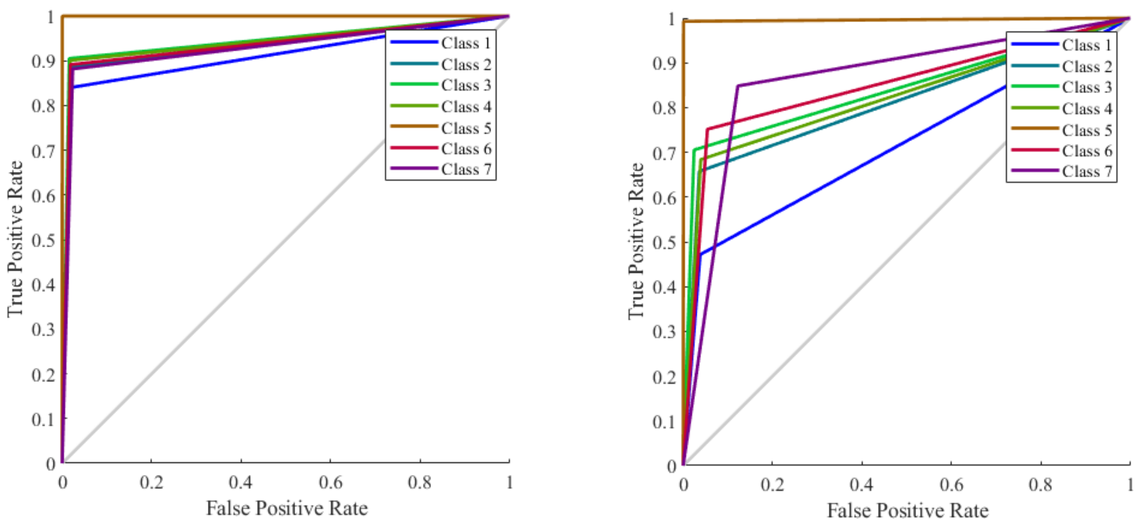
Typically an analytic tool used in medical studies but finding significance in machine learning, the ROC plot visually presents the trade-off between specificity and sensitivity. These two components are inversely related, thus making a ROC plot a valuable graphical means for locating the optimum threshold. Examples of ROCs are shown in Figures 3.30a and 3.30b. From these figures, a similar sentiment shared with the confusion matrices is observed.

A ROC plot is useful in that it helps rank the performance of each class visually, where a poor performing class classification results in a lower AUC. Extending this concept to Figure 3.30a, the ranking of class classification performance (i.e., from worst to best) starts with class 1, followed by class 7, 2, 6, 4, 3 and 5 in that order. Note that the class representations are that same as previously mentioned concerning the confusion matrices, and remains constant throughout this chapter. For Figure 3.30b, the ranking remains the same. To justify the ranking in these figures, it is crucial to first distinguish between good and bad ROC plot results. The 45°diagonal is equivalent to random classification while a right-angle located in the top left corner represents perfect classification. A middle ground with type I and II errors, between these two extremes is what is seen in both figures.

The ROCs for each class shown in the figures are different from those commonly expressed, by way of only consisting of two lines. This is because the optimum threshold was selected before plotting the ROCs. One method used to determine the optimum threshold is Youdens distance; however, a different feasibility method was used. It relies on calculating a cost slope, shown in Equation 3.4.1, and sliding it from the top left corner down until it intersects the ROC plot. This point denotes the best theoretical trade-off between specificity and sensitivity.

$$S = \frac{P(Cost(P|N) - Cost(N|N))}{N(Cost(N|P) - Cost(P|P))} \tag{3.4.1}$$

Equation 3.4.1 defines the slope of the line used to find the optimum threshold, where P and N represent the positive and negative classes. Furthermore, $P = TP + FN$ and $N = TN + FP$. Where TP, FN, TN and FP is as stated in Section 3.1.4. It is worth noting, the confusion matrices shown in Figures 3.29a to 3.29d are developed for the optimum threshold case.



(a) SVM ROC plot.

(b) KNN ROC plot.

Figure 3.30: Optimum threshold ROC plots of the best performing classifiers.

In summary, the cubic SVM exhibited the best overall performance among the non-NN based classifiers investigated. This was further substantiated by considering the accuracy, F1 score, MCC, confusion matrix and ROC plot. This classifier is well-suited as a GR classifier and is shown in literature to be effective in doing so [74].

Henceforth, although being trained in parallel with the BPANN classifier, the accuracy was also used as a baseline for the NN-based classifier to improve upon. The SVM classifier is notoriously computational demanding during training, thus presenting a weak case for unique participant-specific classifiers. This issue, however, does not translate further, making the SVM classifier suitable for real-time applications provided the classifier is already developed.

3.4.4.2 Neural network model

The deep BPANN classifier proposed is developed using the MATLAB framework previously shown in Figure 3.12. Two trial regimes are presented, one for a shallow BPANN and the other for a deep BPANN. For large datasets, a representative sample is typically used, but due to the relatively short training times observed, the trials were trained using the complete dataset.

Accuracy and NMSE were the only two performance specifications used in the selection of the best NN architecture to explore further. This was done to reduce computation time, particularly for the deep NN trials, and was found to be acceptable when considering the directly proportional relationship between F1 score, MCC and classification accuracy presented in 3.5. The results substantiate the indeterminate relationship between NN architecture and performance. The accuracy of the best classifier in Section 3.4.4.1, which is 90.01%, is used as the goal for the deep BPANN classifier.

The first trial regime includes a single layer BPANN classifier, representing a shallow NN, which is a universal approximator and is trained using the SCG algorithm. In each trial, the number of hidden neurons is incrementally set, maxing out at the search limit of 50 neurons. This limit follows a rule-of-thumb which states that the number of hidden neurons in each layer must be less than double the number of inputs [75]. Doing this ensures that the trained BPANN will be less prone to overfitting, however complimentary literature seems to suggest a far higher limit is possible [61], but may not necessarily yield better performance.

An example of this is shown in Figure 3.31, where the accuracy seems to plateau around the upper 70% as the number of neurons in the hidden layer increase. Additionally, in Figure 3.31, the latter fluctuation of accuracy with the number of hidden neurons validates the indeterminate and highly problem-specific relationship between the BPANN classifier accuracy and architecture. This is substantiated further by the sudden dip in accuracy observed at 44 neurons. That being said, the indeterminance is only predominant in excess of 20 neurons. Therefore, initially, the accuracy is found to increase with increasing hidden neurons and follows a family of curves of the form $f(x) = 1 - e^{-kx}$.

In summary, increasing the number of hidden neurons yields better accuracy up to a point, after which very little change is observed. The maximum classification accuracy occurred at 41 neurons and is pointed out in Figure 3.31. Moreover, keeping the hyper-parameters constant in a set or default state allowed adequate comparison between trials with different neurons. This includes fixing the number of training epochs to 200 and retaining a train, validate and test ratio of 70, 15 and 15 with interleaved indices. Also, the initial weights and biases were set to a non-zero value to remove bias that could emanate from random starting conditions.

Comparatively, the accuracies achieved by the shallow BPANN classifiers, after minimal training, were similar to that of KNN classifiers and outperformed DT and DA classifiers. Both the quadratic and cubic SVM classifiers produced far greater accuracy than the shallow BPANN classifiers. This, however, does not assert dominance of the SVM classifiers over the BPANN classifiers since the slow convergence properties of the SCG training algorithm may result in significant performance improvement with an increase in the number of training epochs.

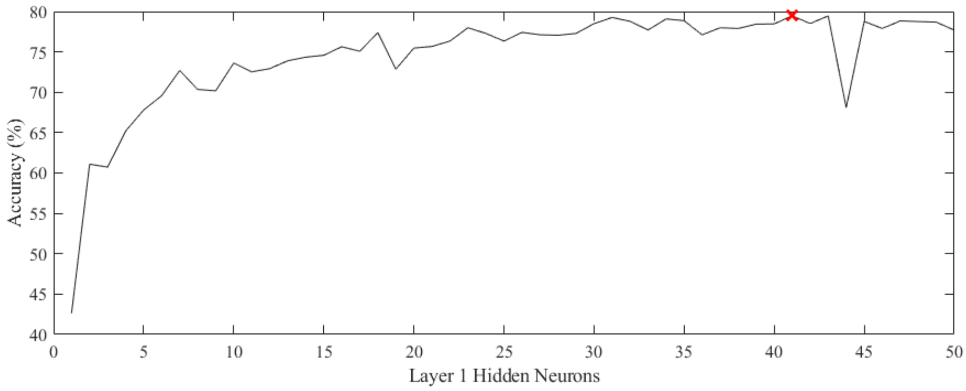


Figure 3.31: Single hidden-layer NN trials accuracy.

Complimentary to Figure 3.31, the detailed accounts of ten partly random selected trials, of the 50 available, are given in Table 3.6. From Table 3.6, it is noted that nine neurons, which is equivalent to a quarter of the number of inputs, produce an accuracy of less than 10% shy of the maximum accuracy. This indicates the possibility for a few neurons to constitute an acceptable classifier architecture.

In addition, the NMSE was found to scale accordingly with the accuracy of the classifier and was used as the cost function for the SCG algorithm, upon which the performance of successive epochs was improved by decreasing the residual. The training time was observed to increase with the number of neurons. It should be noted that this is not entirely true as changes in the processing requirements of behind the scenes operations, i.e., those attributed to the operating system, resulted in outliers which are observed for the 21 and 41 neuron trials in Table 3.6.

A startling observation from Table 3.6, is the accuracy achieved for the case of a single tan-sig hidden neuron. Coupled with the effect of the output Re-LU neurons, the higher than expected accuracy suggests low-order non-linear separable classes which echoes the sentiments expressed in Table 3.6.

Table 3.6: Shallow BPANN NN trials; 200 epochs and SCG training algorithm.

No. of Hidden Neurons	Accuracy (%)	NMSE	Training Time (s)
[1]	42.6560	0.0919	18.3751
[4]	65.1367	0.0635	23.3666
[9]	70.1422	0.0532	29.0127
[15]	74.5926	0.0521	34.0290
[21]	75.6731	0.0425	65.9060
[28]	77.0714	0.0512	51.6455
[35]	78.6864	0.0453	58.7226
[41]	79.4741	0.0506	56.9615
[44]	68.1272	0.0670	61.9738
[50]	77.6993	0.0438	68.5838

The second trial regime comprises a double hidden-layer BPANN classifier, representing a deep NN trained using the SCG algorithm. In each of the 2500 trials, the hidden neurons in the first layer were incrementally set. After which the hidden neurons in the second layer were subsequently set. This is to say, with each iteration of the first layer, there are 50 corresponding iterations of the second layer. The limit of 50 neurons again follows the rule-of-thumb previously stated.

A surface plot of the accuracies associated with the different first and second layer neurons is shown in Figure 3.32 and resembles a tablecloth surface with the furthermost overhangs trimmed. The accuracy again seems to plateau around the upper 70% as the number of neurons in the hidden layers increases. Also, the fluctuation of accuracy or rather unevenness of the observed plateau, again suggests the indeterminate and highly problem-specific relationship mentioned. Drawing to a point, increasing the number of hidden layer neurons yields better-expected accuracy, which is substantiated through the consideration of the colour-map alongside Figure 3.32.

The maximum observed accuracy occurred at 40 first and 42 second layer neurons and is pointed out in Figure 3.32. The hyper-parameters were kept constant and mirrored the states of the previous single-layer trial regime. The accuracies achieved by the best deep BPANN classifiers on average exceeded the shallow BPANN classifiers and therefore remains similar to that of KNN classifiers and outperforms the DT and DA classifiers. Furthermore, the quadratic and cubic SVM classifiers still yield better accuracies than the deep BPANN classifier. The slight improvement in accuracy of the deep BPANN over the shallow BPANN, alone, does not constitute adoption of the model, especially when considering the drawback of additional computation time. On the contrary, deep NNs have been shown to possess superior performance, most notably in cases where GR involves multiple DoFs [75].

In light of this potential, the slightly increased accuracy and the fact that the SCG training algorithm is yet to converge on an optimal classifier, a deep BPANN classifier solution was adopted as the final classifier model. The prior statement made, asserting the performance and computation time dominance of the SCG training algorithm, relating to classifiers, is substantiated by noting that the dataset is large and that pattern recognition generally operates in the saturated region. Therefore go-to algorithms such as LM will typically take longer to compute and struggle to converge. Reiterating, the GR classifier chosen for further training and optimisation, through hyper-parameter tuning, comprises two-layers with the number of neurons in each layer set to those found to yield the best accuracy, that is 40 and 42.

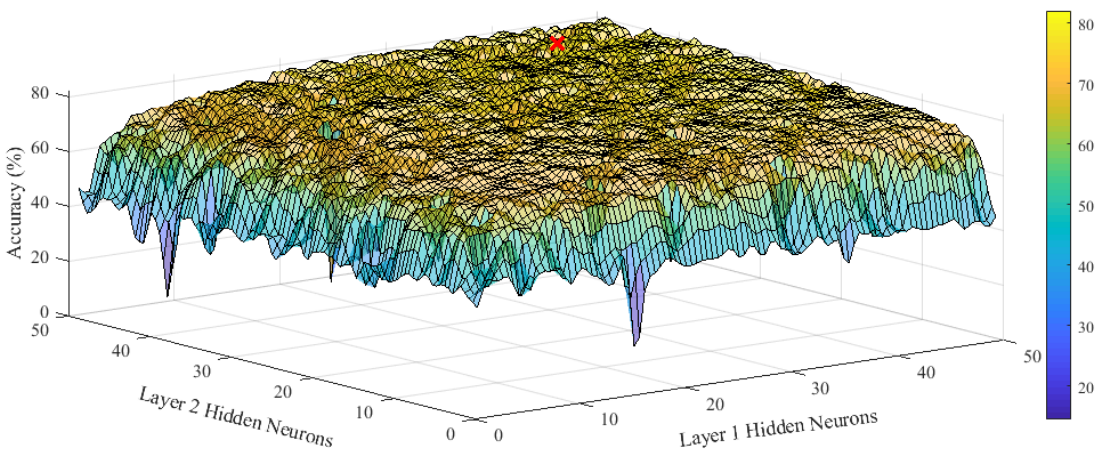


Figure 3.32: Double hidden-layer NN trials accuracy.

In addition to Figure 3.32, the detailed accounts of ten partly random selected trials of the 2500 available, are given in Table 3.7. Two interesting entries in Table 3.7 includes that of {1,39},{40, 42}. Where for instance entry $\{x, y\}$ would indicate that x represents the number of first layer neurons and y the number of second layer neurons. These entries ({1, 39} and {40, 42}) represent the classifier architectures that had the worst and best accuracies respectively.

The poor accuracy of the {1,39} entry, though containing a sizeable number of neurons, may be explained by the training process reaching a validation or stopping criteria. This, however, is not a likely suspect for poor performance, but rather it may be due to the incompatibility of the two layers for the given dataset and specified problem. Converse to this is the 40,42 entry. It is unique in that it is generally expected that the number of neurons in the superseding layers be less than the previous layer. This idea is apparent when considering a two-layer NN. Explaining this statement, the number of neurons in the first layer may be thought of as being equivalent to the number of linear surfaces used to separate the classes whilst the number of neurons in the second layer is equivalent to the cusps linking the linear surfaces, which are the turning points. As previously established, the {40,42} BPANN configuration performs the best which may result from the complimentary number of neurons in each layer linking with the features learned throughout the levels of abstraction. The NMSEs shown in Table 3.7 were similarly found to scale accordingly with the accuracy of the classifier. Likewise, the training time was found to increase, though irregularities were observed, which may prescribe the same cause as stated for Table 3.7.

Table 3.7: Deep BPANN NN trials; 200 epochs and scg training algorithm.

No. of Hidden Neurons	Accuracy (%)	NMSE	Training Time (s)
[1,1]	39.3701	0.0934	19.8822
[1,39]	24.2388	0.1040	43.6044
[6,4]	64.7041	0.0674	30.8187
[15,16]	73.2680	0.0534	46.9539
[23,35]	76.5537	0.0496	70.8444
[26,8]	74.5878	0.0509	39.1433
[35,19]	75.0735	0.0495	79.7534
[40,42]	82.9601	0.0391	107.8979
[46,21]	77.4190	0.0459	91.3682
[50,50]	79.1048	0.0434	127.4806

Following the NN trial regimes conducted, a deep BPANN classifier with 40 first and 42 second layer neurons was further trained using the SCG algorithm and optimally tuned. The hyper-parameters established in the shallow NN trial regime were used here and kept constant during training, but the number of training epochs was increased to 15000, and an NMSE goal of 0.008 was set. The goal was found to be in-line with or exceeding the performance of similar classifiers published in literature [75]. Prior to training, the weights and biases were initialised using the Nguyen-Widrow initialisation function and training commenced.

The deep BPANN classifier completed the 15000 training epochs in a time of 7484s, with an NMSE of 0.0095. After which, it was observed that the accuracies relating to the training, validation and testing sets were respectively in the ratios of 70, 15 and 15 to the whole dataset. This exceeded the accuracies that were achieved by the cubic SVM classifier and amounted to 90.81%. Furthermore, the accuracies differed among sets by less than 0.2%, indicating the possibility for further training without risk of overfitting.

Hence, the deep BPANN classifier was iteratively further trained by capturing the weights and biases after each training iteration and initialising the NN with these parameters upon each further training iteration. As a result, the deep BPANN classifier was trained for a further 5000 epochs before it was decided that overfitting was becoming more predominant in the classifier. This was visually verifiable when observing the accuracies of the sets as they deviated further apart, meaning that the training accuracy improved as the validation and test accuracies decreased or remained equivalent. Following this, a trial-and-error-based hyper-parameter search was conducted, yielding a resultant BPANN GR classifier accuracy of 92.34% with an NMSE of 0.0082, outperforming ANFIS [74], BPANN with LM training algorithm [76], FIS [75], SVM [49] and error backpropagation NN [75] classifiers. In addition, the remarkable performance was found to lie within the top tier of GR classifiers [77]. Although having implemented a manual stopping criteria for training in this dissertation, other techniques exist with the sole purpose of preventing overfitting. These techniques include L1 and L2 regularisation, which is, in essence, regularisation with an added coefficient.

Shifting focus to the architecture of the deep BPANN classifier, illustrated in Figure 3.33. The total number of weights equate to 3503 and is computed using Equation 3.4.2, where Nw is the number of weights, I is the number of inputs, $H1$ and $H2$ the number of neurons in the first and second hidden layers respectively and O is the number of outputs.

$$Nw = (I + 1) \times H1 + (H1 + 1) \times H2 + (H2 + 1) \times O \tag{3.4.2}$$

In light of this, it is obvious that any number of weight possibilities could produce an acceptable deep BPANN classifier. Subsequently, it could be said that this happens to be the premise behind the wide success of NN-based classifiers as a unique solution almost certainly exists.

In Figure 3.33, an important aspect which may go unnoticed is the output softmax layer. In non-classifier NNs, this layer is typically replaced with ReLU activation functions; however, for classifiers, a softmax, probabilistic, activation function is favoured, outputting a probability value between zero and one. In Figure 3.33, it is shown that each layer has a corresponding bias node, which serves to prevent zero-based training stagnation.

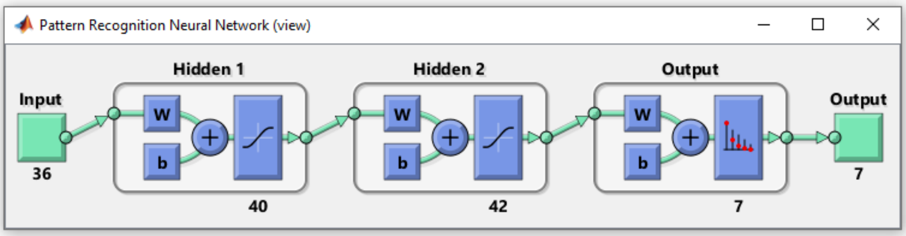


Figure 3.33: Double hidden-layer BPANN architecture.

The evaluation of the resultant deep BPANN classifier includes taking a look at the confusion matrix and ROC plot and comparing the findings with those discussed in Section 3.4.4.1. Firstly, the confusion matrix shown in Figure 3.34a follows a similar composition to those depicted in Figures 3.29a through 3.29d. That is diagonal dominance with the same class structure. Also, the resultant BPANN confusion matrix has characteristics similar to those extracted from the classification learner confusion matrices.

This includes the best classification class (i.e., class 5), as well as the coupled-confusion among classes 1 and 7. As elaborated on in previous paragraphs, rest produced steady, predictable sEMG data that was trivial to separate with even the most basic linear hyperplane. On the other hand, due to similarities in movement and the muscles used, classes 1 and 7 remained the largest contributors of misclassifications, echoing the findings extracted from the classification learner models.

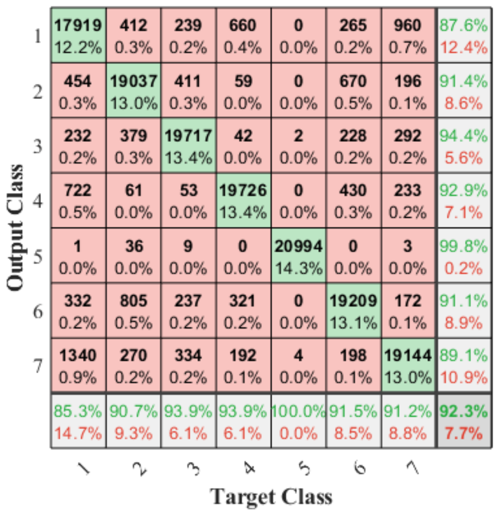
Not readily apparent in the confusion matrices shown earlier, a similar class pair producing considerable misclassification was that of classes 2 and 6, which represent cylindrical and spherical grasps. Scrutinizing the intersection of both classes results in the understanding that the class pair accounts for the second most confused pair, where the cause may be likened to that of class 1 and 7 confusion. In other words, cylindrical grasp is actuated by muscles similar to those of spherical grasp, which is coherent with the understanding that both prehensile motions are in effect power grasps.

Unlike the holdout validation applied to the classification learner models, the set ratios meant that the deep BPANN classifier was trained on 102900 samples while being validated and tested with 22050 samples. This resulted in confusion matrices with different total observations; however, in Figure 3.34a, the combined confusion matrix is shown.

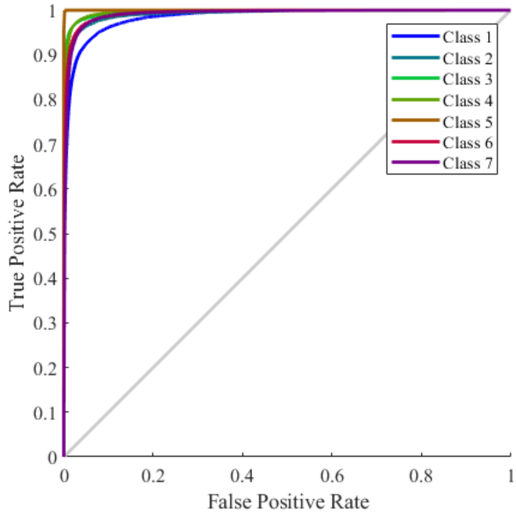
In comparison with the cubic SVM classifier, the deep BPANN classifier outperformed it in every class except class 5, where it trailed by a meagre 0.1%. Secondly, with an AUC of close to unity, the ROC plot shown in Figure 3.34b, illustrates the deep BPANN classifiers ability to discern between the various classes adequately. Moreover, the ordering of the classes, from worst classified to best, starts with class 1, followed by class 7, 2, 6, 3, 4, and this concludes with class 5. The order of the classes reiterates the conclusions drawn with regards to the confused class pairs, i.e. the four worst classified classes.

The ROC plot shown in Figure 3.34b, is what one imagines when envisioning a ROC plot since it is unaltered in contrast to Figures 3.29a and 3.29d. Suffice to say, no optimum threshold was applied; hence a smooth and symmetrical about the $x = y$-axis curve is depicted for each class. The symmetry is a direct result of the overlap created by the TN and FN curves and may be thought of as the intersection between two events in a Venn diagram. Due to the presence of an intersection between the two classes, they cannot be considered as mutually exclusive. In addition, the smooth nature of the curve is a direct reflection of the multitude of thresholds tested between zero and one, which lowers discretisation errors and grants access to an informed trade-off between specificity and sensitivity. As a point of clarification, choosing a different threshold will not influence the confusion matrix in any way, but rather is tailored to the foremost purpose of the classifier.

Figure 3.34b clearly shows a noticeable AUC improvement over a wide range of thresholds when compared to the SVM classifiers ROC plot developed earlier. This then justifies the potential and feasibility of using NN-based classifiers. Evidence in support of this may be found when investigating the following two studies [77]. Other non-visual performance measures include F1 score, MCC and observation speed, which were calcu-



(a) Confusion matrix.



(b) ROC plot.

Figure 3.34: Proposed deep BPANN classifier performance.

lated to be 0.9295, 0.9239 and 270000, respectively. Thus, through the comparison of performance parameters, it is apparent that the BPANN GR classifier is superior to previous classifiers expressed in this dissertation and correlates with literature while showing the most capacity for success.

Next, the deep BPANN GR classifier output and its implication in a real-time RPH system environment are expanded. To do this, a framework needs to be established around what the output is and how to decode it into something meaningful. Starting with the classifier's output, which is shown in Figure 3.35, it is easy to get lost at first glance. However, before defining Figure 3.35, it is important to note that it is based in a Simulink environment, under real-time conditions where the input is the same participant sequential training, validation and testing data previously used. In addition, the discrete time-step-size is in-line with the sliding window duration of 80ms; thus, a total running time of 11760s is expected.

In simple terms, Figure 3.35 shows the softmax-based probability level associated with each of the outputs for any given input. For example, consider the output bounded within the region 1344s to 1680s. This region reflects the output of the deep BPANN GR classifier with muscle features, corresponding to rest. As observed the probability level is highest for class 5, which accurately indicates the prediction of rest, whilst the probability level for the other classes proportionally remains negligible. The minimal fluctuation present from the other classes is interpreted as first-rate class separability and was discussed in Section 3.4.4.1 in relation with the confusion matrices.

Using this same method of analysis, the confusion present between classes 1 and 7 may be viewed under a transient lens. An example of which may be seen contained within the region 7056s to 7392s, i.e. the output of the third participant. Bearing in mind the visual cues for successful and unsuccessful classifications, a basic criterion may be formulated to distinguish good performing classes from poor performing classes, namely, with Figure 3.35 as a reference, the more chaotic the plot, the worse the performance. This criterion is relative and therefore depends on the application and impact misclassifications may have on the overall system.

Furthermore, an important outcome emanating from the analysis of Figure 3.35 was the visually selected threshold. Set at a probability level of 0.4, the threshold could have been more precisely estimated using a least-squares approach, maximising the distance between the threshold, highest class and second-highest class over the entire dataset. This approach was not implemented as later post-processing steps deemed it unnecessary to do so. The principle behind the chosen threshold, in summary, ensures that for a given input, the output reflects the designated class whose probability level is above 0.4.

In the case where multiple classes exceed confidence levels of 0.4 for a given input, the classification is deemed a confusion, and a do-nothing action is taken which is herewith referred to as class 8 (no-class). Likewise, a similar stance is taken when no class exceeds a probability level of 0.4.

Noting the information presented in Figure 3.35, it is obvious that a form of filtering will reduce the rapid fluctuation tendency of the output and thus make the output more stable and determinant. Similar to the effect produced by low-pass filters, a moving average performed signal smoothing and was therefore subsequently used as a method to alleviate some of the fluctuations categorised as misclassifications. Despite this, the method of signal smoothing, institutes system lag equivalent to the buffer size. Therefore, the buffer size was kept to a minimal 80ms at the expense of not entirely removing high-frequency fluctuations and is discussed in Section 3.4.5.

The disturbance rejection properties of the BPANN is inherent through the type and quality of data used to train the BPANN classifier. Thus, by the same token, any individual data outlier recorded or mislabelled input, when incorporated in small quantities, serves to strengthen the disturbance rejection, robustness and generalising capabilities of the classifier.

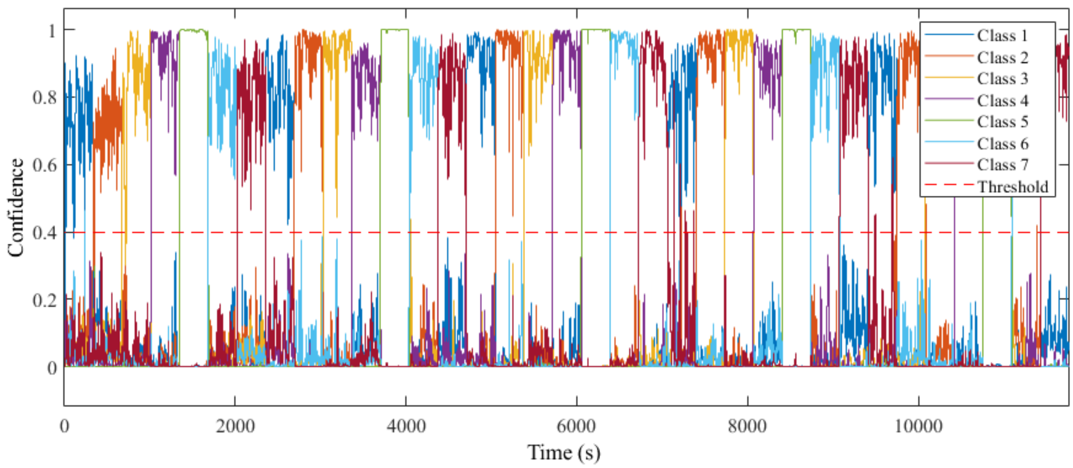


Figure 3.35: Proposed deep BPANN class confidence output.

3.4.5 Post-processed classifications

Processing of the BPANN classifiers output via the threshold method and moving average leads to the one-dimensional categorical output shown in Figure 3.36. From this, a conditional statement may be utilised to enforce different prehensile movements, using changing joint set-points. Seeing as Figure 3.36 reflects the output before post-processing, it is necessary to delve into the issues apparent and commend the progress made thus far. To do this, Figure 3.36 is analysed, starting with the structure which resembles a staircase; which is simply a product of the order in which the data was given to the classifier. Moreover, the repetition of the staircase-like structure occurs for each participant. At this stage, it is crucial to note that had the BPANN classifiers output not been a softmax layer, and additional layers were added to the NN (to deepen the network) the pattern in which the inputs were being added would have been learnt. Therefore, rendering the classifier ineffective in cases where the classes were in a different order.

Seeing as this is not the case with the developed BPANN classifier, the generalising capabilities must therefore be appreciated. As seen in Figure 3.36, there remain spikes in the plot. These anomalies transpire as a result of the classifiers misclassification of classes, but occurs far less frequently, with a smaller duration, than if no threshold or moving average was used. Also, as a by-product, the implementation of the do-nothing class 8 renders the effects of misclassified instances harmless to the structural integrity of the RPH system. However, it adversely impacts its functionality by returning to a rest RPH position.

This, as one could imagine, poses a problem during the action of grasping. As a remedy, the do-nothing state should instead mimic the last known validated grasp, yet this would be too computationally expensive to keep track off. Also, this, there remains class pair confusion, an example of which may be seen at 4032s. The moving average resultant lag, illustrated in Figure 3.36 through the classification trailing the target appears to be non-uniform. Suggesting that perhaps the combination of buffer size and threshold leads to sluggish transitions between certain classes.

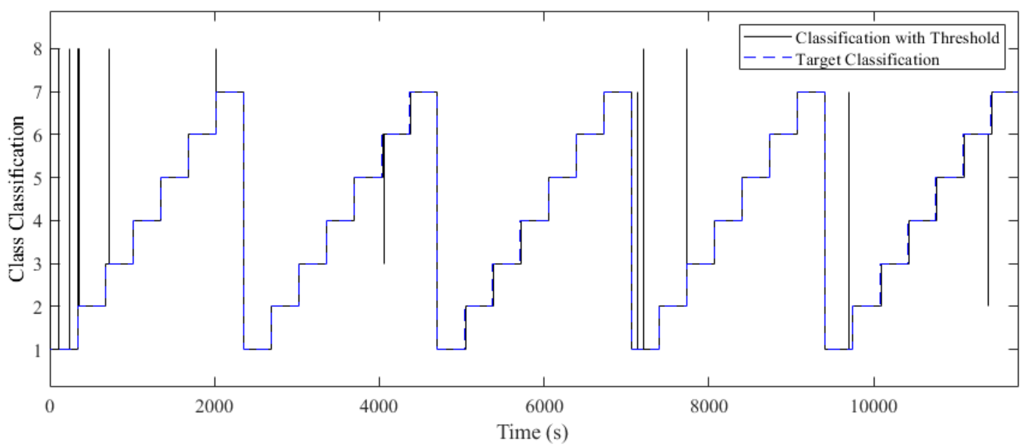


Figure 3.36: Threshold-based class interpretation of BPANN classifier output.

Following on the issues brought to attention with regards to the output classifications, the post-processing took a minimalistic approach. Though being simplistic in implementation, the post-processing procedure had a profound impact in achieving a final GR model classification accuracy of 98.83%. Although having achieved this accuracy, it should be noted (using Figure 3.37 as a reference) that 100% accuracy is reflective of the model, since the output is not misclassified but rather delayed. This important distinction may physically be interpreted as the RPH correctly performing the envisaged grasps, but being slightly delayed in doing so.

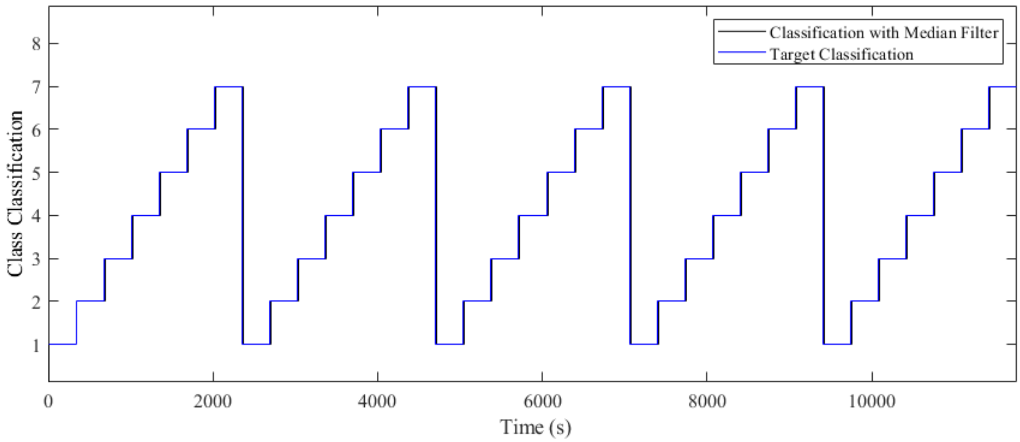


Figure 3.37: Median filtering of threshold-based class interpretation.

Quantification of the lag or delay shows the output classification trailing the target classification in some instances by at most 0.35s. After considering the average movement reaction time resulting from visual stimuli, which amounts to 300ms [78], it is concluded that the introduction of lag has negligible affects on the overall function of the RPH. A rather important factor inherently minimised through the post-processing of the classifier's output is that of jerk. That is because the post-processing method implemented in the Simulink environment, which involved the use of a median filter with a window length of 251 samples, flattened any spikes among classes.

Figure 3.37 serves as an image of the final output from the sEMG GR model and fed into latter processes which delegate the movement of the RPH. In parallel with reducing the number of misclassifications, the median filter helps to attenuate noise components from corrupting the output.

3.4.6 Simulink model

The Simulink GR model, at its core, consists of a deep BPANN classifier, which was generated by using the gensim command on the trained NN object. In doing so, the fundamental structure of a two-layer BPANN, like that shown in Figure 3.4, was created in block form and the relevant weights and biases integrated as constants with the commands getwb and setwb. After exporting the model, the linkages joining the various layers were realised. The details surrounding the inner workings of the deep BPANN classifier were attended to at a later stage.

For now, however, the mask represents the BPANN classifier and is depicted in Figure 3.38 as grasp pattern recognition NN. Shifting focus to the input of the BPANN, a from workspace block is used to import the extracted feature dataset. Instead of acquiring the extracted features through an off-line process, the mathematical computations and data manipulations owing to the extracted features may have been computed on-line. Thus, the only necessary input would have been the raw sEMG data. This was not within the scope of this dissertation and was considered a small extension that bared no influence on the results other than completing the data life-cycle.

Syncing the Simulink solver step-size with that of the inherited off-line data presented a pitfall in the configuration that was foremost dealt with. Following this topic, the solver was set to automatic solver selection with a variable step-size and a maximum step-size of 0.08s. Also, the simulation span was set to 11760s and the initialisation function, found in the model properties, called a script which loaded the extracted feature inputs located in a specific file directory. Apart from being the starting point of the sEMG GR model, the extracted feature data also served as the input for the sEMG-force model discussed in Chapter 3.

Moving on, the softmax output of the BPANN classifier is separated into various streams, where each stream chronologically represents an individual grasp probability. The separation was accomplished by demuxing the data vector. The separated signals were smoothened by buffering 100 samples at a time and computing the mean, hence effectively carrying out a moving average. Using these probability signals as inputs, an embedded MATLAB script decoded and interpreted the output class using a threshold and conditional IF-ELSE statements. As such, the output formerly shown in Figure 3.36, was supplied to the median filter for post-processing. Which ultimately yielded the sEMG GR model output denoted in Figure 3.38.

Regarding the hierarchy of the RPH Simulink model, the sEMG GR and sEMG-force models both work in parallel with one another, and relay set-points to the FLC. Though it should be noted that the set-points from both these models may contradict one another. Therefore resulting in a hierarchy or a cascade-like behaviour to be predefined as part of the foundation for the FLC.

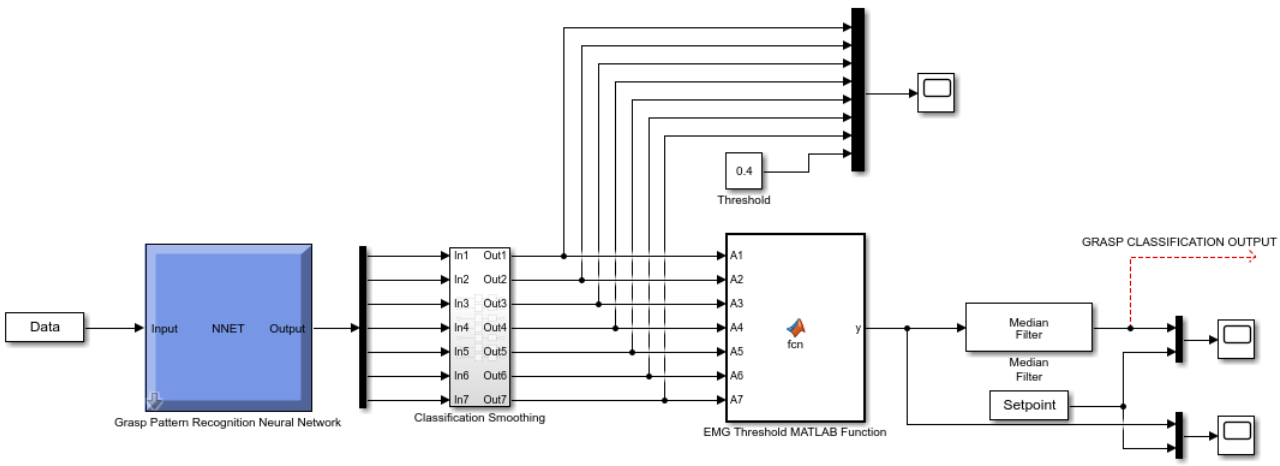
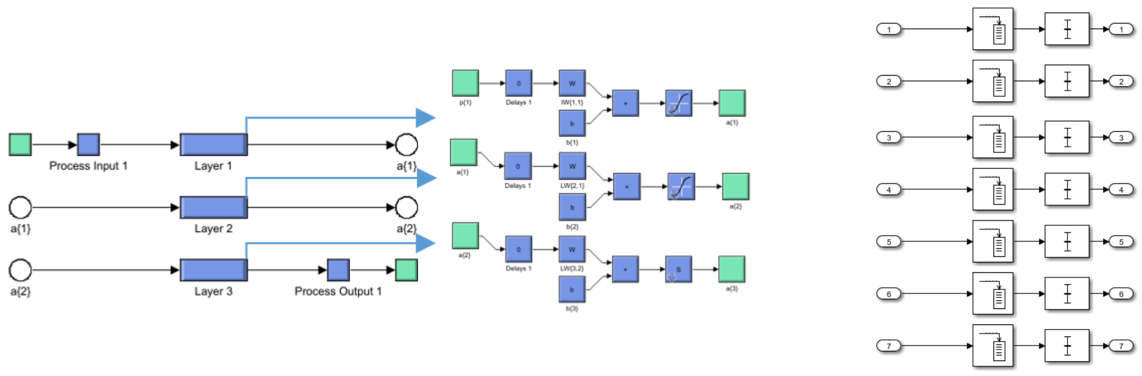


Figure 3.38: Simulink model of GR classifier.

Under the mask of the BPANN classifier block depicted in Figure 3.8b exists the detailed inner workings shown in Figure 3.39a. Figure 3.39a presents three layers, where the top two are hidden layers and the third the output layer. Each layer is linked via input and output ports which should not be likened with synapses, as this is internal to each layer mask. Removing the mask from each layer exposes the delay, bias, weight and activation nodes. A key point observed in Figure 3.39a is the fact the classifier does not contain any delays. Though not typical of classifiers to require delays, delays may be added if the classification depends on past events. In relation to sEMG GR, the trend anticipated, as it relates to the RPH grasp occurrences, is stochastic and should remain as such; therefore, it is strictly advised not to implement any delays. In later sections, the use of delays in NNs will be explained.



(a) BPANN classifier

(b) Moving avergare smoothing

Figure 3.39: BPANN classifier and classification smoothing sub-systems

Also, in Figure 3.39a, the weight blocks (though resembling constant blocks) are in actuality a complex web of constant linkages between the neurons in each layer. The bias blocks are far simpler and are constant vectors. Prominent only within the top two layers are a tan-sig activation function equivalent to the hyperbolic tangent. Consequently, tan-sig activation functions are almost always used in conjunction with classifiers as opposed to a log-sig activation function. At the crux, this is due to two factors, stability over a wider range of inputs and non-positive definite applications.

The third layer incorporates the probability distribution softmax activation function. The Simulink diagram shown in Figure 3.39b is the classification smoothing sub-system and as previously described is a buffering block in series with a mean block.

3.5 Chapter Summary

In this Chapter, several grasp recognition methods were explored in order to classify sEMG data obtained from a group of volunteers. This was necessitated through the use of both hardware and software pattern recognition techniques. Grasp recognition involved acquiring data from the prehensile movements of the human hand. The characteristics of the acquired data were first analysed using Python and Matlab to eliminate unwanted signals from being further processed. This was accomplished by manipulating frequency-based characteristics to filter unwanted signals before full-wave rectification. Using a combination of mathematical models from the Hudgins feature set and Matlab, the features of the rectified signals were extracted for better class separability. Using iterative training, the extracted features were mapped into different classes. This was done by the training of different algorithms to learn the underlying patterns of a labelled output of the extracted features. The results from the BPANN classifier were the most effective, and this classifier was used to develop an effective GR classifier model using Simulink.

Chapter 4

Artificial Neural Network Grasp Force Estimation Using Multi-Channel Surface Electromyographic Signals

Towards Bio-Inspired Grip Compensation of a Robotic Prosthetic Hand

Estimating grasp force from sEMG is an active research topic, which is partially aimed at improving the functional and dexterity capabilities of RPHs. By estimating the grasp force, not only does it provide the prospective user with a greater range of control during grasping, but when combined with haptic feedback devices, may provide a sense of feeling and reduce the learning time required [79].

The sEMG-force relationship lacks consensus between researchers due to the influence of various factors such as the indirect nature of the non-invasive sEMG-based approach used to deduce the muscle activity. To infer this relationship requires complex accountability of these factors, which is still the subject of much research. Some intrinsic conditions found to influence the relationship between grasp force and sEMG include contributions from non-contractile structures, joint position and posture, and fatigue-related hysteresis effects. Conversely, it is suggested that extrinsic conditions such as temperature and sensor placement over time may also account for changes in the registered Motor Unit Action Potentials (MUAPs).

This chapter builds on the isometric contractions (grasps) presented in Chapter 3 and proposes a feasible regression model that maps input sEMG signals to corresponding force outputs. Many of the processes and concepts concerning sEMG are omitted from this chapter as it has previously been addressed. Though much research has gone into determining the relationship between grasp force and sEMG, there seems to be two schools of thought split on whether the relationship is linear or curve-linear. Also, the degree of non-linearity is relatively unexplored. Alternative to a parametric mathematical-based approach, this dissertation considers several machine learning-based solutions. Thereby broadening the non-linear solution space and accounting for a possible non-linear sEMG-force relationship. The concepts developed may be extended to applications, such as biokinetic rehabilitation, exoskeletons, virtual reality and ergonomics.

4.1 Theory and Literature Review

The forearm muscles are primarily responsible for the manipulation of one's fingers and wrist. By using a combination of the 20 available forearm muscles, activities of daily living such as writing, typing, grasping, eating, cleaning, lifting of objects, etc., are made possible. These activities all require different levels of muscle contraction, which, when analysed may be applied to RPH grip compensation.

To get a representation of the force exerted during muscle contraction, researchers make use of a variety of external pressure and torque sensors, located at the joint or end-point contact surface. These sensors are often bulky and restrictive, not allowing for a conducive range of motion and ultimately leading to erroneous conclusions. Furthermore, little consideration is given to the influence these sensors may have on the wrist and joint positions. Taking a less restrictive and indirect approach, Armstrong *et al.*, [80] estimated grasp force using sEMG signals and linear regression. This marked the start of extensive research in experimental modelling of the relationship between sEMG amplitude and force. Other models rooted in parametric data are based on Hill's muscle model [81], which requires extensive participant-specific measurements and knowledge on the dynamic and kinematic modelling of the joints.

Non-parametric regression techniques such as machine learning have produced promising results without the need for a priori knowledge. The approach for determining the relationship between sEMG and force, given static isometric non-fatiguing contractions, is similar to that of GR and is depicted in Figure 4.1. Here, the process follows data acquisition, preprocessing, feature extraction and regression in both sEMG and force aspects.

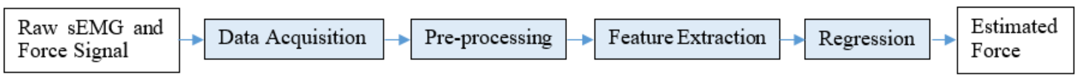


Figure 4.1: Force estimation from sEMG signals for use in an RPH.

4.1.1 sEMG-force data acquisition

Acquisition of sEMG and force data is a preliminary and important step in establishing a non-parametric grasp force estimating model. Though both data streams are inherently coupled, they are often acquired and processed separately. To effectively acquire sEMG-force data, one first needs to understand the nature of the data. Thus, in this section, the origin of force and how it relates to sEMG is foremost considered. Furthermore, the two primary catalysts for force generation, namely MU recruitment and firing rate, are introduced.

Data acquisition systems found in literature, including the types of sensors and setup, are given with emphasis placed on force data acquisition (sEMG data acquisition discussed in Section 3.1.1). Nevertheless, theory related to sEMG data acquisition is included where new or differing concepts are introduced. The section concludes with the consideration of some intrinsic and extrinsic factors which may affect data acquisition and the subsequent sEMG-force model.

4.1.1.1 Origin of force generation

The dynamics of skeletal muscles has intrigued many scientists over the years, resulting in a significant amount of research. Arguably, the most impactful of these was given by Duchenne [82], where the foundation of electrical conduction in the nervous system was established. A comprehensive sEMG-force model hinges on the consideration of the electrical conduction in the nervous system through examination of the anatomical structures that relate the MU action potentials to various stages of muscle contraction.

Illustrated in Figure 4.2, by thinking of the desired action, for example, closing one's hand neural drive is increased and neurons in the brain fire, creating electrical impulses in the process. These impulses then propagate through the central nervous system (CNS) in the spine and collate in motor pools. Here α and γ motoneurons are present, where α motoneurons form the beginning of MUs. Specifically, the stimulation of α motoneurons is responsible for the voluntary contraction of skeletal muscles (extrafusal muscle fibres); while the stimulation of γ motoneurons results in an involuntary response to external forces via intrafusal muscle fibres [83]. In relation to the hand, motor pools are predicted to be large, due to the low innervation ratio. In other words, for the number of muscle fibres innervated by a single motor neuron [84]; a muscle fibre may only be innervated by a single motor neuron.

Following the stimulation of α motoneurons, a neuro-transmitter is released, which travels along motor neuron pathways to junctions on the fascicles (cluster of skeletal muscle fibres). At the neuromuscular junctions (axon), an ionic chemical reaction takes place which depolarises the muscles fibres and gives rise to Action Potentials (APs). Which then extend down the terminal branches that innervate the muscles fibres and terminate at the tendons. These APs then result in the contraction of sarcomeres in the muscle fibres. The contraction occurs bi-axially via the process of sliding filaments (actin and myosin overlapping and interlocking). It is this contraction in the skeletal muscle, along with the interactions of various tendons, bones and joints that results in the closing of one's hand. When extended to grasping, the same principles apply; however, the amount of contraction varies to accommodate the required force.

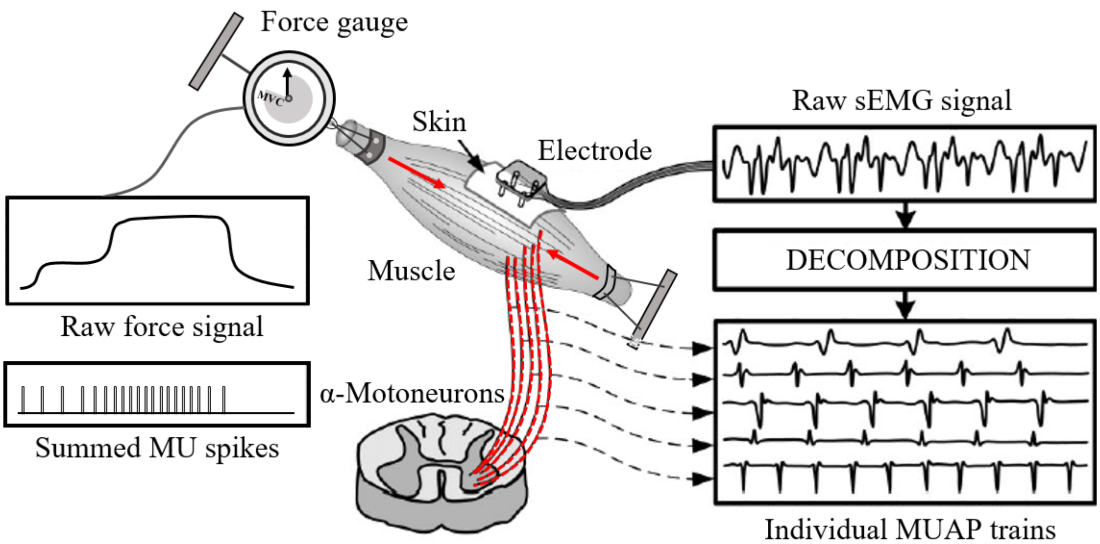


Figure 4.2: A schematic representation of the origin of force in skeletal muscles [83]; Adapted with permission.

In Figure 4.2, it is shown that the firing rate expressed as the summed MU spikes, directly impacts the generated force. Hence, the idea that recording the summation of MUAPs through sEMG (multi-channel sEMG) may give insight into the cross-correlation of MUs and the generated force, producing a definitive relationship. The frequency of stimulation (firing rate) yields different muscle contraction states, such as muscle twitch, wave summation, unfused tetanus and fused tetanus. In addition, the strength of stimulation determines the proportion of MUs recruited, either synchronously or asynchronously. In this dissertation, the isometric contraction of the forearm muscles is expected to originate from fused tetanus stimulation and sub-maximal synchronous motor unit recruitment.

The number of MUs vary across different skeletal muscles. Small muscles found in the hand, have about 100 MUs, while larger muscles as those found in the leg, have in excess of 1000 MUs [85]. The MUs may be categorised into two main classes, namely slow-twitch and fast-twitch (type I and II), depending on the contraction velocity and fatigability of the muscle fibres. Interestingly, MUs may change from one type to another in cases of severe deconditioning or injury [86].

The muscles investigated in this dissertation are categorised, so that a general understanding of the acquired APs may be reached. The flexor digitorum superficialis is reported to consist of a larger proportion of slow-twitch fibres (about 60%), most likely to ensure that the intrinsic hand muscles flex the metacarpal (MP) joint before being followed up with PIP joints when grasping an object [87]. The most predominant type of muscle fibre found in the flexor carpi ulnaris has not yet been established. However, the examination of monkeys reveals that different fibre types dominate on either side of the central tendon (that is, type II fibres dominate the ulna head while the humeral head is dominated by type I fibres) [88]. The same study also found that there was an even dispersion of type I and type II fibres in the extensor carpi radialis longus. This outcome was shared for the extensor digitorum [89]. No definitive research indicated the dispersion of muscles fibres in the flexor and extensor pollicis longus. However, it may be assumed that it is even and heterogeneous since the muscles do not act in an anti-gravity muscle capacity [90].

4.1.1.2 Role of MU recruitment and firing rate in force generation

It is well-known that the recruitment and firing rate of MUs are highly correlated with the output force of the muscle they innervate. For this reason, it is worth reviewing within the context of this chapter.

The recruitment of MUs in principle centres around the size of the MU, directly linked to the number of active MUs. This is known as the size principle and was suggested by Henneman *et al.*, [91]. In short, the size principle states that MUs with small cell bodies are joined to muscle fibres which produce a slower response, lower force output and are more resistant to fatigue (type I). Conversely, MUs with large cell bodies are joined to muscle fibres which produce a faster response, higher force output and are prone to fatigue (type IIa and IIb). Also, the principle states that the recruitment of MUs progress from small to large cell bodies (orderly or task-specific recruitment). This occurs as the synaptic excitation input to the motor nucleus increases (as more force is required) [92] and is due to the threshold associated with the surface area of each cell membrane [93]. To illustrate the principle, consider both precision and power grasp. For a precision grasp (threshold/submaximal stimulus), the AP is only sufficient to overcome the threshold required to activate weaker and slower muscles. While for a power grasp (submaximal/maximal stimulus), the AP is sufficient to activate stronger and faster muscles. In other words, a higher AP leads to the recruitment of larger MUs, which in turn results in greater force output.

Another factor that affects the output force is the firing rate of the MUs. As shown in Figure 4.2, the firing rate of MUs may be described as the frequency of AP spikes. In work conducted by De Luca *et al.*, [94], it is reported that MU firing rate is inversely proportional to the recruitment threshold. That is, MUs with small cell bodies have a firing rate higher than that of large cell bodies, which, when plotted resembles an onion skin [85]. This organisation pattern is currently debated, with other research suggesting the opposite (inverse onion skin) [95]. On the topic of force generation, it is generally found that an increase in the firing rate produces an increase in the output force regardless of the organisation pattern. However, the synchronisation of MUAPs (constructive and destructive spatio-temporal summation) may introduce complexities.

Additionally, it is proposed that the firing rate acts as a buffer, smoothing muscle movement during the recruitment of the MUs [94]. Previous research showed that MU recruitment often does not extend beyond 50% of the MVC, after which the firing rate strongly dictates the increase of contraction. In conclusion, both the MU recruitment and firing rate (known as rate coding) are presumed to increase with increasing force; thus by measuring these components (using indirect means such as sEMG), a relationship may be obtained.

4.1.1.3 Systems used for force data acquisition

Considerable attention is often placed on the sEMG data acquisition system, while the force data acquisition system is often overlooked. This presents a challenge in establishing the most suitable data acquisition system for recording grasp force. In light of this, the types of force sensors and applications are presented.

Force sensing in biomedical applications generally consists of three main types, namely force platforms, force transducers and pressure sensors [96]. However, in this dissertation, force platforms are not considered as they are often used in the gait analysis of lower limbs. Load cells are a type of force transducers used in the analysis of force and moments produced by skeletal muscles during various types of static or dynamic contractions. They utilise either strain gauge (thin-film, foil or semiconductor), piezoelectric, hydraulic, electromagnetic or capacitive measurement elements to yield an electrical signal representative of force (that is, tension or compression), pressure or torque [97]. Strain gauge load cells are the most widely used load cells and come in various configurations and orientations depending on thermal requirements and loading direction. Arjunan *et al.*, [98] used a high-performance S-type strain gauge load cell rated at 100N maximum output, to record the force produced by the biceps brachii muscle during isometric contractions at 50%, 75% and 100% MVCs. The results of the study proved that the force data acquisition system sufficiently captured the forces stemming from a bicep curl, but was not suitable for registering the grasp force. A dynamometer can be used to address this.

An example of its use is given by Yokoyama *et al.*, [99], where sEMG signals at five different force levels, that is 0, 50, 100, 150, and 200N, were investigated. Here, the dynamometer and participant's arm were in a supported horizontal state during the isometric contractions to mitigate the effects of gravity. The approach provided a good indication of the grasp force experienced but is limited in the types of grasps in which it is applicable. This is because the layout of a dynamometer is conducive to measuring the force produced by power grasps.

A somewhat less accurate type of force sensor, such as a resistive polymer film sensor is needed to accommodate for non-power grasps. These sensors are of two types, namely Force Sensitive Resistors (FSRs) and Flexiforce sensors [100]. These sensors have the same working principle where a drop in electrical resistance signifies an increase in pressure. Though, most often used in Force Myography (FMG), these sensors have also shown success in grasp force recording [101, 102]. A recent study carried out by Leone *et al.*, [103], used five FSRs, mounted on the pads and palmar side of the distal phalanges to record force data for a tip pinch and spherical grasp. The FSRs were mounted on a glove that was worn by the participants during testing and three levels of contraction, namely 30%, 60% and 90% MVC (low, medium and high force), were examined and classified. The results showed that the non-linear logistic regression (NLR) and LDA classifiers were able to classify the grasp forces with an accuracy of 98.8% and 98.7% for the spherical grasp, and 96% and 97.6% for the tip pinch.

FSRs are limited by the range of sensitivity (typically in the range of 0.2N to 40N) and hysteresis, where higher forces are associated with significant hysteretic effects [104]. Unlike Leone *et al.*, Hashemi [105] incorporated a coupling disk which served to distribute the load experienced by the FSR uniformly. Also, Ye *et al.*, [106] placed the tail end of the FSRs out of the contact region to avoid interference. In conclusion, several force data acquisition approaches are shown to be successful; however, the most practical solution for acquiring grasp force data, albeit less accurate, involves FSRs. Table 4.1 presents some of the force data acquisition systems and relevant details found in the literature.

Table 4.1: A summary of the types of force data acquisition systems used in grasp force estimation.

Sensor	Force Range (N)		Grasp	No. of Participants	Sampling Speed (Hz)	Reference
	Sensor	*Participant*				
6-Axis Load Cell (NANO17)	0 to 70	2,5 to 10	Pick and Lift	12	10240 (Sampled to 2048)	Martinez *et al.*, [107]
6-Axis Load Cell (ATI MINI45)	0 to 120	0 to 40	Power	10	500	Kim *et al.*, [108]
6-Axis Load Cell (NANO43)	0 to 36	0 to <25	Power	2	1000	Luo *et al.*, [109]
6-Axis Load Cell	-	0 to 90	Various Finger Pinches	3	100	Ma *et al.*, [110]
Adjustable Handgrip Dynamometer	0 to >350	0% to 80% MVC	Force-Varying and Static	20	1000	Wang *et al.*, [111]

Continued on next page

Table 4.1 – *Continued from previous page*

Sensor	Force Range (N)		Grasp	No. of Participants	Sampling Speed (Hz)	Reference
	Sensor	*Participant*				
Hand Dynamometer and 3D Push-Pull Sensor	0 to 30	0 to <30 and 0 to <20	Power and Random Movement	6	1000	Wu *et al.*, [112]
Hand Dynamometer and Pinch Meter	0 to >150	0% to 60% MVC	Power, Precision and Lateral	3	1600	Yamanoi *et al.*, [113]
Hand Dynamometer	0 to 600	0 to 50	Grip Force Profiles	10	500	Kamavuako *et al.*, [114]
FSR (Interlink 400)	0.2 to 20	-	Grasping of Various Objects	-	2000	Barone *et al.*, [115]
FSR (Interlink 402)	0.2 to 20	0% to 100% MVC	Spherical and Tip	31	1000 (Sampled to 100)	Leone *et al.*, [116]
Eeontex Piezoelectric	0.05 to 1000	-	Grasping of Various Objects	2	500	Seedat *et al.*, [117]

4.1.1.4 Aspects to consider during sEMG-force data acquisition

There are several intrinsic (physiological) and extrinsic (measurement) factors to consider during sEMG-force data acquisition. These factors may be prominent in the recorded sEMG signal and may affect the capacity to generate force. Some notable physiological factors include MU recruitment and firing rate, muscle fibre properties (type, length, conduction velocity, cross-sectional area and depth), muscle state (blood flow and hydration), volume conduction, MU potential summation, fatigue and age [117, 118]. On the other hand, measurement factors include the type of contraction (dynamic or static), skin-electrode impedance, location and number of sensors, temperature, time resolution (sampling rate), bandwidth, selectivity and representativeness [119, 120].

While it is suggested that the physiological properties of muscle fibres, previously stated, are correlated with the generated force, Lieber *et al.*, [121] indicated that it is more complicated in the case of the forearm muscles since the effects of the moment arm need to be considered. Lieber et al., [121] justified this fact by analysing the length and pennation angle of several forearm muscle fibres in cadavers. It should be noted that the muscle state factors mentioned are presumed a consequence of the historical generation of force and are heightened by sustained fatiguing contractions [122].

Volume conduction is a highly coupled and influential factor in force generation and is defined as the muscle equivalent of the Poisson effect. During contraction, the "barrelling" of muscle causes spatial changes between the sEMG electrodes and the muscle belly, which have a low-pass filter effect on the perceived sEMG signals [123]. MUAP trains are typically bi/tri-phasic signals (i.e., polyphasic at endplates and in cases of neurodegenerative diseases [124]), which have constructive and destructive segments when summated. This effect is typically disregarded but may be directly related to the stochastic nature of the sEMG signal and plays a role in the distinguished sEMG-force relationship.

A widely researched topic is that of fatigue and its implication in force and sEMG signals. Fengjun [125] observed a shift towards a lower ratio of mean frequency (MNF) and the force generated as a result of fatigue and was thus able to conduct onset detection of muscle fatigue. Furthermore, fatigue skews the sEMG-force relationship. The effect of age is highlighted in work by Sahebjada [126], in which it was most notably found that co-activation of muscles increased with age since individual muscle force degraded with age. In other words, as age increases, so too does the level of coupling between the muscles. Interestingly, co-activation does not only extend to age but also muscle injury. Particularly in this chapter, the effects of the type of grasp on the sEMG-force relationship is investigated. Most studies do not acknowledge this; however, recently, Ma *et al.*, analysed the force produced and classified four different pinches. The results showed excellent predicting capabilities (especially for less involved pinches), but the study remains limited in the grasps investigated. The variability in sEMG signals depends on the location and number of sEMG sensors and has led to the adoption of high-density (HD) sensor arrays. These arrays give insight into causative factors, and their use negates the need for repositioning of sEMG sensors between individuals. Selectivity and representativeness primarily deal with two aspects of the muscles and contact regions. Firstly, the level of cross-talk or coupling and then the proportion of data representing the cross-talk or coupling. In conclusion, several highly influential factors which contribute to the observed force need to be considered.

4.1.2 Pre-processing

The pre-processing of sEMG data is detailed in Section 3.1.2, and will therefore not be reviewed in this section. Instead, the pre-processing of force data will be covered. Additionally, the pre-processing of sEMG data is not required for certain features (as noted in the following section). In the literature, attention is often placed on the pre-processing of sEMG data, with little reference to force data. As such, one is left to assume that force data is in a raw form. Depending on the sampling speed, this may be an issue, since noise (particularly at low forces or low SNR), could corrupt the data.

Ye *et al.*, [106] avoided this by using a low-pass filter with a cutoff frequency of 50Hz, which smoothed the data and was especially useful in ensuring rest periods remained bias-free. Similarly, Kamavuako *et al.*, [114] also used a low-pass filter with a cutoff frequency of 20Hz. The low cut-off frequencies are well-founded because of the speed at which changes in grasps occur; however, these frequencies require revision in instances of undefined contraction patterns involving fast-twitch muscles. In addition, the effects of using a low-pass filter need to be tested since it introduces an increase in the electro-mechanical delay, which will affect the structure of the chosen regression multilayer (ML) approach. Apart from filtering, the force data may also sometimes be subjected to upsampling or downsampling, through repetition, interpolation or dropout. Whether using either of these methods, it needs to be followed up with a low-pass filter to prevent signal aliasing.

4.1.3 Feature extraction

The feature extraction discussed in this section relates to that of sEMG and not force. This is because, extracting intermediary characteristics from the force signal like, a hand force distribution index, would introduce unnecessary links which at a later stage would have to be reincorporated. Also, it is more appropriate in analyses considering the spatial activation of muscles [127]. Feature extraction may be extended beyond the application of classification and beneficial in most machine learning applications. The reason being that feature extraction performs characteristic decomposition in the chosen domain, giving a greater level of detail and reducing the complexity of the machine learning model needed. Most of the advantages and methods for optimally selecting features were discussed in Section 3.1.3; however, it should be noted that feature extraction may be less conducive to real-time implementation due to online computation requirements. To mitigate the online computational cost, Xu *et al.*, [128] investigated the use of CNN, Long Short-Term Memory (LSTM) and a combination of the two (C-LSTM) for real-time force estimation using sEMG data. The promising results achieved by the LSTM and C-LSTM networks suggested that these networks were capable of automatic extraction of features. Two other important findings in the study seem to suggest that the LSTM and C-LSTM networks are multi-participant suited and that the inclusion of varied force patterns are required to improve environment bandwidth.

Though these networks have gained much attention in terms of automatic feature extraction, particularly in vision recognition, it remains that they are more computationally heavy during training. Also, they do not provide any perceived improvement in performance over traditional MLP networks with manual feature extraction in many benchmark problems [129]. Furthermore, these networks are subject to forming artificial correlations which present poor generalisation properties. This is one of the reasons that manual feature extraction is often adopted when formulating the sEMG-force relationship. Some notable and recent studies are shown in Table 4.2, along with the corresponding features. From Table 4.2, it is seen that one of the most used time-domain features is RMS. This feature is amplitude-based and is suggested to be correlated with MU recruitment (a known precursor for force generation) [130].

The use of frequency-based features is scarce, but of them, the most commonly used is MNF. This feature has been noted to be particularly useful in detecting muscle fatigue [131]. In a study by Phinyomark *et al.*, [132], it was noted that there are conflicting opinions on whether MNF and median frequency (MDF) are effective features for predicting muscle force. Three cases were noted by Phinyomark *et al.*, [132]. The first case was a weak correlation, where the force did not affect MNF and MDF. The second observation was a direct correlation, where an increase in force increased MNF and MDF. Lastly, the third case was an indirect correlation, where an increase in force resulted in a decrease in MNF and MDF. Phinyomark's results suggested a weak correlation due to subject variability, stating that the cases observed among the participants differed. However, in conclusion, Phinyomark et al. suggested that applying MNF and MDF feature extraction to the raw data should see an increase in muscle force increasing MNF and MDF, respectively. The suggested relationship may be a simplification, according to Farina *et al.*, [133], since volume conduction is often not taken into account. However, in this dissertation, it is hypothesised that MNF (distinguished through spectral analysis) should increase as the MU firing rate and force generated increases. This is because as the force increases the MU firing rate is responsible for enacting an increase in muscle force; an increase in MNF is expected regardless of the effects of volume conduction and the attenuation due to physiological structures.

Table 4.2: Comparison of classifiers used in sEMG GR.

Feature	Filter (Hz)*	Window Length (ms)†	Performance	Reference
SSE, GDR, CSE and RMS (I)	4^{th} Order Butterworth Filter (100-3000)	200	R^2: 0.88 0.89	Kamavuako *et al.*, [134]
14 Features (I)	4^{th} Order Butterworth Filter (100-3000 I; 20-500 S)	200	R^2: >0.9	Mathiesen *et al.*, [135]
MAV	-	-	R^2: >0.99 and MSE (N): <2.6	Luo *et al.*, [109]
MAV, RMS, SD and WL	-	50	Accuracy (%): >95	Li *et al.*, [136]
RMS	4^{th} Order Filter (30-700)	500	R^2: 0.85 and RMSE (N): 45	Yokoyama *et al.*, [99]
MNF and MDF	Bandpass (20-450)	375	p-Value (ANOVA): 0.0005	Phinyomark *et al.*, [132]
NSM5, MDF, RMS, NRMS, WL and IIS	Bandpass (20-450)	1000	R^2: 0.32-0.96 and p-Value (ANOVA): 0.45-0.001	Arjunan *et al.*, [98]
RMS and Moment Ratios	-	50	NRMSE: 23	Gailey *et al.*, [137]
RMS	-	200	CC (%): >90 and RMSE (N): <10	Ma *et al.*, [110]
26 Features	Bandpass (10-350)	50	RMSE (N): 33	Ye *et al.*, [106]
MAV, VAR, SC and WAMP	Bandpass (10-500)	-	MAVE (N): 1.1, RMSE (N): 1.6 and CC (%): 97.8	Wu *et al.*, [112]
MAV, IEMG, RMS, WL, VAR and PS	-	-	Accuracy (%): 94.9	Prasad *et al.*, [138]
FD, CV, ARV, MNF and Borg	Bandpass (10-750)	-	p-Value (Dunn-Bonferroni): <0.001 (Excl. CV)	Beretta-Piccoli *et al.*, [139]
Ten Features	4^{th} Order Butterworth Filter (20-500)	60	R^2: 0.67 and AE (N): 2.52	Martinez *et al.*, [107]
13 Features	-	256	Accuracy (%): >74.41	Xu *et al.*, [140]
MAV, WL, RMS and FILT	-	400	R^2: >0.89 and NRMSE: <0.045	Baldacchino *et al.*, [141]

(SSE), (GDR), (CSE), (SD), (MDF), (NSM5), (NRMS), (IIS), (VAR), WAMP, (IEMG), (PS), (FD), (CV), (ARV) and (FILT)

Continued on next page

Table 4.2 – *Continued from previous page*

Feature	Filter (Hz)*	Window Length (ms)†	Performance	Reference

Excluding * PL interference notch filter and †window overlap length

With the above time and frequency-domain sEMG features, namely RMS and MNF, a relationship with force may be constructed. The mathematical description surrounding RMS has been given in Section 3.1.3, along with the concept of sliding windows. To extract frequency-domain features such as MNF, the sEMG signal in the time-domain is firstly transformed using a Fourier transform [142]. The transform yields power spectral characteristics, from which MNF can be calculated.

Shown in Equation 4.1.1, is the definition of MNF, which follows the sum of the product of frequency and power, denoted as f_n and P_n respectively, divided by the sum of power. Here, n refers to the frequency bin (which determines the frequency resolution) and N refers to the total number of bins present in the data contained within the window length.

$$MNF = \frac{\sum_{n=1}^{N} f_n P_n}{\sum_{n=1}^{N} P_n} \tag{4.1.1}$$

It is suggested that the sEMG data is normalised to MVC to reduce the variability of MNF between subjects. Additionally, the filtering process typically used (that is, 20Hz to 400Hz) should be avoided as it destroys the information particularly contained in the lower frequencies and associated with deeper MUs. Averaging of sEMG and force data, to present a singular data stream has been proven successful by Chen *et al.*, (2019) [143].

In a nutshell, feature extraction of sEMG signals may or may not be needed depending on the type of ML approach taken. Despite this, it is beneficial in the realisation of force estimation from sEMG signals, since it provides for the incorporation of expert knowledge. Included in expert knowledge are perceptions regarding amplitude, frequency and complexity, giving rise to the adoption of RMS and MNF in this dissertation.

4.1.4 sEMG-force predictive models

The mapping of sEMG signals to force is a complex process that requires advanced mathematical and computational tools. This can be an iterative process, the accuracy of which is largely dependent on the prediction model used during training. It should be noted that the more accurate the predictive model used, the more computationally expensive it will be. Furthermore, both accuracy and computational speed are contributing factors in the selection of the best force predictive model to use. With advancements in machine learning, most computer software are now able to drastically reduce the time taken by a model to map and produce a model input-output relationship.

4.1.4.1 Classical biomechanical sEMG-force models

Models approximating the relationship between sEMG signals and muscle force are generally of two types, mathematical or computational. Despite using mathematical constructs, computational models are less explicitly defined. In this section, some fixed-parameter and bounded mathematical muscle models will be discussed. According to Winters and Stark [144], the number of muscle models increased significantly after the introduction of Hill's muscle model. It can be categorised into three classes, including simple second-order models, Hill-based models and Huxley-based models. The most prominently adopted model is the Hill-based model due to its simplicity, ease of calibration and accuracy (for certain functions) [145]. This model gives a macroscopic overview of the processes of a muscle, however, does not yield information regarding the intricate processes occurring within the muscle (as with a Huxley-based model).

The Hill muscle model is shown in Figure 4.3 and is, in essence, a mass-spring-damper equivalent of a skeletal muscle with the addition of tendinous links. Also, the outlined muscle is a lumped-parameter representation of all individual MUs. The angle that the muscle makes with the tendon (i.e., the pennation angle indicated by ϕ) determines the components of tensile force exerted on the load and dependent on the muscle fibre length in time. Within the muscle model, there are three main elements, namely the isometric Contractile Element (CE), the Series Elastic (SE) element and lastly the Parallel Elastic (PE) element [146]. The CE and SE branch is involved in the active behaviour of the muscle, while the PE branch remains passive. More specifically, the PE may be described as the elasticity of the passive muscle and ligaments (which is a resistive force), and only affects the force if stretched beyond the resting length.

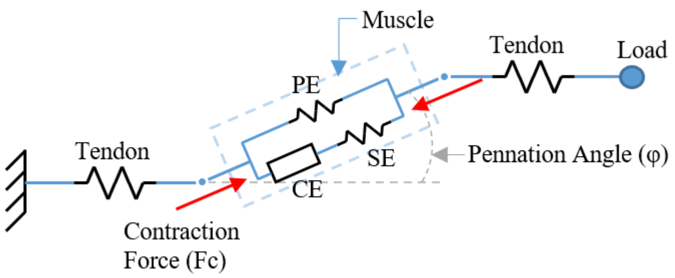


Figure 4.3: A muscle-tendon system with the incorporation of a Hill's contractile element [146].

Input is required in order to enact force generation in the muscle model. This input starts as a neural command, resulting in a delayed twitch response from the chemical processes taking place in the CNS. Equation 4.1.2, relates this neural activation to the sEMG signal through a critically-damped second-order linear differential equation [146, 147]; where $u(t)$ is the neural activation, E is the sEMG signal, d is the delay between the neural activation and the sEMG signal, and α, $\beta 1$ and $\beta 2$ are coefficients describing the differential equation respectively. To ensure stable neural activation $\beta_1 = C_1 + C_2$ and $\beta_2 = C_1 \cdot C_2$, where C_1 and C_2 are arbitrary constants that are within the range $-1 < C_n < 1$. Also, the condition $\alpha - \beta_1 - \beta_2 = 1$ must be met to maintain unity gain [148].

Depending on the muscle, several researchers have reported that the activation dynamics are instead non-linear and may not be assumed to be equivalent to the neural activation. The degree of non-linearity may also differ, and so the proposed equation joining neural activation and the activation dynamics is variable.

$$u(t) = \alpha \cdot E(t-d) - \beta_1 \cdot u(t-1) - \beta_2 \cdot u(t-2) \tag{4.1.2}$$

Equation 4.1.3 describes an attempt at achieving the non-linear relationship between neural activation and activation dynamics by defining two regions [149]. The first region is logarithmic and is valid within the range of 0% to 30% MVC. The second region is linear and extends from 30% to 100% MVC. The exact reason for splitting the relationship is because it was empirically found that beyond 30% MVC, the relationship remains predominantly linear. In Equation 4.1.3, $a(t)$ denotes the activation dynamics, while d, c, m and b denote constants. The constants m and b may be solved by substituting the boundary conditions at 0.3 and 1, while the constants d and c are calculated by preserving continuity at the node joining the derivative of the linear and non-linear parts.

$$\begin{aligned} a(t) &= d \cdot \ln c \cdot u(t) + 1 && 0 \leq u(t) < 0.3 \\ 82pt]a(t) &= m \cdot u(t) + b && 0.3 \leq u(t) < 1 \end{aligned} \tag{4.1.3}$$

Another simple and effective non-linear formulation describing the transition from neural activation to activation dynamics was proposed by Besier and Lloyd [150], shown in Equation 4.1.4. Here, A is a constant, suggested to be in the range 0 to -3 (linear to highly non-linear). Note that in both Equations 4.1.3 and 4.1.4, $u(t)$ term refers to Equation 4.1.2. The active component in the Hill muscle model is dependent on two factors, namely the length of the muscle (force-length) and the contraction velocity (force-velocity). Shown in Equation 4.1.5 is the active force produced by the muscle ($F_A(t)$), where f_1 and f_v are the normalised length and velocity-dependent muscle forces [151]. These relationships are derived experimentally and are given as polynomial curve approximations in the literature [152]. The symbol FmaxIso refers to the participant-specific maximum isometric muscle force and may be easily measured. The maximum active force is achieved at the optimum muscle fibre length when the optimal connection between the actin and myosin is achieved, and the sarcomeres are 2.8μm in length [153]. The total contractile force F_c is a combination of the active and passive force. Information regarding the passive force and thermodynamics (i.e., the heat generated from muscle while shortening) may be found in work by Buchanan *et al.*, [154].

$$a(t) = \frac{e^{(A \cdot u(t))} - 1}{e^A - 1} \tag{4.1.4}$$

$$F_A(t) = a(t) \cdot f_l \cdot f_v \cdot F_{maxIso} \tag{4.1.5}$$

The main limitation of mathematical models relating sEMG to force is the required estimates of activation dynamics and the translation from activation dynamics to muscle force remaining largely unknown [154]. These approximate relationships are commonly the result of curve fitting with minimal parameters as possible. Despite being simple, this poses several problems since real systems do not abide by this. A better alternative, with much more promise, lies in computer-based approaches.

4.1.4.2 Linear vs. non-linear debate

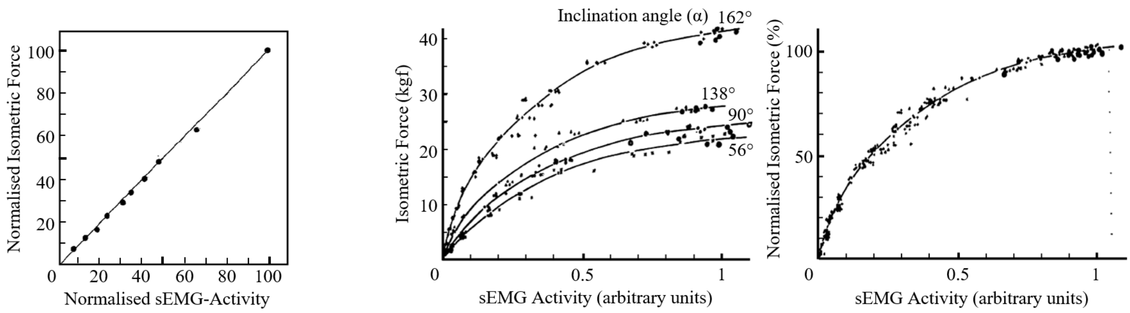
There remains uncertainty in the exact interaction between sEMG activity and force produced by a muscle. Researchers are divided on whether the interaction may be defined as either having negative, positive or no correlation. Also, they are often classed as either linear or non-linear (with upward/downward concavity [155]), for instances where the correlations are observed. Imprecise and differing methods, as well as the factors stated in Section 4.1.4.1, are typically blamed for the disparity in results. According to Farina *et al.*, [133], the most significant cause is the difference in muscle composition leading to the variation in proportionality between MU rate coding and recruitment. Ahamed *et al.*, [156] presented another point, stating that electrodes located suboptimally (at the ends of the muscle) could account for the difference in linearity.

Several studies propose a linear relationship between sEMG activity and muscle force. These relationships often relate the temporal properties of both the sEMG signal and muscle force, with negligible emphasis placed on relating the frequency properties. One such example can be found in work by Jones and Hunter [157] where the relative sEMG activity was mapped to the relative isometric muscle force (see Figure 4.4a). In the study, the muscles (which include the flexor carpi radialis and flexor digitorum superficialis) were investigated separately, and each produced a linear relationship. The variance between the linear fit and the data from the participants fell within the range of 72% and 98%. Despite the linear relationship suggested by Jones and Hunter, they did not consider the influence of cross-talk stemming from a multi-muscle analysis. It may be safe to assume that cross-talk may introduce non-linear aspects [158]; thus, the results of this study is limited to the aforementioned muscles in isolation. As per the findings of Bigland *et al.*, [159], the relationship may differ among participants due to the mixture of type I and II muscle fibres. It is therefore hypothesised that assuming a non-linear relationship would be the safest option.

Similar to Jones and Hunter, Kamavuako *et al.*, proposed that there exists a linear relationship between sEMG features and muscle force [160]; however, they noted that some features may produce non-linear relationships and substantiated this through the use of non-linear ANN approach. In addition, the studies only considered one muscle, highlighting again that a linear profile should not be assumed when dealing with multi-muscle analyses. Kamavuako *et al.*, [160] performed the trials under quasi-static conditions (slow progression of force patterns); hence the results may not reflect the entire muscle behaviour but rather only the portion attributed to type I muscle fibres. This fact is echoed in a study by Coggshall and Bekey [161], in which it was noted that significant linear prediction errors occurred in intervals of rapid force change. From the results of Kamavuako *et al.*, it may be said that the data should reflect a variation in the dynamics of grasping; otherwise, it precludes a thorough mapping of the muscle.

Contrary to the strictly linear relationships suggested by the previous researchers, Lawrence and De Luca [162] found that both the bicep and deltoid exhibited a non-linear sEMG-force behaviour, while the first dorsal interosseous muscle exhibited a quasi-linear behaviour. These results seem to validate the findings of Woods and Bigland, highlighting that different muscles yield different relationships. The non-linear relationship found by Lawrence and De Luca may further be classified as a parabolic relationship which approaches a vertical asymptote. The results remained consistent among various force levels with different numbers of average contraction. Similarly, Vredenbregt and Rau [163] points out that the parabolic relationship persists for different inclination angles, which is the angle between the forearm and the upper arm, as shown in Figure 4.4b.

More recently, Al Harrach *et al.*, [164] used a simulation study to analyse the parameters affect the underlying relationship between sEMG and muscle force. The effects of several parameters, including MU recruitment and firing rate, adipose tissue thickness, tissue thickness and muscle length, were monitored. These parameters did not change the overall shape of the relationship between measures of sEMG amplitude and muscle force and may be related by a 3^{rd}-order polynomial with upward concavity.



(a) Confusion matrix **(b)** ROC plot

Figure 4.4: Interaction between sEMG activity and force produced by a muscle.

The shape of the relationship between sEMG activity and muscle force is shown in Figure 4.5. Here, three mappings are expressed, where the force is on the y-axis and the sEMG activity on the x-axis. The variables are interchangeable since they are co-dependent; however, conventionally force is presented on the y-axis. Typical equations describing the shape are given alongside the graphs.

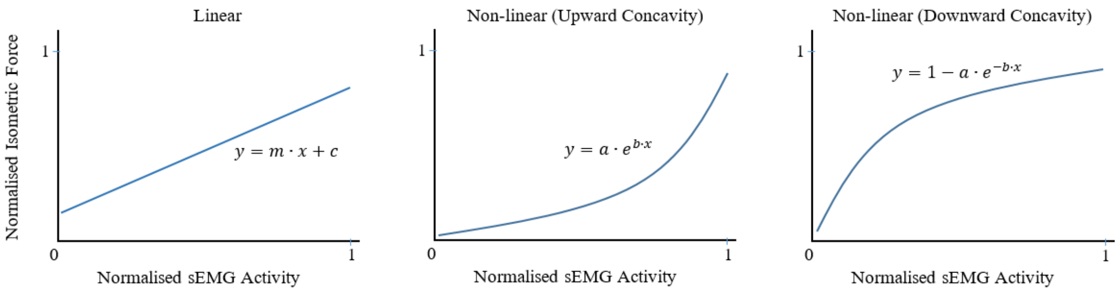


Figure 4.5: A muscle-tendon system with the incorporation of a Hill's contractile element [165].

As pointed out, most studies generally only consider linking temporal amplitude characteristics of sEMG signals to muscle force, leaving the frequency characteristics largely unexplored. In Section 4.1.3, this matter was touched on, stating that three distinct correlations between frequency-based sEMG features and muscle forces are present. However, this fails to clarify the linearity or non-linearity of the relationship. In conclusion, there are numerous cases advocating both linear and non-linear relationships between temporal sEMG characteristics and muscle force, while the same may not be true for the mapping of sEMG frequency features and muscle force.

4.1.4.3 Computational sEMG-force models

The invent of better computing systems has led to an increase in the use of ML in the last 20 years. In particular, this had an effect on the mapping of input sEMG signals to output force, shifting the focus from classical to computational-based methods. The reason being that these methods are fast, accurate (especially for non-linear systems), adapt to change over time and are capable of deducing relationships without any prior knowledge about the system, that is the Black-Box model of the muscle and joint dynamics.

Correspondingly, Wen *et al.*, [166] used a feedforward ANN, trained with adaptive movement estimation (ADAM) algorithm, to map the relationship between sEMG signals recovered from the forearm and the force produced during grasping. The force estimated by the model was used as a reference signal in an admittance Proportional Derivative (PD) controller, to control a 6-DoF Kinova arm with endpoint gripper. With an sEMG-force regression fit of $R^2 = 0.95$, the gripper was able to grasp and lift deformable objects without producing any noticeable deformation. Also, the ANN regression model showed good generalisation abilities, which was verified by considering the difference in RMSE between the training and validation sets. The study neglected the effects of various grasps and therefore is limited in its results.

Similarly, Choi *et al.*, [167] developed an sEMG-force model for the real-time prediction of palmar pinch force. The model consisted of an ANN with a single hidden layer and 12 hidden neurons. The MAV inputs were taken from three muscles in the forearm (optimally determined through LDA). The research consisted of two sessions, with the first being static and dynamic force patterns and the second being random (participant specified). The results from the first session were slightly better, with a normalized root mean square error (NRMSE) and correlation coefficient of 0.081 and 0.968. In general, the results reiterated that the sEMG-force relationship is non-linear at lower forces and that the relationships are participant-specific. However, the latter conclusion is questionable following the disregard of normalisation.

Further evidence of the use of ANNs in estimating force from sEMG signals was found in work by Zhang *et al.*, [168]. Here, an ANN was combined with modified aspects of a Hills muscle model to predict the downward touch force applied. The ANN classified the sEMG signals into four distinct hand motions and acted as a switch, activating subsequent blocks in the model if a force exerting hand motion was sensed. These blocks consisted of a muscle model and a predictive function, that was calibrated through Bayesian linear regression. The ANN had an accuracy of 96%. In contrast, the combined effort of all the sub-systems resulted in an RMS error of less than 2.27N. Despite having good performance, the filtering parameters used in the study are met with disagreement by several researchers [169, 170], calling into question the validity of the results.

Luo *et al.*, [109] combined the architecture of an ANN with FL abstractions to produce a three-domain fuzzy wavelet NN for describing the sEMG-force relationship. The proposed network was comprised of four layers, namely an input layer, fuzzy wavelet layer, defuzzification layer and output layer. The performance of the network (MSE less than 0.083N and an R^2 greater than 0.99) was found to surpass the performance of a radial basis function NN. The radial basis function NN was associated with an sEMG-force model that tended towards an underdamped system.

Reference to NNs in estimating force from sEMG signals extends beyond those mentioned in this section. Furthermore, other computational approaches such as SVM [171], extreme learning machine, CNN [172], LSTM [173], fast orthogonal search [174] and Gaussian Process Regression (GPR) [112] have been used successfully to model the sEMG-force relationship.

Together with the type of regression approach, the algorithm used to minimise the cost function makes a significant impact on the convergence rate and the error produced. Though many training algorithms exist, the LM algorithm has proven successful in particular with cases where a moderate number of parameters (say 1000) are concerned. For these reasons, the LM algorithm was selected in this chapter to train the ANN sEMG-force regression model. The LM algorithm is used to solve a non-linear least squares problem and takes advantage of two techniques to minimise the mean or sum of the squared errors. The first technique is gradient descent, which takes steps towards the local minimum in proportion to the gradient of steepest descent and learning rate. This technique performs better further from the local minimum. The second technique is Gauss-Newton, which similarly aims to reduce the sum of the squared errors by iteratively solving it in a linear least-squares manner. The Gauss-Newton technique is highly sensitive to the initial guess and converges exceptionally well if this guess is close to the minimum.

The LM algorithm combines the best of both techniques and behaves like the gradient descent technique when the residual is far from minimal and like the Gauss-Newton technique when the residual is close to the minimum. Shown in Algorithm 4 is the LM algorithm, which may be used to update the weights and biases of a NN in application to non-linear regression [175]. Supplementary to Algorithm 4 is the Equations 4.1.6 to 4.1.15. These equations briefly describe the concept of backpropagation with gradient descent as well as the Levenberg-Marquardt algorithm. The derivation and explanations that follow are based on work by Hagan and Menhaj [176].

Backpropagation relies on first considering the forward pass of a NN. In other words, how the input propagates through the network and results in an error. Related to the forward pass, Equation 4.1.6 states that the output of the i^{th} neuron in the first layer, denoted by a_i^1, is equal to the input of the i^{th} neuron, denoted by x_i. A general expression defining the input to each neuron is given by Equation 4.1.7. In Equation 4.1.7, the input to each neuron in the layer $l+1$, is given by the sum of the outputs of the previous layer l multiplied by the corresponding weights from the k^{th} to the j^{th} layers, and an added bias b_i^{l+1}. Note that the k^{th} and j^{th} layer refer to the layers $l+1$ and l, respectively. Also, in Equation 4.1.8, the output vector of the neurons in the $l+1$ layer, symbolised by $\underline{a}^{l+1}$, is equal to the activation function vector (see Section 3.1.4), given as $\underline{f}^{l+1}$, evaluated at the corresponding input vector ($\underline{z}^{l+1}$).

$$a_i^1 = x_i \tag{4.1.6}$$

$$z_i^{l+1} = \sum_{i=1}^{N} a_i^l \cdot w_{k\to j}^{l+1} + b_i^{l+1} \tag{4.1.7}$$

$$\underline{a}^{l+1} = \underline{f}^{l+1}(\underline{z}^{l+1}) \tag{4.1.8}$$

Using Equations 4.1.6 to 4.1.8, with initial weights and biases (randomly or optimally selected), the output of the NN is computed. Effectively updating the weights and biases requires the use of the networks error, that is the difference between the desired (target) and actual outputs. Shown in Equation 4.1.9 is the total error of the network, given as the MSE with an additional $\frac{1}{2}$ term to negate the derivative. Here, E_{total} is the total error of the network, N is the number of neurons in the last layer, t_i is the target output of the i^{th} neuron, and p_i is the output of the i^{th} neuron in the last layer, that is Equation 4.1.8 (scalar form) evaluated at layer N.

$$E_{total} = \frac{1}{2N}\sum_{i=1}^{N}(t_i - p_i)^2 \tag{4.1.9}$$

Backpropagation is a process whereby the errors are passed backwards through the network, evaluating the sensitivity of both the weights and biases along the way. Shown in Equations 4.1.10 and 4.1.11 is the premise of the backpropagation algorithm. In these equations, the updated weights and biases, denoted as$\hat{w}^{l+1}_{k\to j}$ and $\hat{b}^{l+1}$, are equivalent to the previous weights and biases minus the product of the learning rate (η) and the partial derivative of the total error with respect to the weight or bias. The partial derivative is the implementation of the gradient descent algorithm and shows how much either the weight or bias needs to change to minimise the cost function. Also, the learning rate is defined within the range $\in \{0; 1\}$ and characterises the scaling of steps, where the learning rate is proportional to the step size. If the learning rate is large, it will tend to overshoot the cost minimum, while if it is small, it will converge slowly and be susceptible to a local minimum.

$$\hat{w}^{l+1}_{k\to j} = w^{l+1}_{k\to j} - \eta \cdot \frac{\partial E_{total}}{\partial w^{l+1}_{k\to j}} \tag{4.1.10}$$

$$\hat{b}^{l+1} = b^{l+1} - \eta \cdot \frac{\partial E_{total}}{\partial b^{l+1}} \tag{4.1.11}$$

The gradients defined in Equations 4.1.10 and 4.1.11 must be computed using the chain rule. Equation 4.1.12 and 4.1.13, show the premise behind the chain rule with respect to the weights and biases. Note that the first term, denoted as ∂^l_i, is sometimes referred to as the sensitivity of the error with respect to the parameter [177].

$$\frac{\partial E_{total}}{\partial w^{l+1}_{k\to j}} = \frac{\partial E_{total}}{\partial z^{l+1}_i} \cdot a^l_i \tag{4.1.12}$$

$$\frac{\partial E_{total}}{\partial b^{l+1}} = \frac{\partial E_{total}}{\partial z^{l+1}_i} \tag{4.1.13}$$

In multivariate optimisation, the change in the parameter for Newton's method is shown in Equation 4.1.14. $\Delta \underline{x}$ is the change in the updated parameter vector, H is the Hessian matrix (matrix of second-order derivatives), $\dot{E}(\underline{x})$ is the derivative (gradient) of the error with respect to the parameters, and superscript -1 refers to the inverse matrix. Both the Hessian and derivative are often complex to compute, particularly when applied to NNs. Thus, the Gauss-Newton method proposes a Jacobian approximation, given by $\dot{E}(\underline{x}) = J^T(\underline{x} \cdot \underline{e}(\underline{x}))$ and $H(\underline{x}) = J^T(\underline{x}) \cdot J(\underline{x})$. J is the Jacobian matrix (or partial derivative of the error with respect to the parameter), and T is the transpose of the matrix.

The LM Algorithm 4 is an adaptation of Equation 4.1.15, with the addition of a scaling parameter which switches between the gradient descent and Gauss-Newton method. The scaling parameter is given as μ, and I is an identity matrix. Note that μ is changed during each iteration of the optimisation of a parameter. The modification is done by setting $\mu = \frac{\mu}{\beta}$ if the updated error < error and $\mu = \mu \cdot \beta$ if the updated error > error. The value β is decided upon at the start of training and reflects the rate at which the LM algorithm switches between the two constituent algorithms. The modification may sometimes be expressed as a deduction or addition of β [178]. To compute the updated error, the new value of x is, $x = x + \Delta x$, is used. The steps required to apply the LM algorithm are shown in Algorithm 4. An important factor to remember is that the parameters only get updated once the backwards pass is complete.

$$\Delta \underline{x} = -[H(\underline{x}) \cdot E(\underline{x})]^{-1} \cdot \dot{E}(\underline{x}) \tag{4.1.14}$$

$$\Delta \underline{x} = [J^T() \cdot J(\underline{x})]^{-1} \cdot J^T(\underline{x}) \cdot \underline{e}(\underline{x}) \tag{4.1.15}$$

Algorithm 4: Pseudo-code for GR feature extraction of sEMG data

Data: sEMG features (input) and force (output)
Result: Optimum weights and biases
1 Load data;
2 Using initial weights and biases, perform a forward pass and calculate the outputs;
3 Determine e_1;
4 Compute the cost;
5 Evaluate J;
6 Solve Equation 4.1.15;
7 Update parameter (weight or bias) and compute cost;
8 Bool = True;
9 **while** *Bool = True* **do**
10   **if** *Solution to line 7 < solution to line 4* **then**
11   set $\mu = \frac{\mu}{\beta}$ and return to Step 4;
12   **end**
13   **else**
14   set $\mu = \mu \cdot \beta$ and return to line 6;
15   **end**
16   **if** *Epochs > $Epochs_{Max}$ or Time > $Time_{Max}$ or Performance < Goal or Preformance Gradient < $Gradient_{Min}$ or $\mu > \mu_{Max}$ or Validation > $Fail_M ax$* **then**
17   Bool = False
18   **end**
19 **end**

The performance measures of regression are given in Equations 4.1.16 to 4.1.19. These measures are in line with the performance specifications in Section 2.1.2 and were used in this dissertation to assess the difference between the predicted model outputs and the actual outputs. Equation 4.1.16 defines the sum of the squared errors, where t_i and p_i are the target and predicted model outputs, respectively. In addition, i refers to each target-predicted output pair, and N is the total number of target-predicted output pairs. Equation4.1.17 defines the mean squared error, a metric commonly used to evaluate the performance of a NN, while Equation 4.1.18 defines the mean absolute error. The aforementioned performance metrics are defined within the range 0; . The final performance metric to consider is the coefficient of determination, shown in Equation 4.1.19, which indicates the degree of correlation between two variables.

$$SSE = \sum_{i}^{N} \underbrace{(t_i - p_i)^2}_{e_i} \tag{4.1.16}$$

$$MSE = \frac{SSE}{N} \tag{4.1.17}$$

$$MAE = \frac{\sum_{i}^{N} |e_i|}{N} \tag{4.1.18}$$

$$R^2 = 1 - \frac{SSE}{\sum_{i}^{N} (x_i - \underbrace{\frac{1}{N} \sum_{i}^{N} x_i)}_{\bar{x}}} \tag{4.1.19}$$

In summary, many computational methods are used to relate sEMG activity to muscle force. These methods may be linear or non-linear, yet the vast majority of these are non-linear due to the prominence of ANN-based models. Computational modelling offers numerous advantages over classical models; however, computational models often lack insight or expert direction. This may be curbed through the careful selection of the computational method as well as the features.

4.1.5 Literary relevance and conclusion

An sEMG-force model is based on electrical conduction in the nervous system. The force generated depends on the MU recruitment and firing rate for different skeletal muscles. The size principle relates MU recruitment to the force generated by the muscle action. Various force-sensing devices are available; an example uses FRS for data acquisition. However, most of these devices can only accurately capture a limited range of force types. The use of a low-pass filter can be used to avoid raw sEMG force data from being corrupted by noise at low frequencies. The feature extraction process can be implemented to reduce the machine learning model complexity, but it is computationally expensive. In this study, the frequency-based MNF feature extraction model is used. This model can be related to the force and can also be used to detect muscle fatigue. The use of computational methods is much more efficient in predicting the force relationship by taking advantage of the Black-Box model. However, unless modified, most computational models are grasp specific and are thus limited in application. Based on literature findings, the Hill-based mathematical model is the most used biomechanical sEMG force model.

4.2 Experimental Facilities

The experimental test facilities used in this chapter considers both the hardware and software elements used to acquire and process sEMG and force data to develop a grasp force regression model. The experimental test facilities involved in force data acquisition includes a force glove and Grip Visualisation (GVis) application. A calibration curve of the Force Sensitive Resistor (FSR) with coupling disks is presented. The alternative curve (excluding coupling disk) can be found in the datasheet in the Appendix. The code written for the GVis application is two-fold (Universal Asynchronous Receiver/Transmitter communication protocol) and is submitted as part of the Digital Appendices.

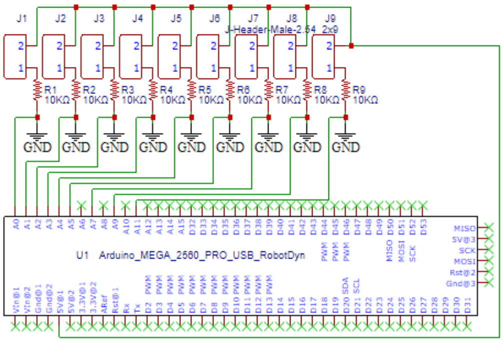
The experimental facilities used in Chapter 3 were required in the execution of the sEMG procedures detailed in this chapter. These include the Delsys Trigno setup, grasping apparatus, processing system, EMGworks, Python and MATLAB. For more information on the setup, refer to Section 3.2.

4.2.1 Touch-sensitive glove

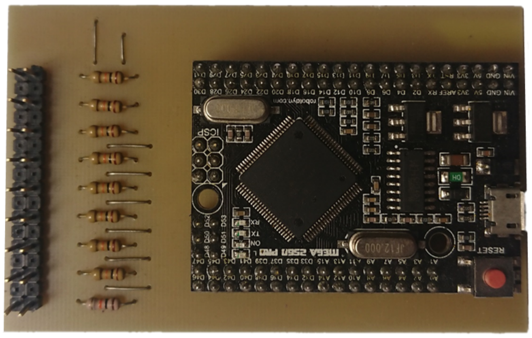
The touch-sensitive glove makes use of nine FSRs located on the pads of the distal phalanges and the palm. The FSRs are divided as follows: one FSR on each of the pads of distal phalanges (that is, five total); four on the palmar side of the hand located at the thenar, hypothenar and below the index and pinky MCP joints, respectively. These positions were established after performing an ink touch test with the various GOs referred to in Section 3.2.1.2. The same procedure is noted in the work by Seedat *et al.*, [179].

Each FSR is made up of two comb-like layers which intermingle but do not make contact with one another. If a force is applied perpendicular to the layers, they deform, and the distance between them and the conductive film decreases. The distance continues to decrease at which point the comb-like layers make contact with the conductive film; a larger contact area equates to a lower measurable resistance. Using a measuring resistor in a voltage divider configuration; the resistance change may be related to a change in voltage. The voltage may further be digitised using an ADC. Owing to the deformation-resistance relationship; it should be noted that without offset calibration, the FSRs should strictly remain in linear alignment with the substrate. Furthermore, bending the tail of the FSR changes the resistance and must be accounted for through calibration.

Before adhering the FSRs in the locations specified on the glove, a signal routing board with nine voltage dividers was constructed. Figures 4.6a and 4.6b respectively show the board and its corresponding circuit diagram. The primary purpose of the board was to link the output of the FSRs to the analogue input pins of the Arduino Mega 2560 PRO MINI through a voltage divider with a resistance of 10kΩ (measuring resistor). Constructing the board had three profound advantages; it reduced the risk of jumper wires coming loose during data acquisition, lessened the potential for noise and compactified the setup. This made it easier for the participants to perform the investigate grasps.



(a) Confusion matrix.



(b) ROC plot.

Figure 4.6: FSR signal routing board and circuit diagram.

Considering the pressure-sensitive nature of the FSR, a coupling disk was placed on either side of the FSR. This is in-line with the recommendations given by Hashemi [105]. Adding a coupling disk ensured that the pressure was uniformly distributed, improving reliability between tests and avoiding fatigue failure. The coupling disks were layers of hot glue approximately 1mm in thickness, covering the active area and tail on either side of the FSR. By laying the hot glue slightly over the edge of the FSR, the coupling disks were fully constrained. It should be stated that during the process of attaching the coupling disks, several FSRs were broken. This may be attributed to the heat from the hot glue, causing the FSR layers to delaminate. The FSRs (with coupling disks) were adhered to the glove using superglue.

The orientation of the FSRs was of particular significance to its fatigue life. It was observed that FSRs adhered to the palmar side of the glove resulted in fractures occurring along the axis of the DIP joint. To resolve this, a similar approach to Ye *et al.*, [106] was taken, positioning the FSRs such that the tail resided on the dorsal side of the hand. Consequent to the position of the FSRs, static calibration was necessary and opposed dynamic calibration, which would have resulted from adhering the FSR to the palmar side of the hand (movement of the DIP joint). In this dissertation, the orientation of the FSRs on the glove does not translate to the RPH. This is because the DIP joint is fixed in a set position.

The participants were right-arm dominant; therefore, a right-handed glove was required. Also, the glove had to maintain the participant's range of motion and cater to a variety of hand sizes; thus, a stretchable cycling glove was settled upon. The two solder tabs on the tail end of each FSR were soldered to an insulated wire (0.8mm in diameter). Using superglue; the wires were combined into a single ribbon cable and soldered to male header pins, which attached to the board. The touch-sensitive glove is shown in Figure 4.7. Upon activation of the board, the software discussed in the following section came into effect.

4.2.2 Grip visualisation application

The GVis application, shown in Figure 4.7, was developed to capture the force data from the touch-sensitive glove in real-time. An alternative approach would have entailed

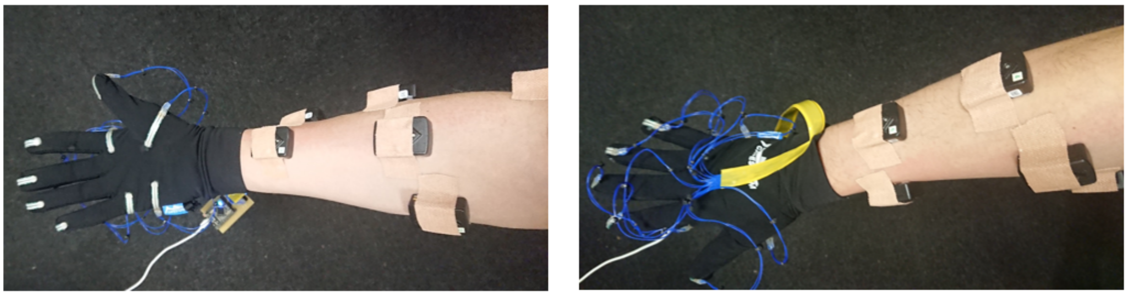


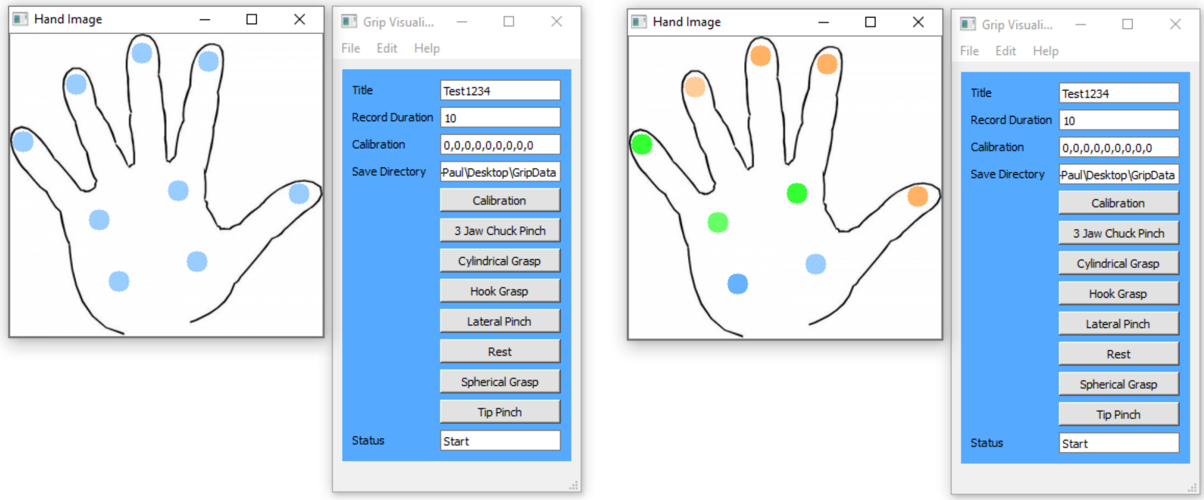
Figure 4.7: Touch sensitive glove and FSR location in the set position.

using a breakout board with a micro SD module. Two drawbacks of using the micro SD approach are the cost associated with the breakout board and the inability to process the data online (real-time), to serve as a visual cue for the participants.

The GVis application was written in Python, with the layout of the GUI widgets established in PyQt. By determining the layout of the application in PyQt, the task of manually adjusting widget placement was avoided. Furthermore, compiling the GUI developed in PyQt using PySide resulted in a class structure. The class structure was linked to the functionality of the application in a separate master script through object-oriented programming. The functionality entailed the following:

1. The calibration and offset adjustment of the FSRs upon start-up ensured accountability of the deformation due to the coupling disk and tail (see Figure 4.8). The offset was unique to each FSR and involved subtracting the current reading from the initial reading. The calibration was fixed to a 10s period, in which the participant's hand was to remain in the set position shown in Figure 4.7.

2. The real-time visualisation of the force distribution on the hand while grasping the various GOs was shown in the GUI of the GVis. The output of the FSRs was segmented in ranges and assigned to a colour spectrum (visually representing the force), where light blue and saturated red respectively represented the lowest and peak force (refer to Figure 4.8). By visualising the force distribution, the participants could assess the progression of force, from light grasping to heavy grasping. Also, the duration of testing (30s default), title and saving directory of the data (in CSV format) could be adjusted before the start of each grasp.



(a) GVis application calibration. **(b)** GVis application recording.

Figure 4.8: GVis graphic user interface.

The calibration of the FSRs was established through the use of 100g mass increments; which was done up to a maximum of 1kg. Each FSR was calibrated individually since the coupling disks acted as variable dampers. The calibration curves used for each FSR are shown in Figure 4.9. Assuming environmental conditions insignificant; the error associated with each voltage reading is ± 2.5mV (half the resolution of the ADC).

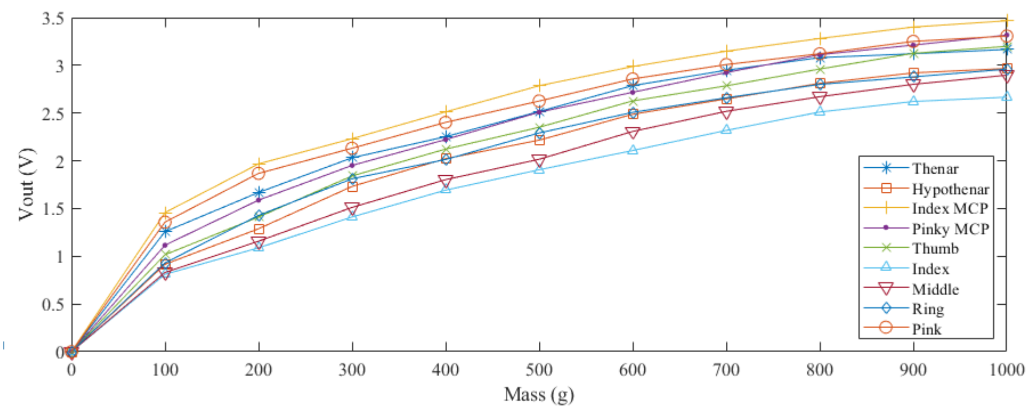


Figure 4.9: FSR calibration curves.

As part of the two-fold communication protocol; the Arduino Mega 2560 PRO MINI (client) output serial data was used. Each serial sample was a string of integer values (FSR readings), separated by a comma and terminated with a unique trailing character. Communicating through a designated *COM* port; the serial data was read in sequentially with the stopping bit set to the unique trailing character. The factors which determined the sampling rate of the force data were, the speed of the *AnalogRead* task and the communication protocol baud rate. By measuring the loop time, it was concluded that the *AnalogRead* task (taking 6ms per loop) was the most significant hindrance to the sampling rate. The sampling rate was 167Hz, which was well below that of the sEMG sensors.

4.3 Methodology

The methodology outlined in the following section is broken down to four stages, where the first three stages are built on concepts outlined in Chapter 3. The first stage covers the simultaneous acquisition of sEMG and force data. Most importantly focus is shifted towards the acquisition of force data using the developed GVis application. The second stage briefly summarises the aforementioned stance on sEMG data pre-processing and introduces the force data pre-processing procedure undertaken. Stage three considers the extraction of time and frequency-based features, which are linked to MUAP amplitude and MU firing rate, respectively. Upsampling (via linear-interpolation) is put to use to ensure the sampling rates of the sEMG and force signals are coherent and synchronised with one another. The last stage describes the approach used to develop several regression models which include linear, regression trees, Gaussian process, SVM and ANN models.

The outcome of the methodology is a logical and structured approach to producing an acceptable sEMG-force regression model, which, when implemented, will serve as the grip compensation component for the RPH. Each stage is summarized with a flow-diagram before being presented in a step-wise manner. It should be noted that the participants referred to in this chapter are the same as those employed in Chapter 3.

4.3.1 Electromyography

Concurrent with force data acquisition, sEMG signals were acquired much in the same manner as described in Section 3.3.1. In summary, the procedure entails skin preparation, target muscle palpation, sEMG sensor attachment and assessment, task definition, sEMG data recording, MVC data recording, removal of sEMG sensors and batch converting of recorded data. Elements of the procedure were adapted to suit the focus of this chapter; hence each grasp was conducted for one iteration spanning 30s. Also, the MVC data consisted of a single iteration, per participant, spanning 10s. Note that the duration of the MVC iteration did not extend to 30s, but was limited to 10s to reduce fatigue-related biases.

The pre-processing of sEMG data for the development of the sEMG-force model follows the methodology explained in Section 3.3.2. Here the data is manipulated before being filtered with a combination of bandpass and notch filters. Rectification is not required due to the definition of the features intended on being extracted.

Feature extraction is explained in Section 3.3.3 and is loosely described as a process in which underlying characteristics are retrieved from a signal. In the development of the sEMG-force model, two features were selected, namely RMS and MNF. These features represent the MUAP amplitude and firing rate, respectively. The mathematical description of the features is given in Equation 4.1.14 and 4.1.15. This is calculated for each window, refer to Algorithm 4 for the sliding window pseudo-code. Note that the window and overlap length were kept constant throughout this dissertation that is 80ms and 20ms. The features extracted from the pre-processed sEMG data are used as the input to the regression sEMG-force model.

4.3.2 Force

Muscle force generation is a result of neural stimuli that cause the movement of intermediate connective tissue which actively pull bones resulting in hand movements. Depending on the type of hand movement required, a specific motor pool is activated to coordinate muscle contraction for the specific hand movement accurately. The output from this process, which is the skeletal muscle movement was measured using the developed GVis. The process of force data acquisition is outlined in the section that follows.

4.3.2.1 Data acquisition

Force data acquisition is a process in which the outputs of skeletal muscle movements are measured and recorded. From the explanation in the previous section, muscle movements are transferred to the touch-sensitive glove, which has sensors attached to it. These sensors, as outlined in section 4.2.1, measure the force generated by the hand during different grasping actions.

The measured sensor contractions due to the hand movement can be transmitted to the developed GVis. Also, due to the various contraction levels of the sensors on the glove, different force magnitudes would be recorded. Different colour codes categorically visualise these force results on the GVis depending on their magnitudes. Categorically visualising the force results using the GVis assists in determining the endpoint of force applied by the participant. This is achieved by continuously monitoring changes to the colour codes on the respective FSR locations on the touch-sensitive glove. Figure 4.10 shows a summary of the data acquisition process, followed by steps detailing the process of data acquisition.

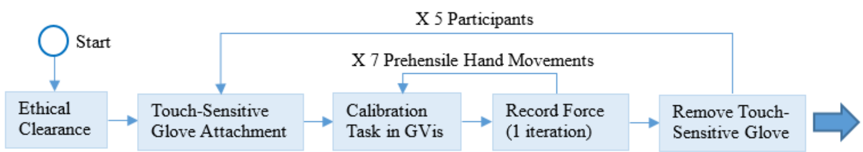


Figure 4.10: Summary of force data acquisition methodology.

Step 1 As with GR, the methodology described in this chapter is contingent on receiving ethical and registrar clearance (see Appendix; M190766). The ethical and registrar clearance must cover the scope of the work being investigated, and no data collection may commence before receiving them.

Step 2 The participant was instructed to put on the touch-sensitive glove. Care was taken to ensure the FSRs were situated directly above the specified locations stated in Section 4.2.1. The routing board was taped to the upper-arm of the participant with the intent of improving the range of motion.

Step 3 After connecting the routing board to the processing system, the GVis application was launched and the calibration process initiated. Once calibration was completed, the status changed from *start* to *end*, and the calibration values in the line edit were updated.

Step 4 The principal investigator demonstrated the various grasps, after which the duration, title, and save directory of the test were specified in the GVis application.

Starting with 3 jaw chuck; the test was conducted for one iteration spanning 30s. Again, a status of *end* signified the completion of the test. During the test, the participant was instructed to vary the grip force from a light grasp to heavy grasp (randomly). This was verified using the real-time force indicator in the GVis application. The tests were repeated for the other grasps, namely cylindrical grasp, hook grasp, lateral pinch, rest, spherical grasp and tip pinch.

Step 5 The routing board was disconnected and removed from the upper-arm after verifying the force data. Finally, the participant was asked to remove the touch-sensitive glove gently and was subsequently given hand sanitiser.

Step 6 Step 2 to Step 5 was repeated for a total of five participants.

4.3.2.2 Pre-processing

Once force data has been measured and recorded, the next step would be to classify the data through a series of structuring and filtering processes. This data manipulation process takes place in the GVis developed. The data is manipulated so that corrupt and unwanted signals can be eliminated before the feature extraction process. Figure Figure 4.11 shows a summary of the force data pr-eprocessing, followed by the steps detailing force pre-processing. The class structures used for data structuring within the GVis were developed using Python object-oriented programming. The filtering process removes unwanted signal frequency bands that do not meet the criteria of the defined threshold. The results from this process will be signals that are directly related to the grasp force.

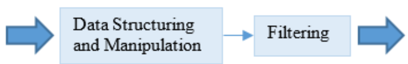


Figure 4.11: Summary of force data pre-processing methodology.

Step 1 The force data contained both calibrated force and raw analogue outputs. The raw analogue data was used to verify the correct mapping (refer to Figure 4.9), after which it was discarded from the dataset.

Step 2 The force data were batch processed using a high-pass filter, with a cutoff frequency of 20Hz. By filtering the data, dynamic elements due to the grasping action were attenuated, strictly limiting the relationship to be between sEMG and grip force.

Step 3 Manually initialising the recording of the force and sEMG data induced lag between both signals. Thus, the data was adjusted on an individual basis to realign the signals with one another.

4.3.2.3 Feature extraction

Feature extraction is performed using MATLAB programming software and is considered in this section. The features described in section 3.1.3 are applied to an sEMG signal, thus extracting time-based features linked to MUAP amplitude. The MNF-based features discussed in section 4.1.3 were applied to the force. These frequency-based features are linked to the MU firing rate. The method of feature extraction is coupled with sliding windows and also Min-Max normalised to the MVC to improve classifier robustness. Also, the data is up-sampled using the linear interpolation technique. This ensures that signal sampling rates are coherent and synchronised with each other.

In odd cases where the maximum feature values from the MVC data were superseded by the maximum feature values from the sEMG signal and force data, the maximum feature value from the respective signal was used as the normalising divisor. Figure 4.12 shows a summary of the feature extraction methodology, which is expanded on in Step 1 to Step 3 below.

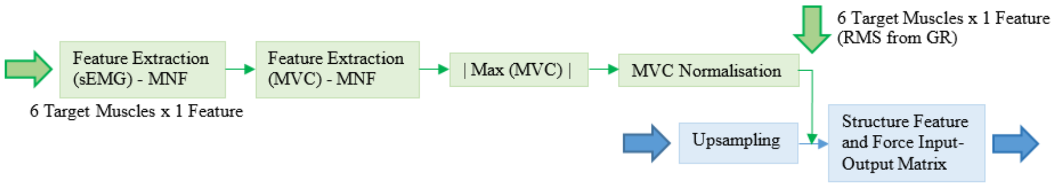


Figure 4.12: Summary of force data for feature extraction methodology.

Step 1 The disparity in sampling rate between the sEMG and force data was corrected by employing the same sliding window process described in Algorithm 4, while maintaining a window and overlap length of 80ms and 20ms, respectively. This process, conducted in MATLAB, was initiated on the data from each FSR separately, with a MAV feature calculated for each window.

Step 2 Temporally, the MAV force features were averaged across all FSRs, giving a global indication of the grip force and forming the output force vector.

Step 3 Combining the output force vector with the sEMG RMS and MNF features previously attained, resulted in an input-output matrix from which the regression model was trained.

4.3.3 Regression

In this section, several regression models that map the sEMG signal to the force are trained using MATLAB and compared. This process uses the input-output matrix obtained from section 4.3.2.3 Step 3. The regression models are trained in such a way that the force is the target dependent variable and the sEMG signal as being the independent or predictor variable.

Figure 4.13 shows a summary of the regression modelling process that was followed in this dissertation. The linear, regression trees, Gaussian process, SVN and ANN models were all trained using the feature extraction matrix. The models are individually trained until they meet the MSE requirements. Once the model's MSE is reached, its results are recorded and plotted for comparison with the other models. From the results plot, the best regression model would be selected based on performance and will be used as the gripcompensation component of the RPH.

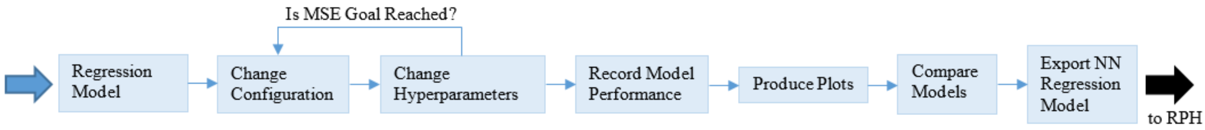


Figure 4.13: Summary of force data for the regression methodology.

4.4 Results and Discussion

In this section, the findings stemming from the development of the sEMG-force regression model are dissected. The findings are contingent on the methodology described in Section 4.3 and split into data acquisition, pre-processing, feature extraction and regression. The correlation of two features (RMS and MNF) and the grasp force is assessed separately. Furthermore, the degree of linearity between the sEMG-based features and grasp force is analysed. Within regression, several models are investigated and compared to an NN-based model. Finally, a Simulink model is presented for real-time use in the RPH control system.

4.4.1 Data acquisition

To develop a feasible sEMG-force regression model, it is imperative, to initially analyse the raw data and make judgements accordingly. In this section, two data sources are considered; however, greater significance is placed on the unestablished force data acquisition results. Also, to formulate a unified sEMG-force model, the data was directed towards principally investigating the dissimilarities between grasps and not participants.

The first source of data gathered through the sEMG data acquisition process (covered in Section 4.3.2.1) and applied to this chapter, yielded raw data with similar offset and noise characteristics, highlighted in Section v (see Figure 4.14). The sEMG signals were observed to oscillate about the offset, residing in both the positive and negative domain. During periods of sub-maximal isometric contraction (grasping), the raw sEMG signals were noticeably more erratic and pronounced.

The sEMG data acquisition process yielded 30 data files, each amounting to 20s worth of data per participant and grasp. To clarify; six grasps were investigated, with rest being excluded due to the ineffectual associated force, for the five participants concerned. With a sampling rate of 1926Hz, each 20s data packet amounted to 38520 individual samples per target muscle. Also, the samples were reflective of potential differences, and the inertial measurements were disregarded. In preparation of the raw sEMG data, the sEMG data was collated for each grasp, to test inter-grasp sEMG-force linearity. The second source of data, which was gathered concurrently with sEMG data, was that of force data. Following the methodology outlined in Section 4.3.2.1; the touch-sensitive glove produced raw force data which indicated the level of grasp force being applied by the participant.

An important consideration was maintaining contact between the FSRs and the GO in question. As such, it was observed during the trials that the ink test accurately indicated the most effective FSR locations. However, these locations were difficult to maintain due to differences in the hand dimensions between participants. The placement of FSRs on the touch-sensitive glove was then manually adjusted after each trial to resolve this.

Another important consideration was the variable deformation of the FSRs, induced by the participant's hand. The deformation occurred in the tail portion of the FSR and was accounted for through an offset calibration in the GVis application. The offset was observed to fluctuate between 5mV and 100mV and may be correlated to the size of the participant's hand, that is larger hands tend to deform the FSRs more, consequently increasing the required offset.

The force data was made possible by calibrating the digitised FSR outputs according to the calibration curves shown in Figure 4.9. The purpose of FSR specific calibration was demonstrated in the consistency of accuracy across all FSRs, principally illustrating the intended impartiality to the added coupling disks. Because of the FSRs lack of definition above 10N (that is the calibration curve plateauing), a limitation is reserved for the valid range of forces. Furthermore, the force data is limited to a smooth increasing and decreasing (fused tetanus), fixed muscle length (isometric), submaximal stimulus and contraction of forearm muscles for the identified grasps. Though it should be noted that no contraction truly exhibits an isometric behaviour and that muscle length variation will occur to some degree [180].

Spectral analysis validated the prominent low-frequency aspects associated with the forces produced from grasping (0hz to 20Hz) [181]. As such, the sampling rate of the touch-sensitive glove (41,5Hz) adequately captured the dynamic forces inherent in the investigated grasps. However, it is pronounced that a faster sampling rate will reduce the interpolation-related intervention required for upsampling. Faster sampling rates were limited by the capabilities of the Arduino Mega 2560 PRO MINI, associated stability delays and the delays attributable to the AnalogRead function. Another interesting fact surrounds the allocation of timestamps to each sample. Accordingly, it was presumed that a timestamp allocated by the GVis application would reduce the computational requirements of Arduino. Albeit a correct assessment, background processes of the processing system were found to introduce dynamic time warping. Thus, it is recommended that the introduction of timestamps be dealt with on the real-time Arduino system. The structure of the output force data summated to a ten column matrix, where the first column captured time entries of time and each succeeding columns represented the forces corresponding to the FSRs. In addition, the matrix was 24900 rows in length, with each row representing a sample.

By visually comparing the force data with the sEMG data, it was observed that a significant delay was present on a per-trial basis. This shows strong evidence that the manual cueing of data capturing introduced a delay between the two data streams. However, a portion of the delay is attributable to the inherent leading of the sEMG signal, that is the signal leads the action. This delay is commonly referred to as electro-mechanical delay and is suggested to be attributed to serial and parallel elements present within the muscle, dampening the contraction velocity [182]. The electro-mechanical delay is reported to be in the range of 25ms to 75ms, measured from the onset of contraction.

An example of the lag observed in the trials is shown in Figure 4.14. Here the force trails the sEMG signal by an average of 0.65s. Among other processes, the delay was addressed during the feature extraction step, where peak to peak alignment was performed. Also, shown in Figure 4.14 is the apparent offset of the sEMG sensor (noted in Section 3.4.1) and the visually verifiable link between sEMG amplitude and force.

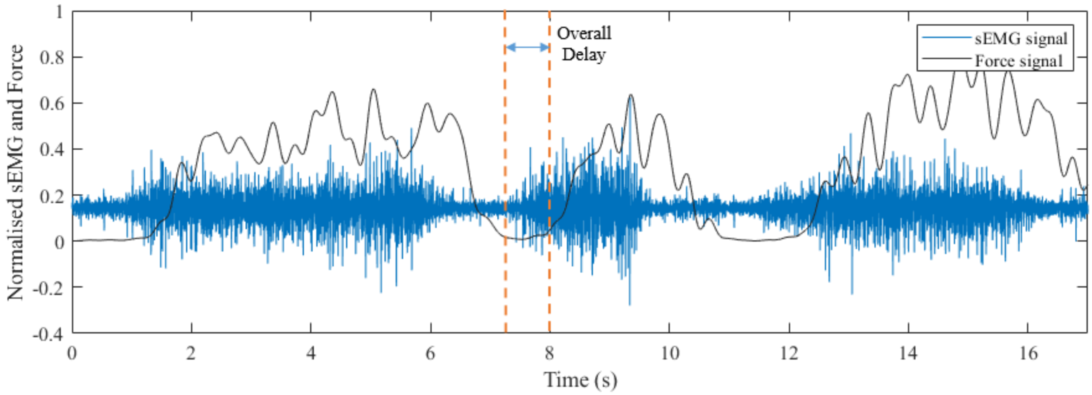


Figure 4.14: Normalised raw sEMG and force signals.

4.4.2 Pre-processing

The pre-processing of the sEMG signals was split due to the requirements in the construction of each feature. Firstly, for RMS, the process was kept mostly the same as that described in Section 3.1.2; therefore, the results are similar. In summary, the process involved a combination of data structuring, low-pass, high-pass and notch filters. Full-wave rectification is inherent in the RMS equation (see Equation 3.1.1) and was therefore omitted from the process. Also, it was found that PL interference remained the same as that discussed in Section 3.4.2; however, changes were perceived in the harmonics and background interference. The interference resulted in the notch filters being shifted to 78.20 Hz and 226.32 Hz (an increased difference of 4.12Hz) while similar notch strengths were maintained. The SNR of the sEMG data acquired in this chapter was visually identical to that acquired in Chapter 3.

Furthermore, no quantifiable assessments of the SNR could be made. Secondly, for MNF, the sEMG data was kept untouched, since it was observed that applying the same low-pass, high-pass and notch filters as before, resulted in the degradation of the MNF output and resembled white Gaussian noise, with no observed trend. Contrary to being neglected in classification, the sEMG sensor offset was removed by detrending the data (both RMS and MNF). This ensured that there was no offset in the regression fits.

To present a unified mapping of the APs and firing rate of the target muscle to the grasp force; the sEMG signals were averaged for each sample. That is to say, that any discretely sampled data point represents the averaged activity of the muscles investigated. It should be noted that the summation of sEMG signals is equally viable as an indicator of activity. Likewise, the force data were averaged, meaning the forces of the nine FSRs were averaged for each sample. As previously noted, this was critical in attaining a general overall representation of the grasp force initiated by the participants.

Due to the disparity in sampling rates between the sEMG and force data (with the former being 46 times larger to the latter), it was necessary to employ upsampling of the force data. Upsampling was achieved through linear-interpolation, where artificial sample points were uniformly introduced along the gradient between any two consecutive data points (as required). The resultant from upsampling, the data length increased to 38520 samples per 20s trial, ensuring the sEMG and force data lengths were the same after window feature extraction. Furthermore, the resolution and anti-aliasing properties were improved. As exemplified in Figure 4.14, the force data was noticeably free from high-frequency noise, the cause of which may be attributed to the low sampling rate. As such, the force data was not filtered, thereby avoiding the low-pass associated lag.

4.4.3 Feature extraction

Extracting features from sEMG data increases the information density and grants insight into the relationship present between characteristics of interest. In this dissertation, two features, namely RMS and MNF, were chosen due to their suitability in defining the recruitment (indirectly through the sEMG amplitude) and firing rate of MUs. In Section 4.3.2.3 features in the time, frequency and time-frequency domain are mathematically computed for a given window and overlap length. A sample of the results from RMS feature extraction is shown in Figure 4.15. From these results, it was observed that the offset reduced the baseline RMS. Also, the results are normalised to participant-specific MVC data. This is particularly important to remove inter-participant variance.

For example, before normalisation of the RMS feature, a one-way analysis of variance (ANOVA) indicated that there was a significant difference between at least one of the participants $p<0.001$ ($\alpha = 0.05$). At the same time, after employing normalisation, there was little significant difference ($P>0.05$). Furthermore, it was observed that there were differences in the peak values between grasps, which seemed to scale according to the complexity of the associated grasp. A possible explanation of this may be due to the averaging of sEMG signals across the six forearm muscles and the fact that they have different fast and slow-twitch fibre compositions [180]. Further details on the process taken to mitigate the effects are given in Section 3.4.3, where the output of the sEMG-force model is adjusted depending on the classified grasp. Another critical assessment is that the RMS is directly correlated with the force applied during the grasping action.

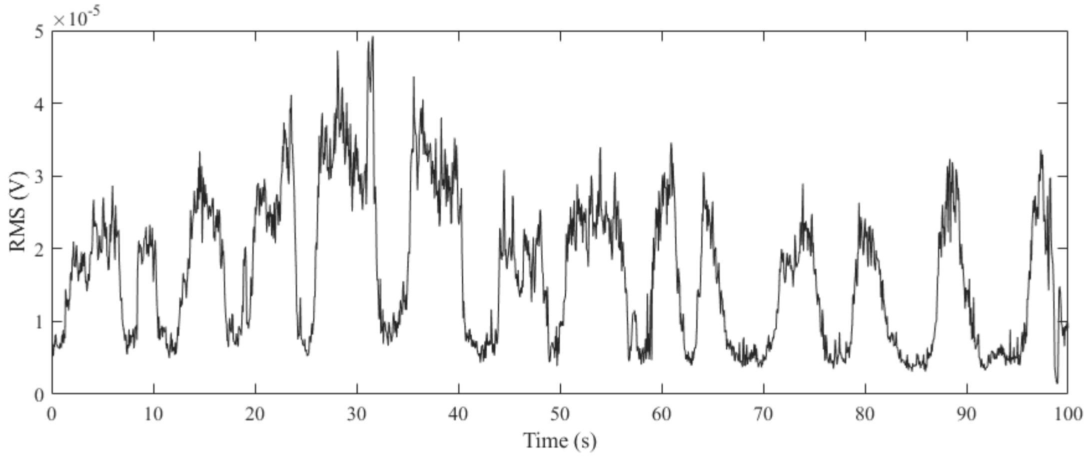


Figure 4.15: RMS feature extraction.

For MNF feature extraction, a similar process of events unfolded. This included computing the MNF with a window and overlap length specified previously concerning RMS. The MNF features must be extracted from the unfiltered, pre-processed sEMG data. Shown in Figure 4.16 is a sample of the MNF feature extracted from the sEMG data. In this figure, it is seen that the baseline MNF (30Hz, during rest) is approximately a third of the peak MNF (85Hz). The significance of the baseline MNF may be due to several reasons, including high-frequency noise; however, a more appropriate cause may be that the resting state of the hand is in slight flexion. Also, during the trials, gravity may have added to the activity of the muscles during a resting and/or set-position state. An increase in force is reflected as an increase in MNF. In Figure 4.16, the MNF is given in a normalised state due to reasons previously mentioned. Contrary to Chapter 3, the process of trimming each trial was not done. The reason being that any reflection of activity within the force domain (0N to 10N) was beneficial in constructing the regression model.

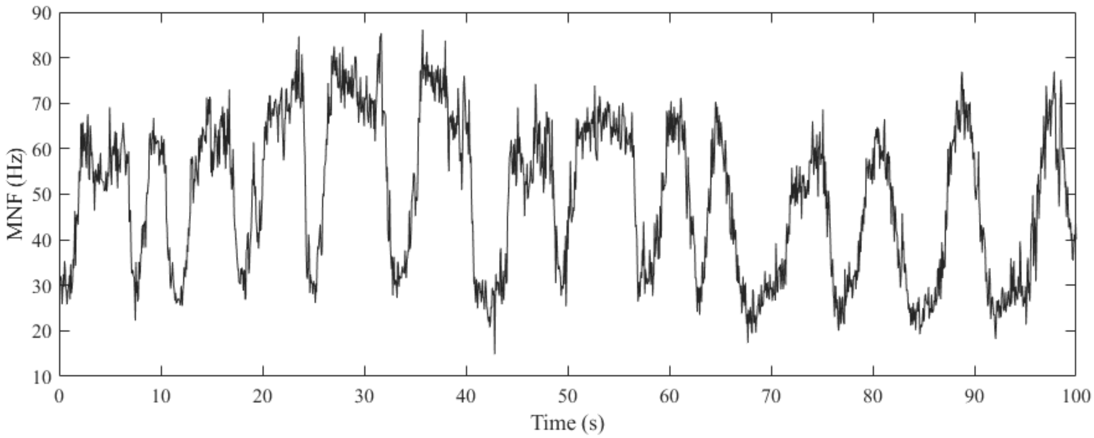


Figure 4.16: MNF feature extraction of sEMG data.

Brought to attention in Section 4.4.1; a delay was present between the sEMG and force data. It was necessary to address the delay since regression cannot account for past values and is driven by the error at any given time segment. Moreover, in this chapter, the aim is to define the relationship between sEMG AP amplitude and grasp force, and sEMG AP firing rate and grasp force, and not develop a transient forecasting model. As mentioned, the delay was primarily due to the methodology and therefore, if incorporated, would lead to the incorrect development of the sEMG-force model.

Also, neglecting to remove the electro-mechanical delay would place undesirable constraints on the real-time capabilities of the RPH; therefore, the delay was entirely removed. Removal of the delay, post-feature extraction, started with the utilisation of a median moving average being applied to the RMS, MNF and force data. The window size was heuristically adjusted using a smoothing factor of 0.15. This ensured that periodic trends in the data and the impact of outliers was minimised. Furthermore, it resulted in data which was smooth and had distinct peaks and valleys. This was important since it meant that the corresponding peaks of the smooth RMS, MNF and force data could be aligned and the path effectively transferred to the original data. Put differently, the signal was segmented at each valley and the signals padded to align the peaks. Each segment was truncated to retain the same number of samples. The path transferred to the original data entails the padding amounts and locations, and the details of truncation. Using a method such as warping would unrealistically augment the data and introduce non-uniform sampling.

4.4.4 sEMG-force regression models

It is suggested that different hand grasps produce different sEMG-force relationships due to factors such as muscle length, moment arm and movement of the innervation zones during grasping [106]. Therefore, to present a unified sEMG-force model for the grasps investigated in this dissertation, the implications of such suppositions need to be investigated. In response, this section analysed the degree of linearity between the sEMG features and grasp force for each grasp, respectively. Then compares the findings with those of a combined grasp analysis. Subject to the findings, comparative regression models were developed based on the combined grasp data. The use of NNs in regression-based applications are perceived to yield promising prospects, particularly in the non-linear mapping of relationships [182]. In this dissertation, this perception was investigated with aims of achieving an acceptable real-time sEMG-force regression model for use in an RPH. Several comparative regression models were developed preceding the NN regression model. These models served as performance-related benchmarks and justified the need for NN trial hyper-parameter optimisation regression models.

4.4.4.1 sEMG-force linearity analysis

In presenting a unified sEMG-force regression model, one first needs to establish the degree of linearity present between the sEMG features and force for the specified grasps. Description of the linearity defines a point of reference, upon which the regression models developed in this chapter are based. Table 4.3 shows the linear regression fits for the investigated grasps. These fits do not include robustness techniques such as least absolute residuals or bi-squared weights since it may be erroneous to exclude the so-called outliers (defined according to a linear fit).

The fit is distinct for each sEMG feature and is specified in the relationship column. Also, the extent of linearity for the specified relationship and given grasp is indicated by the coefficient of determination. An equation representing the non-normalised linear regression fit is provided in the last column. Here, the dependent variable is the grasp force, while the independent variables are RMS and MNF. In Table 4.3, the relationship which resembles that of a linear relationship is signified with the highest R^2 measure.

Accordingly, this is found to be the RMS-force and MNF-force relationships of the lateral pinch. Conversely, the RMS-force and MNF-force relationships for the spherical grasp shows the least degree of linearity (reflected with a low R^2). An interesting point raised in Chapter 3, linking 3 jaw chuck and tip pinch, seems to be validated by the fact that the gradient and y-intercept of the RMS-force and MNF-force relationships are notably similar. This suggests that the same extrinsic forearm muscles were used to accomplish the grasps.

From the results, it is clear that cylindrical, hook and spherical grasp, which are all deemed power grasps, have the lowest linear sEMG-force correlations. This seems to indicate that there is a link between the type of grasp and the level of non-linearity observed between sEMG and force. This means that the grasps requiring multiple MUs result in a more non-linear sEMG-force relationship or the muscles involved in power grasp are non-linear. A possible explanation, which is based on the work by Raikova *et al.*, [183], is that a sustained power grasp leads to an asynchronous MU recruitment, which in turn produces a non-linear relationship due to constructive and destructive APs. The results for the linear regression of the combined grasps is given in the last two rows of Table 4.3.

The R^2 suggests that the unified sEMG-force model, for the various muscles and grasps is non-linear. Though an *ANOVA* of the R^2 measures between the grasps shows a significant difference ($\alpha < <0.05$), the combination of the grasps is required for a unified sEMG-force model. This will be accounted for in the non-linear models developed in the subsequent sections together with a scaling factor that accounts for the averaging of muscles and force sensors, respectively.

Table 4.3: Linearity between sEMG features and force.

Grasp	Relationship	R^2	Equation
3 Jaw Chuck Pinch	Force-RMS	0.8094	$Force = (8424) \times RMS - 0.3039$
	Force-MNF	0.7665	$Force = (0.0485) \times MNF - 1.3030$
Cylindrical Grasp	Force-RMS	0.6712	$Force = (3361) \times RMS + 0.0949$
	Force-MNF	0.6924	$Force = (0.0531) \times MNF - 2.1530$
Hook Grasp	Force-RMS	0.6044	$Force = (5408) \times RMS - 0.1642$
	Force-MNF	0.6278	$Force = (0.0366) \times MNF - 1.2280$
Lateral Pinch	Force-RMS	0.8839	$Force = (6803) \times RMS - 0.1722$
	Force-MNF	0.8240	$Force = (0.0383) \times MNF - 1.0110$
Spherical Grasp	Force-RMS	0.5228	$Force = (4580) \times RMS + 0.1842$
	Force-MNF	0.5140	$Force = (0.0341) \times MNF - 1.0230$
Tip Pinch	Force-RMS	0.7793	$Force = (8219) \times RMS - 0.3369$
	Force-MNF	0.7336	$Force = (0.0425) \times MNF - 1.1840$
Overall	Force-RMS	0.5902	$Force = (3998) \times RMS + 0.1978$
	Force-MNF	0.6215	$Force = (0.0354) \times MNF - 0.9616$

The overall RMS-force equation stated in Table 4.3, is shown with the line of best fit in Figure 4.17. Here, the RMS values for all grasps and participants are depicted, with corresponding force. From Figure 4.17, it is clear that the relationship is non-linear and that the data digresses from the linear fit, particularly for grasp forces exceeding 3.5N. The data shows that there is very little increase in the grasp force above an RMS value of 8x10^{-5}V, which is possible due to the saturation of the FSRs. Furthermore, the data is clustered towards the lower spectrum of RMS, reassuring that the contractions achieved were, by in large, confined to lower proportions of the MVC (at 15%; typical of submaximal skeletal muscle contractions).

In the lower RMS regions (from 0V to 5x10^{-5}V), the data seems to be more scattered with a trend coinciding with an exponential function with two terms. While, for the upper RMS regions (>5x10^{-5}V), the data is closer and more linearly distributed. The outliers improve generalisation and were, therefore kept. An example of the outliers is shown in the pocket of data encompassing minimal force and substantial RMS. Most importantly, an increase in RMS is perceived as an increase in force. In conclusion, Figure 4.17 shows that the RMS-force relationship is non-linear, but not to a significant degree.

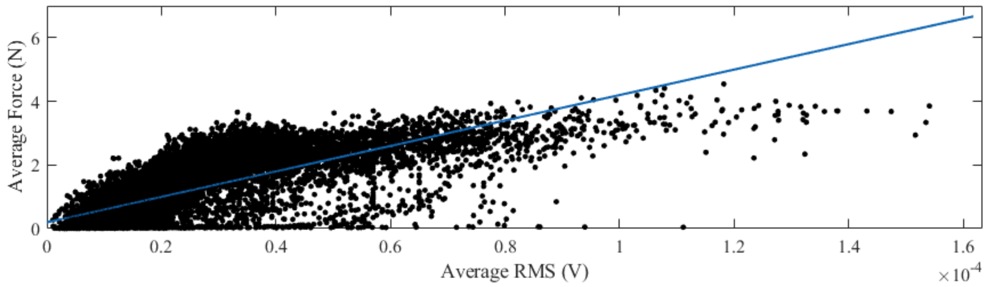


Figure 4.17: Relationship between RMS and force.

Similarly, the equation representing the linear regression of the MNF-force data is illustrated by the line of best fit in Figure 4.18. An interesting deduction stemming from the comparison of Figures 4.17 and 4.18 is that the MNF-force relationship is more linear than the RMS-force relationship. However, both relationships show non-linear trends comparable to an exponential function. In Figure 4.18, the cluster of data bounded by 45Hz, 55Hz, 2N and 4N, represent the outliers. As shown through multiple studies [184, 185], muscle fatigue is indicated by a shifting of the MNF signal to lower frequencies. Hence, it may be presumed that the outliers may be a reflection of fatigue. Since real-world applications may involve muscle fatigue, the outliers are kept and form part of the generalisation of the models going forward. At the upper bound of frequencies, the force lowers. Further research would need to be conducted to determine the cause of this anomaly.

A physical interpretation of Figure 4.18 sees the increase in MU firing rate resulting in a greater grasp force, that is an increase in MNF results in an increase in force. In comparing Figures 4.17 and 4.18, it is seen that Figure 4.18 is more scattered. This may be an indication that sEMG MNF is a feature which is more participant and grasp specific. Thus, the R^2 measure for the MNF force relationship of the overall grasp in Table 4.3 may not be used to substantiate whether or not an overall model is advisable.

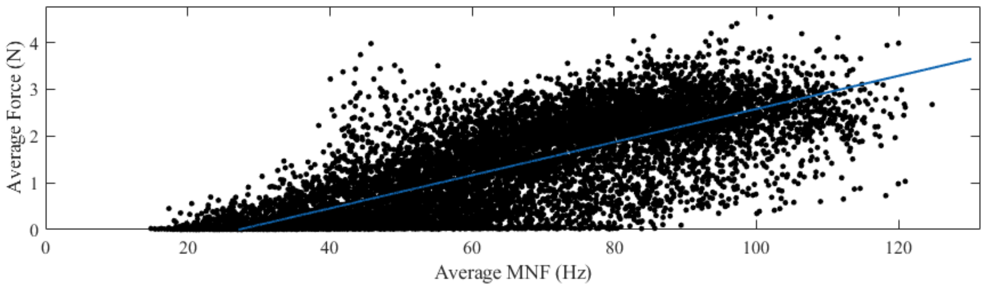


Figure 4.18: Relationship between MNF and force.

4.4.4.2 Comparative models

In order to justify the NN regression model implemented in this dissertation, several other comparative models were developed and used as performance-related benchmarks. These models predict the relationship between the sEMG and the force. The input-output relationship, analogous to a transfer function, gives an equation that maps the sEMG (independent variable) to the force (dependent variable). Table 4.4 shows the performance indicators of the different comparative regression models developed. These indicators are in the form of statistical good fit measures discussed in section 2.1.2.

From the results in Table 4.4, all the comparative regression models developed have an RMSE in the range 0.5-0.65. On average, the GPR models exhibit the lowest RMSE. This shows that this model better predicts the output variable than the other regression models. Similarly, due to the direct relationship between the RMSE and the MSE, this model also produces the lowest MSE. These results can be attributed to the ability of the GPR model to select covariance and hyper-parameters from the training data directly.

The developed models can also be compared based on their speed of observation and training time. Respectively, these performance parameters give an indication of which models have stronger processing capabilities, and which ones are computationally expensive. The regression tree (RT) models have the largest observation speed and the smallest training time. This is because, in a regression tree model, it is not necessary to input missing data, making it fast in observing data with missing values. Also, this model requires less analysis and coding, making it easy to train.

Table 4.4: Performance of regression models.

Model	**RMSE**	**MSE**	**R^2**	**SSE**	**MAE**	**Observation Speed (Obs/s)**	**Training Time (s)**
Linear	0.5922	0.3507	0.68	4115	0.4508	120000	4.37
Interactions Linear	0.5788	0.3350	0.70	3932	0.4375	530000	0.48
Robust Linear	0.5956	0.3547	0.68	4163	0.4476	640000	1.19
Stepwise Linear	0.5788	0.3350	0.70	3932	0.4375	620000	1.06
Coarse RT	0.5614	0.3152	0.71	3699	0.3972	1000000	0.33
Medium RT	0.5897	0.3476	0.68	4081	0.4174	990000	0.55
Fine RT	0.6482	0.4202	0.62	4931	0.4553	610000	2.98
Boosted Tree	0.5474	0.2996	0.73	3516	0.3982	160000	3.08
Bagged Tree	0.5556	0.3087	0.72	3623	0.3944	77000	3.64
Linear SVM	0.5957	0.3548	0.68	4164	0.4469	100000	102.16
Quadratic SVM	0.5816	0.3382	0.69	3969	0.4236	74000	502.22
Cubic SVM	0.6071	0.3686	0.66	4325	0.4352	70000	2348.20
Coarse Gaussian SVM	0.5683	0.3230	0.71	3790	0.4011	32000	17.73
Medium Gaussian SVM	0.5557	0.3088	0.72	3623	0.3836	47000	18.04
Fine Gaussian SVM	0.5551	0.3081	0.72	3616	0.3847	47000	21.65
Rational Quadratic GPR	0.5412	0.2930	0.73	3438	0.3857	5000	183.58
Exponential GPR	0.5420	0.2938	0.73	3447	0.3859	7300	126.20
Squared Exponential GPR	0.5430	0.2948	0.73	3460	0.3870	7900	123.47
Matern 5/2 GPR	0.5409	0.2926	0.73	3433	0.3856	4800	132.48

4.4.4.3 Neural network model

The NN-based regression model discussed in section 3.4.4.2 was trained to map the sEMG to force. This model was developed using the MATLAB framework and used as a prediction function. The prediction results were then compared to the results of the regression models discussed in the previous section. The training process of an NN model seeks to obtain the relationship between sEMG and force by varying the number of layers and neurons in the trained model.

Figure 4.19 shows the variation of force MSE with the number of hidden neurons in the first layer. From the graph, it can be observed that there is a general decrease in MSE with an increase in the number of hidden neurons. Increasing the number of neurons allows the model to predict the output variable in smaller packages. This is because, with more neurons, model flexibility increases as a result of having more mathematical calculations within the layer. This process, in turn, improves model accuracy, and thus cause a reduction in MSE.

It is also observed that as the number of hidden neurons in the first layer increases above 5, the MSE becomes appreciably constant. This can be attributed to model over-fitting, which results in a decrease in the generalisation capacity of the regression model. Furthermore, it should be noted that having a large number of neurons in the single-layer causes the model to be computationally expensive as a result of having to do a large number of calculations.

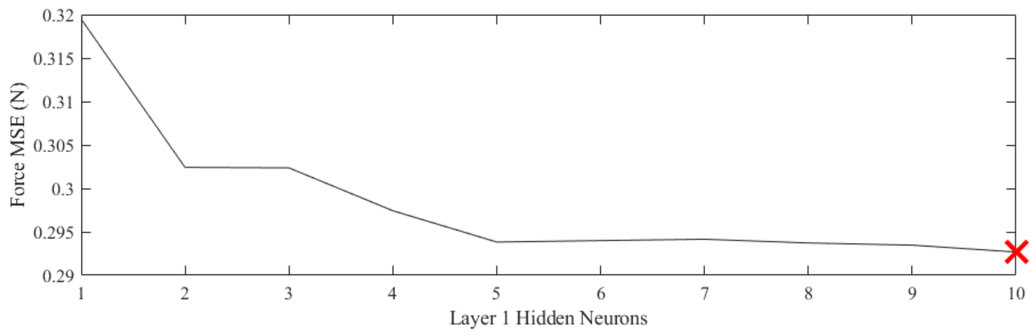


Figure 4.19: Variation of force-MSE with first layer hidden neurons.

As previously explained, NN models can also have multiple layers and neurons. These multiple layer models are usually implemented to improve model accuracies as they can recognise different aspects of the input data. Figure 4.20 shows a surface plot of force MSE as a function of the number of hidden neurons in the first and second layers of the NN model. It can be seen that as the number of neurons in both the first and second layers increases, the MSE tends to plateau around 0.295. This is similar to the trend observed in Figure 4.19, which shows the effect of increasing layer one neurons on the MSE and follows the same explanation. The plateauing observed in Figure 4.20 can be attributed to the fact that for higher hidden neurons, the difference in training errors as the number of neurons increases approaches zero. This is analogous to obtaining the derivative of a polynomial curve for very small changes in the independent variable, that is to say:

$$lim_{x\to 0}\left(\frac{\partial y_1}{\partial x_1}\right) = lim_{x\to 0}\left(\frac{\partial y_2}{\partial x_2}\right) \tag{4.4.1}$$

x and y are respectively, the large values of the independent and dependent variables. From Figure 4.20, it can also be observed that this minimum MSE occurs when there are eight hidden neurons in both the first and second layers of the NN model.

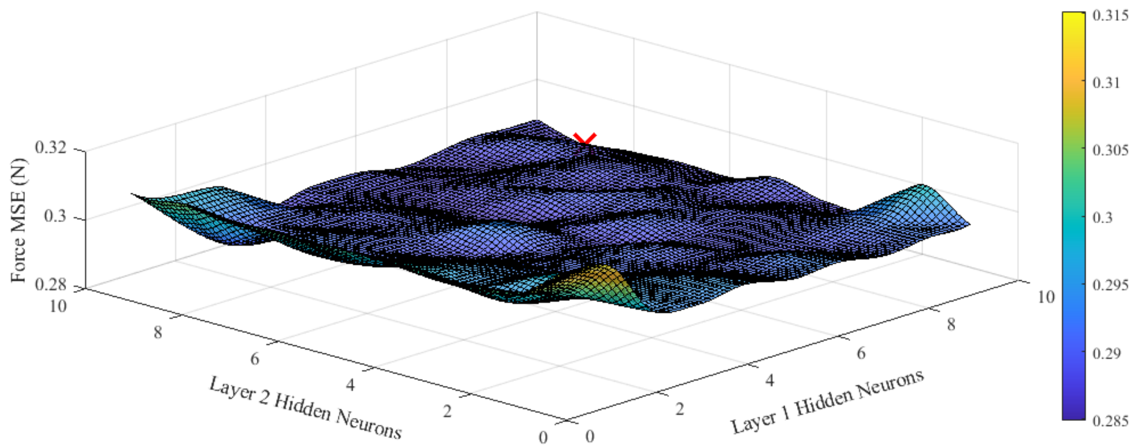


Figure 4.20: Surface plot of force-MSE as a function of number of hidden neurons, in the first and second layers of the NN model.

When performing the different grasping actions, the participants' muscle contraction levels vary from the start to the end of each process. These differing contraction levels are a result of the different commands from the central nervous system, causing changes in MNF. Figure 4.21 shows the variation of force with MNF. From the figure, it can be observed that there is a general increase in the predicted force as the MNF of both the true and predicted results increases. The reason for this trend is that as new MUs with high-frequency amplitude is recruited, the MNF will increase. This increase in MNF will, in turn, cause an increase in force. Also, the muscle fibre diameter increases with increasing MNF.

This increase in fibre diameter causes the muscle conduction velocity to increase, in turn increasing the force. This phenomenon was validated by Gerdle *et al.*, (1991) [186], in which they investigated the effect of force generation and muscle fibre type on MNF for knee extensors. Their study showed that there was a positive dependency of the MNF and signal amplitudes of the knee extensor to the force. It can be seen that the force prediction model was appreciably accurate when compared with the true results with very little variance in output predictions. This also justifies the selection and use of the NN model for RPH control. However, the observed differences between the predicted and true results can be attributed to the grasping actions being done by different participants. For the same MNF, muscle contractions between participants may be different, resulting in different force results.

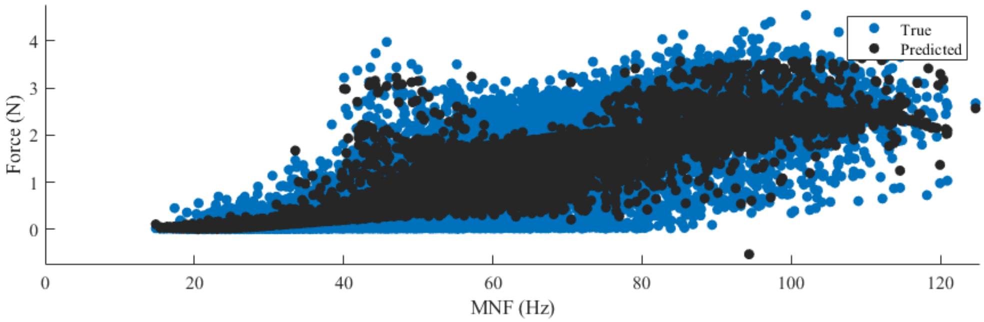


Figure 4.21: Relationship between MNF and force of the NN model.

Muscle activation level can also be used as a measure of the predicted muscle force. Muscle activation levels are analysed using RMS, which represents the time-domain signal power. Figure 4.22 shows the variation of the model predicted force with the RMS. From the figure, it can be seen that the predicted force increases with RMS. This can be attributed to an increase in both MU recruitment and the firing rate of currently active MUs. This neural activity, therefore, increases muscle signal power. A high-power signal causes greater muscle contraction, which then causes a commensurate increase in force. This phenomenon was validated by Cao *et al.*, (2016) [187] by examining the force load effects of sEMG signal features due to lateral arm movements. In their study, using *ANOVA*, they found out that the load force had a notable effect on RMS. Furthermore, from Figure 4.22, beyond RMS values of (>1x10^{-4}V), it can be observed that the force becomes independent of further increases in RMS.

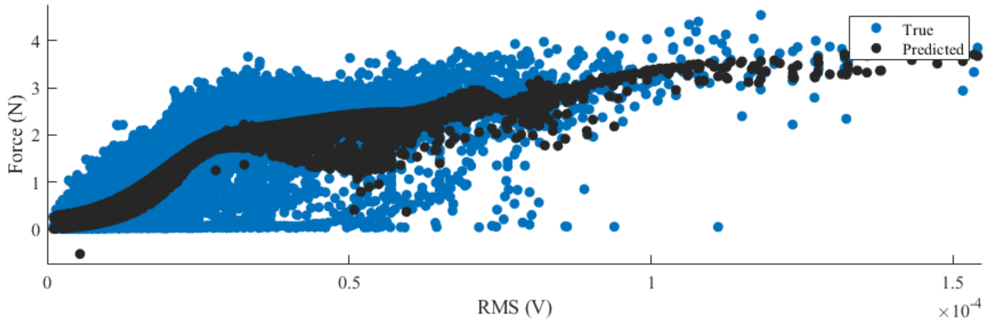


Figure 4.22: Relationship between RMS and force of the NN model.

4.4.5 Simulink model

The Simulink model used was based on the results obtained from the regression analysis. The chosen NN model was applied within the Simulink Deep Learning Toolbox and used to map sEMG signals (RMS, and MNF) to force. This NN model was trained in MATLAB and transferred to a Simulink block diagram using the *gensim* command.

4.5 Chapter Summary

In this chapter, different techniques used to estimate the force generated by muscle contraction were developed. Firstly, extensive research into the origin of muscle force generation was conducted. This assisted in understanding the type of neural activity occurring, and the resulting signal transferred to the appropriate muscle for contraction. A methodology for extracting skeletal muscle movement data resulting in hand movement was developed based on prior knowledge. A touch-sensitive glove together with a MATLAB developed grasp recognition software were used to measure output signals as a result of the different grasping tasks. Based on the recorded sEMG data and the resulting grasp force; various force prediction models (both linear and non-linear) were developed. These models were trained to map the sEMG signal (input) to the force (output) data, thereby providing a mathematical relationship between force and sEMG signals. The NN regression model provided the best force-sEMG signal relationship and was selected for the development of the Simulink model for use in the RPH model.

Chapter 5

Nonlinear Autoregressive with Exogeneous Excitation Neural Network System Identification of a Shape Memory Alloy-Actuator

Towards an Artificial Muscle for a Robotic Prosthetic Hand —

SMA actuators are increasingly being used as artificial muscles due to their exceptional strain and power density properties. However, non-linear phenomena such as the shape memory effect, pseudo-elasticity and hysteresis make it difficult to model these actuators accurately. In this chapter, a novel closed-loop NARX NN electro-mechanical model is developed using test-rig data to estimate the displacement of a stabilised SMA wire-based actuator. In comparison to analytical and Finite Element Analysis (FEA) models, the NARX NN model yields less error and better generalisation abilities. The inherent time-delay of the NARX NN model effectively captures the hysteretic behaviour, further substantiating its use for modelling SMA actuators.

5.1 Theory and Literature Review

Since nitinol was discovered in the early 1960s by Buehler *et al.*, [188], SMAs have influenced many fields of research including robotics, aerospace, structural and biomedical. Particularly, in the field of robotics, the use of SMAs has been geared towards the development of micro-actuators and artificial muscles [189]. In this chapter, the use of the latter is explored in actuating an RPH. Figure 5.1 presents a simplified illustration of an SMA-based artificial muscle, that is connected to the phalanges of an RPH finger. The highlighted region in Figure 5.1 defines the experimental scope of this chapter.

Specifically, modelling the displacement of an SMA actuator under the influence of various power and loading conditions. SMAs are a class of smart materials which recover their original shape in response to a stimulus. This recovery is exploited in an RPH to actuate flexion of the finger. Often it may not be feasible to test the implications of using different control parameters on the actual system. Therefore, analytical and computational models are developed to behave similarly.

Recently, it has been shown that computational models produce better results, possibly

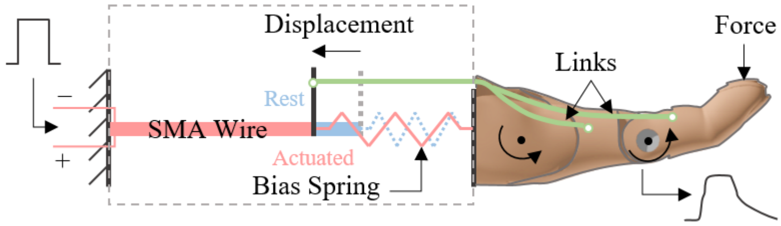


Figure 5.1: RPH finger actuation using an SMA wire and bias spring assembly. Resistive heating triggers a displacement from rest to an actuated state.

due to the lack of parameter estimation common in analytical models [190]. In this section, the principles of phase transformation and the non-linear mechanisms governing the displacement of NiTi (nickel-titanium composition) SMAs are discussed. Also, the types of SMA actuators, selection criteria and series and parallel NARX NN modelling of transient SMA systems are introduced.

5.1.1 Fundamentals of SMAs

As previously mentioned, SMAs undergo macroscopic recovery of large deformations. This occurs in response to either temperature or magnetic changes. In practice, SMAs which respond to temperature changes are far more commonly used, since they are arguably less susceptible to electromagnetic noise in robotic systems and do not require the use of bulky magnets or coils. Therefore, going forward only temperature-based SMAs, in particular NiTi alloys, will be discussed.

It is the microscopic behaviour of SMAs that give rise to the observed macroscopic behaviour. SMAs undergo a diffusionless thermo-elastic phase change from martensite to austenite. The former is thermally stable at low temperatures, while the latter is thermally stable at high temperatures. This is responsible for the memory and spring-like behaviour of SMAs [191]. In material science, the memory-like behaviour is referred to as the shape memory effect, while the spring-like behaviour is known as pseudo-elasticity. This phenomenon may be visualised in the example shown in Figure 5.2.

In Figure 5.2, there are four distinct states, namely A, B, C and D. In order, these states are no-load, load, heating and cooling. Considering a one-way SMA wire, from A to B, a load is applied which pseudo-elastically stretches the SMA. Heat is then introduced from B to C, changing the material phase and recovering the deformation through shape memory. Upon cooling from C to D, the load again stretches the SMA. Without a load, the process from C to D closes the hysteretic loop. These phenomena determine the rate of expansion or contraction of an SMA. In Figure 5.2, the terms ℓ_e and ℓ_{Res} denote the recovered and residual lengths, respectively and will be discussed later. In the subsequent sections, the shape memory effect, pseudo-elasticity, hysteresis and cycling are discussed. Also, the types of SMA actuators are reviewed with examples of operating conditions including rate of loading, composition, ambient temperature, stress and strain.

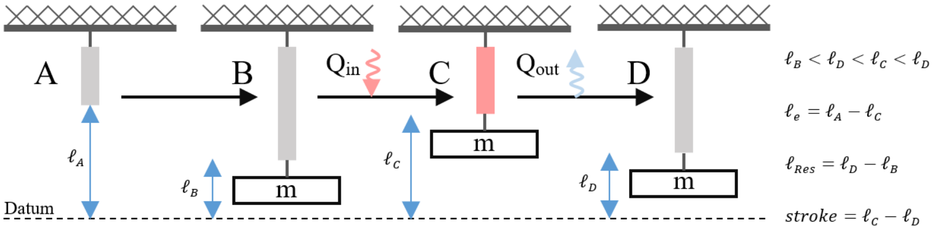
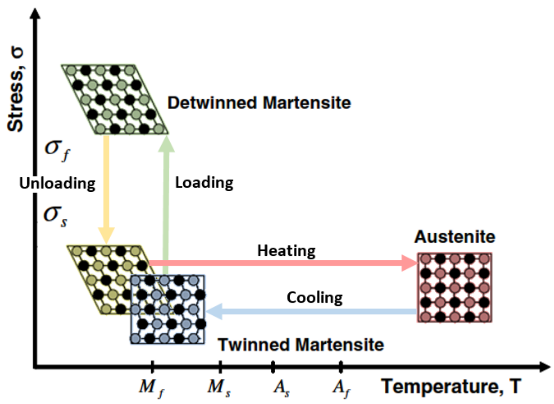


Figure 5.2: Stress-induced martensite formation and thermally-induced strain recovery in an SMA wire during initial cycling.

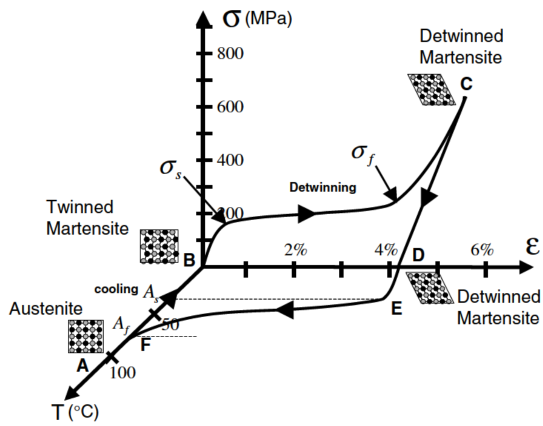
5.1.1.1 Shape memory effect of NiTi alloys

Metals or metallic alloys are ordinarily comprised of crystal lattices such as Face Centred Cubic (FCC), Body Centred Cubic (BCC) or Hexagonal Close Packed (HCP) crystals. When these crystal structures are deformed, defects within the crystal structure known as dislocations, propagate along the slip plane in the direction of the shortest lattice translation vector (slip direction). This causes irreversible changes in the order of the atoms within the lattice and differentiates common metals or metallic alloys from SMAs.

One of the reasons why SMAs remain intriguing is due to the phenomenon known as the Shape Memory Effect (SME). This effect considers the reversibility or rather memory of an alloy after being elastically deformed. The SME may be explained by considering the NiTi alloy stress-temperature phase diagram depicted in Figure 5.3a. Starting at A, the NiTi alloy is in a twinned martensitic phase. This phase is associated with a crystal lattice that is both symmetrical and consists of an ordered rhombus arrangement of atoms. Moreover, this arrangement is only possible at temperatures below the final martensitic forming temperature (M_f).



(a) Phase diagram.



(b) Phase transformation process.

Figure 5.3: NiTi stress-temperature diagram. [192]

From A to B, a process of loading takes place, in which stresses such as axial, bending, torsion and shear, cause plastic deformations in the crystal lattice. The de-twinning process is initiated when the combined induced stress exceeds the transition stress (σ_s) and is completed at the de-twinning stress (σ_f). Unlike common metals or metallic alloys, the structure remains ordered after being loaded. Specifically, any change in a region of the lattice structure is mimicked by the entire lattice.

After being unloaded in the transition from B to C, the NiTi alloy regains some of the plastic deformation equivalent to the elastic modulus, while retaining the de-twinned martensite structure. In instances of axial unloading, the NiTi alloy typically maintains between 2% to 6% elongation. At point C, all SMAs are in a pre-actuation step, where external influences such as temperature or magnetism may trigger a response.

NiTi is a thermo-responsive alloy; therefore, the process of heating which occurs from C to D, results in the transition to austenite, also known as the reverse phase transformation. Austenite is slightly denser in comparison to martensite, which means that there is a relative change in volume during the transition. The volumetric change is beneficial in cases where contraction is the main mechanism of actuation but leads to a reduction in stroke when expansion is considered. Heating beyond the austenite starting temperature (A_s), causes the de-twinned martensite to re-orientate into the pre-programmed geometry. This memory of the original geometry is the crux of the SME and is exploited in the creation of SMA actuators. Consider, for example, a NiTi alloy wire that has been plastically elongated. When heated, the wire wants to return to the original length, in turn creating a contractile force. The complete transition to the austenitic phase occurs at the final austenitic temperature (A_f).

Finally (from D to A) the NiTi alloy is allowed to cool. Upon reaching the martensite starting temperature (M_s), the austenite starts transforming into one of the variants of twinned martensite. There are 24 NiTi variants, and the choice variant is subject to the preservation of the macroscopic geometry of the alloy, that is the dimensions of the NiTi alloy are kept constant. The complete formation of twinned martensite occurs at M_f.

The graphical interpretation of the processes is shown in Figure 5.3b. Notably, the shape memory effect is depicted in the inelastic recovery of strain from E to F. In reality, the conditions of isothermal loading and isobaric cooling, shown in Figure 5.3b, are hardly seen. Instead of loading, heating, cooling and unloading occur in superposition to one another. Two other points worth noting are the thermal expansion and latent heat taking place during heating and the effects of loads and composition on the phase start and final temperatures.

Looking at the stress-strain-temperature phase diagram shown in Figure 5.3b, the first point may be partly noted as a non-zero strain at F. The effect that load has on the transition temperatures corresponds with an increase in load to a proportional increase in the transition temperatures. For no-load cases, the mass fractions of nickel and titanium are said to influence the transition temperatures. Frenzel *et al.*, [193] showed through the use of scanning calorimetry and density functional theory that an increase in the mass fraction of nickel seems to decrease the activation temperatures. Furthermore, Frenzel *et al.*, touched on the analysis of latent heat and concluded that similar twinned martensite and austenite crystal lattices yielded less latent heat.

To sum up, the SME is an integral phenomenon that sets SMAs apart from common metals or metallic alloys. The memory-like property means that a stimulus like resistive heating may be used to induce a change in the geometry. This may be incorporated into actuators which behave like muscles and can subsequently drive the movement of an RPH.

5.1.1.2 Pseudo-elasticity of NiTi alloys

The pseudo-elastic phenomenon exhibited by SMAs is described by the ability to recover large strains under isothermal conditions above A_f through a process of phase transformation. This ability makes SMAs particularly useful in construction applications geared towards seismic vibration damping. More importantly, this phenomenon is what allows SMAs to achieve the strains necessary to actuate an RPH and mimic the behaviour of skeletal muscles without plastically deforming. A distinct difference between the SME and pseudo-elasticity is highlighted by choice of the independent variable. Specifically, in the case of the SME, it is the temperature, while in pseudo-elasticity it is the stress.

The exact mechanism behind pseudo-elasticity in SMAs may be illustrated with the aid of Figure 5.4. Depicted in Figure 5.4 is the stress-strain behaviour of an SMA under isothermal conditions above A_f at a no-load reference. Commencing from A the SMA is in a stable parent austenitic phase. Upon introducing a load, the austenite present at A undergoes elastic deformation proportional to the Austenitic Modulus (E_A) until the yield stress is reached. This corresponds to the process from A to B. The yield stress is signified with a sudden change in the stress-strain slope.

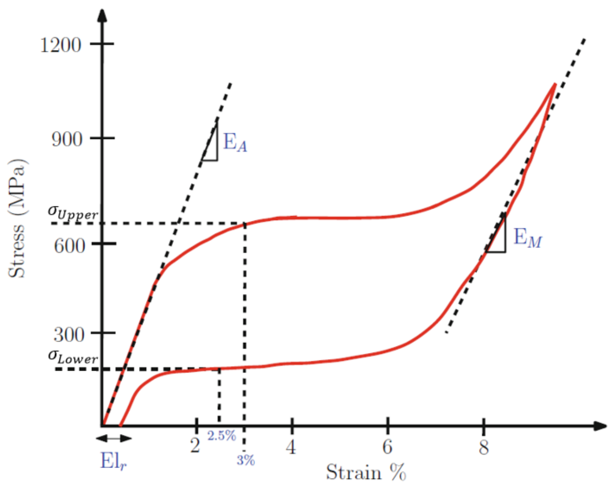


Figure 5.4: Stress-strain behaviour of an SMA under isothermal conditions [194].

The path from B to C addresses the pseudo-elastic behaviour of SMAs and is associated with a combination of super-elasticity and a rubber-like property [195]. In general, the rubber-like property corresponds to the reverse re-orientation of martensite and negligibly contributes to the overall strain; thus, pseudo-elasticity is sometimes regarded as super-elasticity. The pseudo-elasticity from B to C increases in inelastic strain, commonly in the vicinity of 6% for NiTi alloys, while the stress remains almost constant. The stress at which pseudo-elasticity occurs in the forward and reverse phase transformations is denoted in Figure 5.4 as σ_{upper} and σ_{lower}. These transformations shift to different stresses depending on the temperature (see Section 5.1.1.3). Upon further loading from B, the austenite starts becoming less thermodynamically stable. The austenitic crystal lattice augments itself with the inclusion of single variant martensite pockets to reach stability. This formation is also known as stress-induced martensite. The proportion of de-twinned martensite increases as the load increases and reaches complete de-twinned martensite at C, which is signified by a much steeper slope. Though, in reality, some regions of austenite are still present.

Ordinarily, SMAs are not loaded past C due to an increase in hysteresis and poor fatigue life. This not only makes SMAs less durable but also increases their non-linearity, which in turn affects the modelling of SMAs. Therefore, for the control of an RPH to be effective, the SMA needs to be limited to the stress at C. Furthermore, the path from C to D occurs due to an increase in the applied stress and results in the elongation of the de-twinned martensite. The theoretical recovery limit is found at D, any further loading beyond this point will result in plastic deformation and increased residual stresses.

Unloading from D to E results in the recovery of the elastic strain of the de-twinned martensite. The recovery rate is proportional to the martensitic modulus (E_M) and is substantially lower than the austenitic modulus. This results in possible larger strains for the NiTi alloy in the martensite phase. Šittner *et al.*, [196] showed through macroscopic stress-strain curves, x-ray diffraction and theoretical predictions that the austenitic modulus was almost twice that of the martensitic modulus. Though, it should be noted that the x-ray diffraction yielded some inconclusive outcomes.

Further unloading of the NiTi-based SMA (path E to F) results in the reverse transformation phase and the recovery of large strains. The recovery is linked to the linear increase in transition temperatures as the stress increases. This means that unloading from E lowers the transition temperature of austenite and therefore allows its formation. This path sees the transformation from a monoclinic B19' crystal structure to a rhombohedral cubic distorted austenitic phase (R-phase). At F, the stress-induced martensite is fully converted into austenite at the end of the elastic region.

The final step in the pseudo-elastic loading cycle takes place between F and A, where further unloading results in the elastic recovery of the martensitic phase. As shown in Figure 5.4 and denoted as El_r, in reality, the strain is never fully recovered due to residual strains stemming from the heterogeneous existence of both phases during the transitions. Another important factor to consider is the effect of temperature. In the previous discussion of pseudo-elasticity, it was taken that the temperature remained constant above A_f. However, as shown in Figure 5.4, the pseudo-elastic cycle changes as the temperature increases, and generally, the cyclic behaviour shifts towards lower strains and higher stress. Also, the effect of reprogramming the SMA becomes more apparent over prolonged use at higher temperatures.

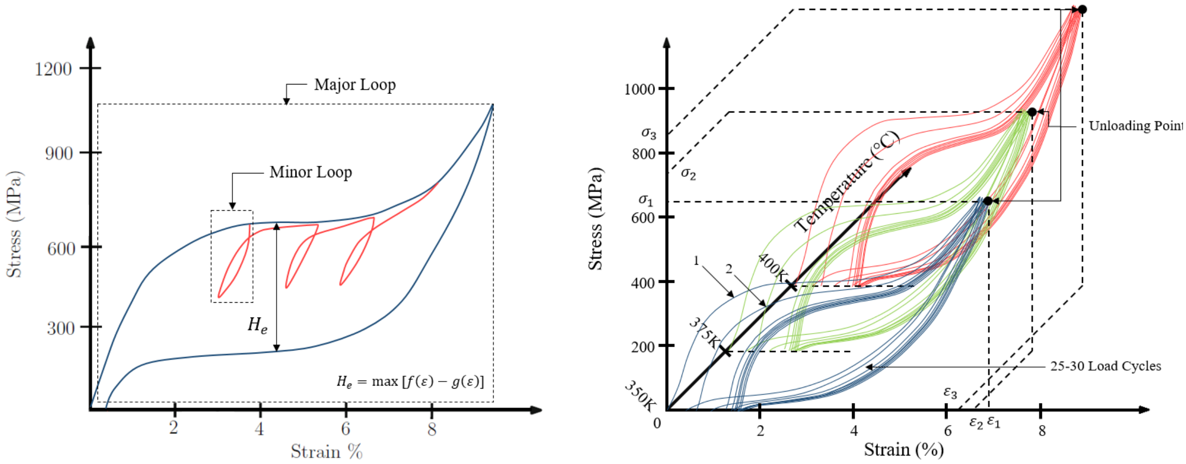
In conclusion, pseudo-elasticity is a highly important aspect of SMAs, which make them suitable for use in RPHs. For example, a 200mm NiTi wire with a maximum strain of 6% should yield a stroke of 12mm, which caters for the complete flexion of fingers from a set position. Nevertheless, the pseudo-elasticity of SMAs in isolation does not exhibit the full capabilities of an SMA and needs to be combined with the SME to provide meaningful actuation.

5.1.1.3 Cyclic and hysteretic response of SMAs

The cyclic behaviour of SMAs refers to the repetition of mechanical or thermal stimuli through a process of loading and unloading. This creates major and minor hysteretic loops depending on the load path taken. For example, in the major loop shown in Figure 5.5a, the SMA is driven to its maximum stress at a temperature above A_f. The region encompassed by the major loop defines the solution space, while the area remains an indication of the energy dissipated during the cycle (hysteresis). Considering Figure

5.5b, it is possible to observe the effects of load cycling. As the cycles increase, the deviation from the previous cycle decreases. Also, the cycle tends to shift towards higher starting strains. This is thought to be due to the accumulation of residual strains or defects present within the SMA over the number of cycles.

Important to notice in Figure 5.5b is the slowing down or saturation in the deviation of the stress-strain curve at higher cycles. This saturation corresponds to hysteretic stabilisation and is a significant factor which should be accounted for in SMA actuator modelling. Stabilisation may be affected by the processing history, loading conditions such as the rate of loading, composition and the number of loading cycles. Additionally, if the load path is repeated, the hysteretic stabilisation of an SMA results in the formation of a two-way SMA instead of a one-way SMA. It may be beneficial in the control of SMAs to reach hysteretic stabilisation. The reason being it reduces the uncertainty and non-linearity associated with tracking, while at the same time decreasing the energy dissipated, yielding a more efficient SMA. Nevertheless, in applications involving damping, SMAs with large hysteretic major loops are preferred since they are capable of getting rid of vibratory energy through super-elasticity.



(a) Major and minor hysteretic loops. **(b)** Effects of load cycling on hysteretic loops.

Figure 5.5: SMA stress-strain behaviour [197].

Hysteresis is a term used to describe the effect of system history on the current state of the system. Specifically, in the stress-strain diagram shown in Figure 5.5a, for any given strain, there are two corresponding stresses and vice versa. This indicates that SMAs are time and process-dependent, where the subsequent state is contingent on the previous states plus some change. Hysteresis is caused by the internal friction occurring at the martensite-austenite interface as well as the deformation of twinned martensite due to nucleation, rotation and shrinking. The hysteretic error is denoted in Figure 5.5a as H_e, represents the maximum difference between the reverse and forward phase transformation.

For the minor loop (Figure 5.5a), the SMA is loaded and unloaded such that only partial transformation from one phase to another occurs. These internal loops are typical of SMA actuation and have a parameter called return point memory (RPM). This parameter quantifies the difference between the minor and major loops, upon returning to the same loading path. In the book by Lagoudas *et al.*, [198], thermo-mechanical cycling is said to have the following influences on both major and minor hysteretic loops:

(1) The stress required to start the transformation between phases is lower.

(2) There is an increase in transformation hardening due to an increase in the dislocation density.

(3) There is a reduction in the hysteretic error.

(4) The net transformation strain is reduced.

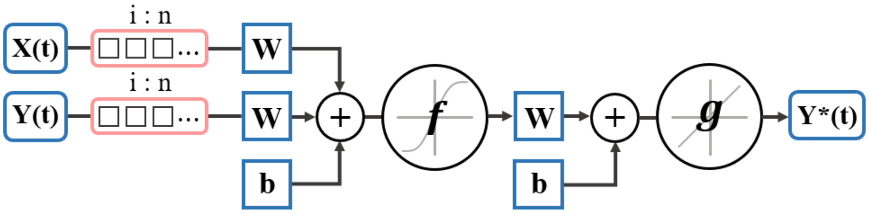
In summary, the coupled cycling and hysteretic behaviour of SMAs are highly important to the accurate modelling of SMAs. In general, SMAs display a variety of cyclic and hysteretic responses, which may be due to changes in the resistivity in the case of a wire SMA. However, the variation in these responses seems to decrease with increasing cycles. Therefore, in an application such as an RPH, it is advised that the SMA actuators undergo a breaking-in period, such that the behaviour of the actuator becomes more deterministic.

5.1.2 Time-series modelling with NARX NNs

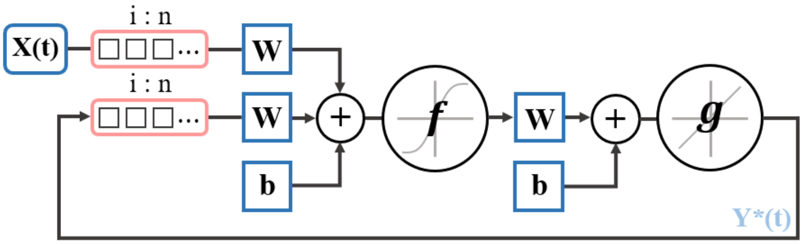
systems. This is because they are vital in ensuring the stability, disturbance rejection and noise attenuation properties of the system while undertaking a specific goal. Often the use of different controllers and tuning parameters cannot be directly imposed on the system due to the constant supervision requirements or the risk of damaging the system. Therefore, models are developed to simulate the behaviour of the plant and actuator, upon which different control, optimisation, supervision and fault detection structures can be tested [199]. Since the quality of the model determines the outcome of the final solution, it is favourable for the model to reflect the behaviour of the system directly.

System identification can be performed on either linear or non-linear systems, by use of either white, black or grey-box models. Due to the properties discussed in Sections 5.1.1.1, 5.1.1.2 and 5.1.1.3, SMAs are inherently non-linear and accordingly require non-linear modelling techniques. Also, hysteresis and cyclic-induced creep of SMAs leads to non-stationarities, which can only be account for through transient modelling. In this chapter, a NARX NN model is used to identify the dynamics of an SMA actuator, such that it may be used in the simulation-based design of an RPH control system. Consequently, this section deals with the theory and literature review of NARX NNs.

Similar to an RNN, a NARX NN seeks to introduce delayed feedback (tapped delay lines) to capture short and long-term dependencies. However, these networks differ from RNNs, such that the source of the feedback is the estimated output and not the internal state. In other words, current and past estimated output states are used for one-step-ahead output prediction. This means that hysteresis and the effects of cycling are retained in the time-delayed memory of the network in terms of an SMA actuator. NARX NNs also include delayed exogenous inputs which are correlated with the output and therefore have a significant impact on the model behaviour. An advantage of a NARX NN lies in the ease of training. This is related to the first of two modes of a NARX NN, which is discussed next. The first mode, shown in Figure 5.6a, is known as the series-parallel or open-loop mode.



(a) Open-loop mode.



(b) Closed-loop mode.

Figure 5.6: NARX NN model structure [200].

This mode defines the structure of a NARX NN during the initial training phase and in cases where the true output is known (by some sensor). An advantage of using this structure during training stems from the fact that it is representative of a feedforward network and therefore can be trained using common backpropagation techniques. Like a standard NN, the weights and biases are updated according to the gradient of the error between the true output and the estimated output. In Figure 5.6a, $X(t)$, $Y(t)$ and $Y^*(t)$ represent the exogenous input, true output and estimated output, respectively. The functions f and g are arbitrarily chosen activation functions; however, a ReLU activation function is most commonly used for the output. This choice of activation function ties in with the positive displacement output of an SMA actuator.

The second mode, shown in Figure 5.6b, represents the parallel or closed-loop mode of a NARX NN. Here, the next estimated output is based on the current and past values of the exogenous inputs and past values of the estimated output. In system identification, the parallel mode is typically used to define the black or grey-box system model, since the true output is unknown. Through a process similar to transfer learning, the updated weights and biases from the open-loop mode are used in the closed-loop mode. As noted, training in an open-loop mode is far more computationally efficient; however, using the weights and biases acquired through open-loop training in the closed-loop mode generally lowers the performance. Consequently, closed-loop modes are often retrained, using the open-loop parameters as starting points.

5.1.3 Past SMA actuator models

Various SMA actuator models have since been developed for robotic system applications. These different models are developed in order to optimise device performance. This is achieved by improving the device's intelligent control technique, rates of heating and cooling, and selecting the most suitable shape.

Liang and Rogers [201] developed a novel mathematical SMA force and displacement actuator model based on the constitutive thermomechanical relations. In this study, the transient response of both the bias spring and differential force actuators were analysed together with their hybrid systems. In these actuators, heating is an active process, and thus the cycle time is more of a function of the passive cooling process. Therefore, the differential force actuator showed a faster transient response due to the presence of the active reformation element. By further performing a numerical simulation of SMA actuators, they showed that the coupled analysis has a larger hysteresis than the decoupled analysis. Numerical simulations also proved that during phase transformation, the superposition principle could not be applied for as long as the material is made up of more than one phase.

Dutta *et al.*, [202] also developed a mathematical model for an electric current-driven SMA spring-based wire actuator. In this model, the working of the actuator was based on heat convection, electrical resistance, and stress-strain variations during the phase transformation process. Independent sub-models were developed for each phenomenon. The phase transformation process had a significant temperature hysteresis and therefore required a differential hysteresis model to depict both minor and major loops appropriately. This hysteresis model enabled the SMA actuator model's differential inverse to be obtained. Finally, a feedback control system with inverse feedback was proposed and validated using simulation results. By using an inverse compensator and PD feedback controller, tracking accuracy was improved. The major advantage of this model is that for any temperature profile, unlike the Preisach models, only four parameters are required to generate both minor and major loops.

Terriault, and Brailovski [203], integrated the simplified Likhachev's micromechanical model with ANSYS in order to simulate the system response of an SMA actuated morphing wing that is directly controlled by Joule heating. In their study, the Likhachev's formulation is integrated into an electro-thermal ANSYS model in order to simulate SMA actuator temporal response as a result of direct Joule heating. This study provided vital information such as heating and cooling rates, electric current, as well as the actuation force and stroke. This information was vital in the design of a suitable wing prototype.

5.1.4 Literary relevance and conclusion

The use of SMAs plays a significant role in the design of actuators for use in robotic systems. By taking advantage of their displacement response due to various loading conditions, the SMAs can be used in artificial muscles. The Pseudo-elastic and shape memory effect are the most important properties for SMAs that allows them to actuate muscle movement. The pseudo-elastic behaviour allows the SMA to achieve very large strains under the influence of a mechanical load. The SME allows the SMA to return to its original shape after undergoing elastic deformation due to a stimulus like resistive heating. These combined effects are taken advantage of in SMA actuators to accomplish an effective actuation. As these actuators are always in continuous use when in operation, it is of paramount importance to understand their cyclic behaviour so that error estimation and elimination methods are developed and implemented. Using ANN, the response of an SMA actuator can be modelled for effective control, optimisation, disturbance rejection and noise attenuation, whilst performing the intended task.

5.2 Experimental Facilities

Both the input power and output displacement were recorded using the constructed test-rig and advanced serial data logger. The input power was accessed through an INA219 current and voltage sensor. The sensor was connected via an I^2C bus to an Arduino, where embedded code queried the voltage and current across the shunt resistor (see Digital Appendix for code). The voltage and current were subsequently used to determine the power applied to the SMA actuator. An incremental optical encoder, in conjunction with an Arduino, was used to determine the displacement of the SMA actuator. Quadrature encoding meant that both the direction and position were accounted for and that the resolution could be scaled up by a factor of four. For a more detailed explanation of the instruments as mentioned earlier and procedures refer to Appendix 10.

5.3 Methodology

This section describes the methodology undertook in modelling the behaviour of the SMA wire. SMA response due to the various input signals was modelled using three different displacement prediction methods. The first approach used analytical methods to obtain the SMA response. In this method, constitutive mathematical equations were used to derive the SMA displacements as a result of Joule heating. The methodology followed during analytical modelling is presented in section 5.3.1. Next, an FEA approach was developed using ANSYS to model the response of the SMA using a coupled static-structural and steady-state thermal analysis. The FEA modelling procedure presented shows the material property definition and associated model boundary conditions. The results from FEA modelling enabled a visual interpretation of the effect of various input loads on the wire by viewing the graphical results in the ANSYS solution component. Lastly, a Simulink model was developed in the MATLAB environment to capture the response of the SMA under the same input conditions as the mathematical and FEA models. For the Simulink model, the step, pulse, pulse-train, and sinusoidal power inputs were used to capture the hysteretic behaviour of the SMA.

5.3.1 FEA Modelling

A three-dimensional FEA model of the SMA wire proposed by Auricchio [204, 205] was conducted using ANSYS workbench built-in library. This model was used to capture the behaviour of the SMA wire used under different loading conditions. In order to model the full behaviour of the SMA due to both the power and mass inputs, coupled static-structural and steady-state thermal models were implemented in ANSYS. The static structural model was used to obtain the macroscopic behaviour of the wire. This was then transferred to a steady-state thermal analysis to obtain the thermal stresses developed in the model as a result of resistive heating from the power input. The deformations that arise from these two processes in actual case give rise to the actuation capabilities of the SMA wire when used in an RPH. This section outlines the procedure adopted for the development of the FEA model.

5.3.1.1 3D CAD model and meshing

Before starting the simulation, the SMA wire geometry is developed in ANSYS Design modeller, and the model broken down into simple discrete elements. In these small elements, the constitutive mathematical equations are used to approximate the response of the wire due to an external disturbance Figure 5.7 shows the mesh details of the SMA wire model. The wire is 105mm long and has a diameter of 0,305mm. The mesh structure has 9647 elements, 50908 nodes, with the wire modelled as a solid (226) element. In order to accurately capture the gradients across the model structure, a uniform fine mesh with quadratic element order was selected.

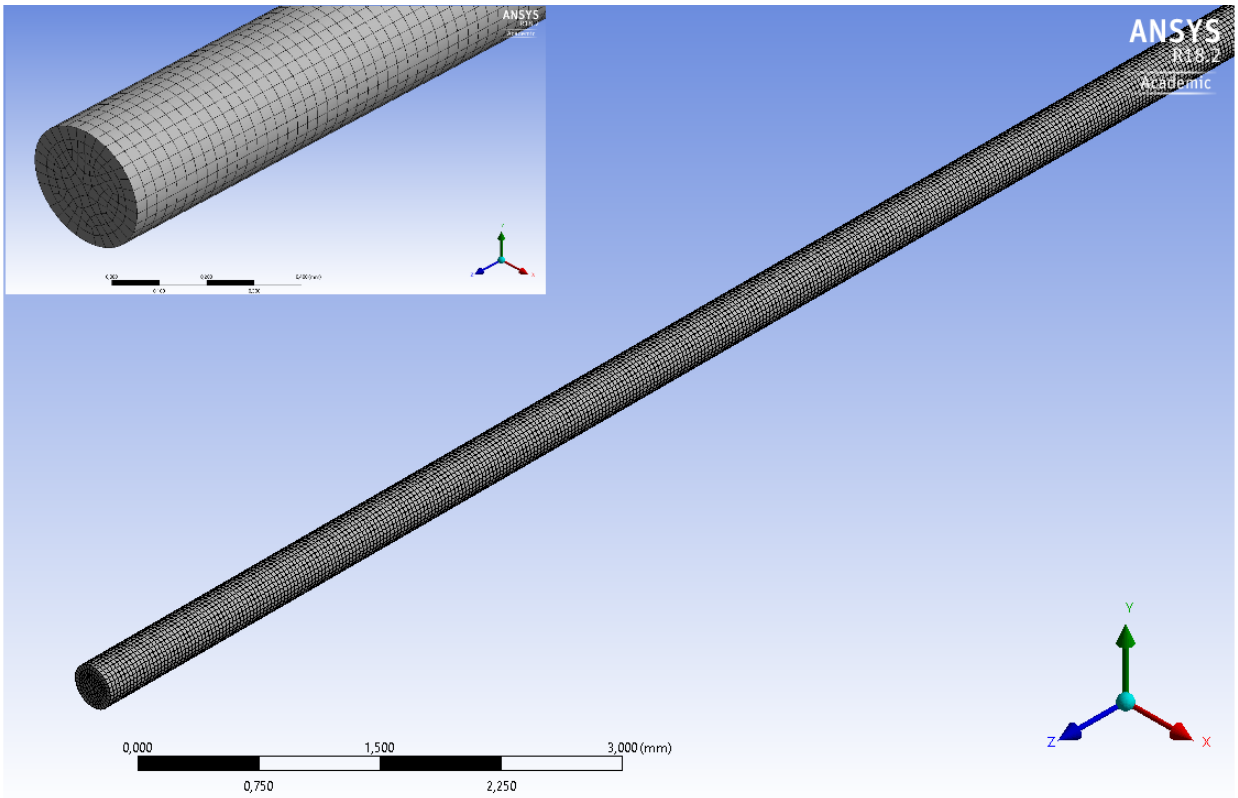


Figure 5.7: Mesh details of SMA wire model.

5.3.1.2 Material properties and boundary conditions

The material properties of the SMA were chosen based on the work done by Souza *et al.*, (1998-ANSYS docs). Their study investigated the effects of thermal stress on phase transformation with classical thermodynamics irreversibility. This model was used to derive the underlying features of SMAs undergoing multi-dimensional stress states. Table 5.1 shows the selected material properties of the SMA wire that were used in this study, initialised via the Engineering data material property command in ANSYS-Workbench. The constants in the table describe the stress-strain response of the SMA during cyclic thermal loading. Furthermore, the model boundary conditions used and inputs are shown in Figure 5.8. These boundary conditions impose constraints on certain locations of the model.

Table 5.1: Mechanical and thermal properties of the NiTi wire uses in the Ansys model [204].

Property (unit)	**Specimen A**
"Elastic modulus" of austenite (MPa) [205]	51700
Elastic modulus of martensite (MPa) [205]	42000
Poisson's ratio [206]	0.3
Hardening parameter, H (MPa) [205]	85
Elastic limit, R (MPa) [205]	192
Temperature scaling parameter, β ($MPa.°C^{-1}$) [205]	5.6
Maximum strain (mm/mm)	0.04
Loading Temperature, $T_{SIM} = M_s + \Delta T_1$,$T_R = M_f - \Delta T_{II}$ ($°C$)	45
Density ($kg.m^{-3}$) [209]	6450
Thermal conductivity ($W/m.°C$) [207]	18
Average heat capacity ($J/kg.°C$) [207]	320
Martensite finish temperature, M_f ($°C$) [205]	50
Martensite start temperature, M_s ($°C$) [205]	85
Austenite start temperature, A_s ($°C$) [205]	90
Austenite finish temperature, A_f ($°C$) [205]	115

Figure 5.8 shows the imposed boundary conditions at different locations on the SMA wire. The initial temperature of the wire was set to ambient (23°C). The convection coefficient was calculated using Equation 5.4.4 to enable heat dissipation from the wire. The macroscopic response of the wire due to the mass was modelled using transient structural analysis. This analysis was then coupled with a transient thermal analysis to include the resistive heating boundary conditions through coupled-filed physics. At this stage, it should be noted that in order to model the full shape memory effect of the wire, the structural and thermal loads were applied alternately. In order to accurately model the motion of the wire during the cycles, a remote displacement was imposed on one end of the SMA (Power input boundary). By doing this, the wire displacement would just be the calculated displacement of the surface on the opposite end (Force input boundary).

5.3.2 NARX NN SMA actuator model

5.3.2.1 Data acquisition and pre-processing

SMA data acquisition is a process in which the output displacement of the wire due to various input signals are measured and recorded. Pre-processing of the measured results is accomplished in order to remove any unwanted signal properties before the data is used in the training of the NARX NN model. Figure 5.9 shows a summary of the data acquisition and pre-processing flow diagram, followed by the detailed steps.

Step 1 Connect the SMA to the power input source, and make sure there is sufficient working space to allow SMA displacement in the different directions.

Step 2 Switch on the power input, and at the same time apply the mechanical load for about 5-10 s. While maintaining the mechanical load, quickly increase the power input in

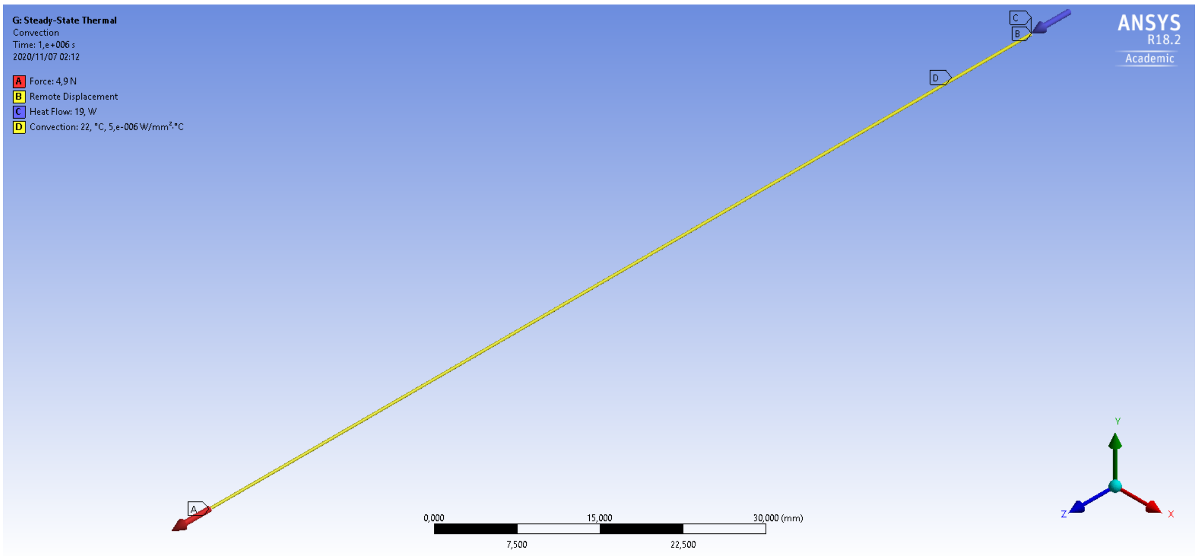


Figure 5.8: SMA wire model supports and boundary conditions.

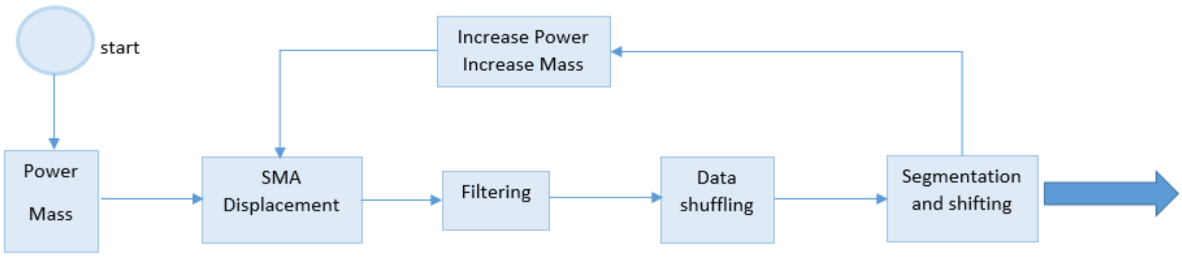


Figure 5.9: Data acquisition and pre-processing.

Step 3 Measure and record the SMA displacement using the Arduino and incremental encoder. Increase the power input by 1W up to 19W, and repeat Step 2 while maintaining the same mechanical load.

Step 4 Increase the mass by 100g until it reaches 1000g, and repeat Step 2 and Step 3.

Step 5 When all the data has been measured and recorded, a moving average filtering is applied to lower down and smooth out unwanted signal noises. Data shuffling is done to remove any periodicity that may cause poor model training. Lastly, segmentation and shifting of the data set is accomplished to remove negative or non-zero offset.

5.3.2.2 Model structure, training and optimisation

In this process, the structure of the model used in the training process was selected. This model was selected in the Simulink environment and assigned its respective parameters based on system requirements. The model was trained to predict the input-output relationship of the SMA using both the open and closed-loop systems. Figure 5.10 shows a summary of the steps followed during the model training and optimisation process.

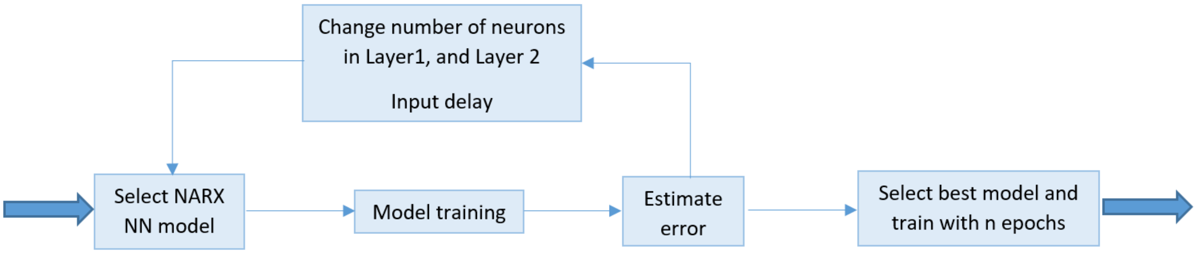


Figure 5.10: Model training and optimisation process.

5.3.2.3 Model validation

In this process, the performance of the selected neural network is investigated. This is accomplished by examining the system response due to excitation from various known input signals. The process is similar to section 5.3.2.1 but without any further pre-processing steps.

5.4 Results and Discussion

5.4.1 Analytical SMA actuator model

In this section, two of the most popular SMA actuator models are developed that mathematically describe the response of an SMA due to distinct input signals. The first model describes how the actuator displacement depends on the temperature as it is electrically activated. The developed model is based on the assumption that the only heat entering the system is from the input power source. The second analytical model describes the actuator response as it is thermally activated.

5.4.1.1 Thermo-electrical behaviour of a SMA wire

In this section, a thermo-electric analytical model of an SMA wire is presented. The model and concepts discussed will remain brief since the model is only used as a theoretical baseline, with the main focus attributed to the ML-based system identification of the SMA actuator. The thermo-electric behaviour of an SMA wire can be expressed in the form of a transient heat transfer model. This model relates the input energy stemming from resistance or Joule heating (that is, induction or an external heat source) to the temperature of the SMA wire over time. Moreover, the heat transfer model is used in conjunction with the thermo-mechanical model to create an analytical solution that describes the relationship between input energy, input stress and output strain.

Depicted in Figure 5.11 is the boundary of the heat transfer model. This is defined as a control volume and is taken around the SMA wire. The conservation of energy for the control volume is given in Equation 5.4.1, where E_{in} is the energy entering the control volume, E_{gen} is the internal energy generated, E_{out} is the energy leaving the control volume and E_{stored} the energy stored within the control volume. In this case, there is no energy transfer from the surroundings to the SMA wire; therefore, $E_{in} = 0$. In the model, internal heat is generated through a process of resistive heating which is

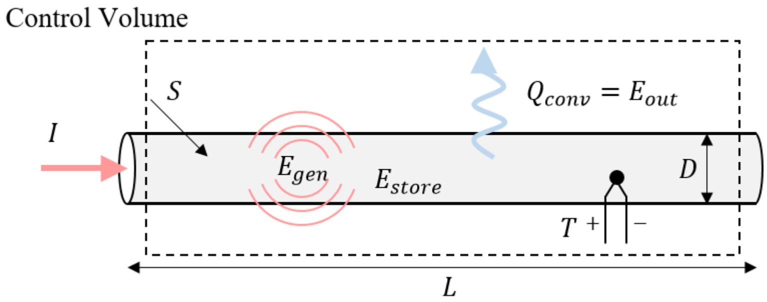


Figure 5.11: SMA heat transfer model and control volume.

governed by Ohms law and given by Equation 5.4.2, where I is the current and R is the resistance of the wire. For simplicity, it is assumed that the resistance of the wire remains constant. Though, in reality, the resistance decreases as the phase progresses from martensite to austenite [208]. This is due to the orderly crystal structure of austenite and the proximity of the atoms.

$$\cancelto{0}{E_{in}} + E_{gen} = E_{out} + E_{stored} \tag{5.4.1}$$
$$E_{gen} = I^2 \cdot R \tag{5.4.2}$$

There are three different modes of heat transfer, namely conduction, convection and radiation. For the SMA wire, dissipation of heat occurs primarily through convection since conduction and radiation are respectively limited by the cross-sectional surface area at the ends of the wire and the wire temperature (usually below 200°C). Therefore, in Equation 5.4.3 the energy leaving the system is described by Newton's law of cooling. In Equation 5.4.3, $\bar{h}$ is the mean convective coefficient for the wire, S is the surface area of the wire and T_w & T_∞ are the outer and bulk fluid temperatures, respectively. For horizontal cylinders in air, with laminar natural convection, it can be shown empirically that the mean convection coefficient is given by Equation 5.4.4. With the assumption of uniform heating ($D \ll L$), T_w represents the temperature of the wire, herewith referred to as T.

$$E_{out} = \bar{h} \cdot S \cdot (T_w - T_\infty) + \cancelto{0}{Q_{cond}} + \cancelto{0}{Q_{rad}} \tag{5.4.3}$$
$$\bar{h} = 1.32 \cdot \frac{\Delta T}{D}^{\frac{1}{4}} \tag{5.4.4}$$

The energy stored within the SMA wire can be broken down into two components, namely internal energy and phase transformation. The components are given in Equation 5.4.5 and are correspondingly represented by the terms in the right-hand side equation. m is the mass of the wire, c is the specific heat of the wire (assumed to be constant), and ΔH is the heat transformation constant. The transient terms include the rate of change of temperature and martensitic fraction (ξ) with respect to time. Substituting Equations 5.4.2, 5.4.3 and 5.4.5 into 5.4.1, results in Equation 5.4.6. This is the governing equation that relates the temperature of the SMA wire to the input current.

$$E_{stored} = m \cdot c \cdot \frac{\mathrm{d}T}{\mathrm{d}t} - m \cdot \Delta H \cdot \frac{\mathrm{d}\xi}{\mathrm{d}t} \tag{5.4.5}$$

$$I^2 \cdot R = \bar{h} \cdot S \cdot (T_w - T_\infty) + \frac{m}{\mathrm{d}t} \cdot (c \cdot \mathrm{d}T - m \cdot \Delta H \cdot \mathrm{d}\xi) \tag{5.4.6}$$

The only unknown left to determine is the martensitic fraction. Many studies have succeeded in describing the martensitic fraction as a function of temperature and stress. The most notable of these are derived by Tanaka [209, 210], Liang and Rogers [201], and Boyd and Lagoudas [211], where exponential, cosine and polynomial hardening rules for the phase transformation were used. In this dissertation, the heating and cooling hardening rules presented by Liang and Rogers were used. These rules are stated in Equation 5.4.7 and 5.4.8 for heating and cooling respectively. Furthermore, these rules include the variation in the martensitic fraction with varied stress, by including the terms $b_A \cdot \sigma$ and $b_M \cdot \sigma$.

$$\xi = \frac{\xi_M}{2} \cdot \{cos[a_A \cdot (T - A_s) + b_A \cdot \sigma] + 1\} \tag{5.4.7}$$

$$\xi = \frac{1 - \xi_A}{2} \cdot cos[a_M \cdot (T - M_f) + b_M \cdot \sigma] + \frac{1 + \xi_A}{2} \tag{5.4.8}$$

The material constants, a_A, a_M, b_A and b_M are given by Equation 5.4.9, where $tan(\beta)$ and $tan(\alpha)$ shown in Figure 5.12 are usually equal and indicate how the stress affects the transition temperatures Equations 5.4.10, and 5.4.11 respectively show the heating and cooling transformation start and stop stress ranges for an SMA.

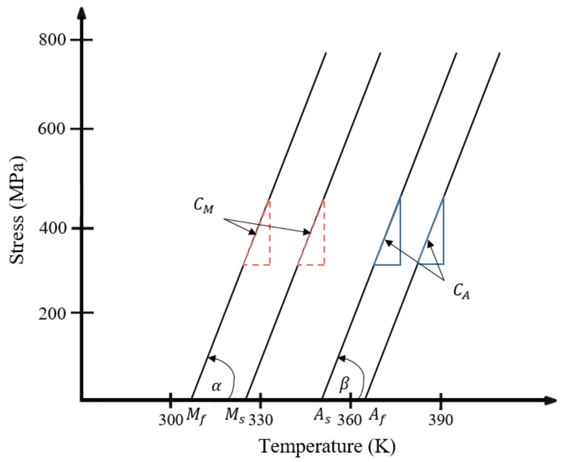


Figure 5.12: Stress-temperature relationship, with a SMA during phase transformation [195].

$$a_A = \frac{\pi}{A_f - A_s} \; a_M = \frac{\pi}{M_s - M_f} \; b_A = \frac{-a_A}{tan(\beta)} \; b_M = \frac{-a_M}{tan(\alpha)} \tag{5.4.9}$$

$$C_A \cdot (T - A_s) - \frac{\pi}{|b_A|} \leq \sigma \leq C_A \cdot (T - A_s) \tag{5.4.10}$$

$$C_M \cdot (T - M_f) - \frac{\pi}{|b_M|} \leq \sigma \leq C_M \cdot (T - M_f) \tag{5.4.11}$$

5.4.1.2 Thermo-mechanical behaviour of an SMA wire

In this section, the basic elements governing the modelling of the SMAs thermo-mechanical behaviour are introduced. The three elementary components of continuum mechanics that help to understand thermo-mechanical behaviour are kinematics, conservation laws and constitutive equations. Here the former (kinematics) describes the deformation of a continuum body the geometry of motion, with no regard to the origin of the deformation or motion. The conservation laws explain the interaction of external effects and the continuum body motion. Lastly, the latter, (constitutive equations) mathematically define the key characteristics of material behaviour, only predicted through analysis of experimental observations [195]. These three components are collectively viewed as the constitutive laws that govern the thermo-mechanical behaviour of SMA.

The thermodynamics of SMAs are assumed to be wholly defined by the set of variables (ε, T, ξ), where ε and ξ are respectively the infinitesimal strain tensor (Green strain) and martensite fraction. In this study, a variable speed phase transformation model proposed by Liang [201] is used to elaborate on the thermo-mechanical behaviour of the SMA wire.

Based on the results of the previous section, the thermo-mechanical model of an SMA wire relates the effect of the thermal loads and temperature distribution to the developed strains in the wire. Thermo-mechanical coupling of SMA responses gives an understanding of the total behaviour of the wire due to the different loading conditions. Temperature variations occur in the wire both in the forward and reverse phase transformation processes. These temperature variations are a consequence of latent heat being produced or absorbed.

The martensite fraction introduced in Equation 5.4.7 strongly dictates the amount of latent heat produced during transformation. In order to model the thermo-mechanical coupling of the wire, the Legendre transformation is used in conjunction with the first and second laws of thermodynamics [195]. These are coupled to give Equation 5.4.12, which is based on the conservation of energy with the assumption of an adiabatic heat transfer process. In Equation 5.4.12, the first term on the left-hand side describes the effect of the material stress state on the temperature changes. The second term on the left-hand side is related to the material heat capacity, while the third term describes the temperature changes due to variation in martensite fraction. This variation in the martensite phase occurs during heating and cooling and is associated with the latent heat during phase transformation.

$$T\alpha : \dot{\sigma} + \rho c\dot{T} + (-\pi + \rho\Delta s_0 T)\dot{\xi} = 0 \tag{5.4.12}$$

In the forward transformation, the latent heat absorbed causes the volume fraction of austenite to increase until full phase transformation is reached. Referring to Equation 5.4.7, this means that the time rate of change in the austenite volume fraction will be positive. Similarly, during cooling, the latent heat released during phase transformation results in a negative time-rate of change in the austenite volume fraction. Equation 5.4.13 represents the transformation function used to capture the thermo-mechanical properties of the wire during the forward and reverse transformations. The thermo-mechanical processes that occur at this transformation surface can be modelled by introducing a transformation function [212], such that:

$$\Phi = \begin{cases} \pi - Y & \text{whenever} \quad \dot{\xi} < 0 \quad (A \to M) \\ -\pi - Y & \text{whenever} \quad \dot{\xi} > 0 \quad (M \to A) \end{cases} \tag{5.4.13}$$

The transformation function, Φ, satisfies the condition of:

$$\Phi = 0 \tag{5.4.14}$$

The absorption and release of latent heat occurs at constant stress and temperature. Equation 5.4.15 is an explicit mathematical description of the consistency condition ($\dot{\Phi}$ = 0 [213]) and is given by:

$$\dot{\Phi} = \frac{\partial \Phi}{\partial \sigma} : \dot{\sigma} + \frac{\partial \Phi}{\partial T}\dot{T} + \frac{\partial \Phi}{\partial \xi}\dot{\xi} = 0 \tag{5.4.15}$$

The use of the formulation in Equation 5.4.15 is based on the assumption of the wire being rate-independent. This is also substantiated by the fact that the martensite phase transformation is diffusionless. From the forward transformation case, the terms in Equation 5.4.15 are given by:

$$\frac{\partial \Phi}{\partial \sigma} = \Delta S : \sigma + \Lambda \tag{5.4.16}$$

$$\frac{\partial \Phi}{\partial T} = \Delta \alpha : \sigma + \rho \Delta s_0 \tag{5.4.17}$$

$$\frac{\partial \Phi}{\partial \xi} = -\frac{\partial^2 f}{\partial \xi^2} = -\rho b^M \xi - \mu_1 - \mu_2 \tag{5.4.18}$$

The above expressions can be used to obtain Equation 5.4.19, which gives the evolution of the martensite volume fraction during phase transformation:

$$\dot{\xi} = -\frac{(\Lambda + \Delta S : \sigma) : \dot{\sigma} + (\rho \Delta s_0)\dot{T}}{\rho \Delta s_0 (M_s - M_f)} \tag{5.4.19}$$

Further increase in the wire temperature above A_f, the volume fraction of the austenite phase stays constant [195]. This is as a result of the wire having undergone full phase transformation. At this stage, $\dot{\xi} = 0$, which allows for the temperature of the wire to be solved, since all the other variables in Equation 5.4.19 are known. Finally, using Equation 5.4.20, the strain induced in the wire can be obtained, which enables the calculation of the wire displacement. From Equation 5.4.20, the relationship between wire temperature and strain is represented graphically in Figure 5.13 for different stress levels.

$$\xi = 0; \qquad S = S^A; \qquad \alpha = \alpha^A$$

$$\varepsilon = S^A \sigma + \alpha^A (T - T_0) \tag{5.4.20}$$

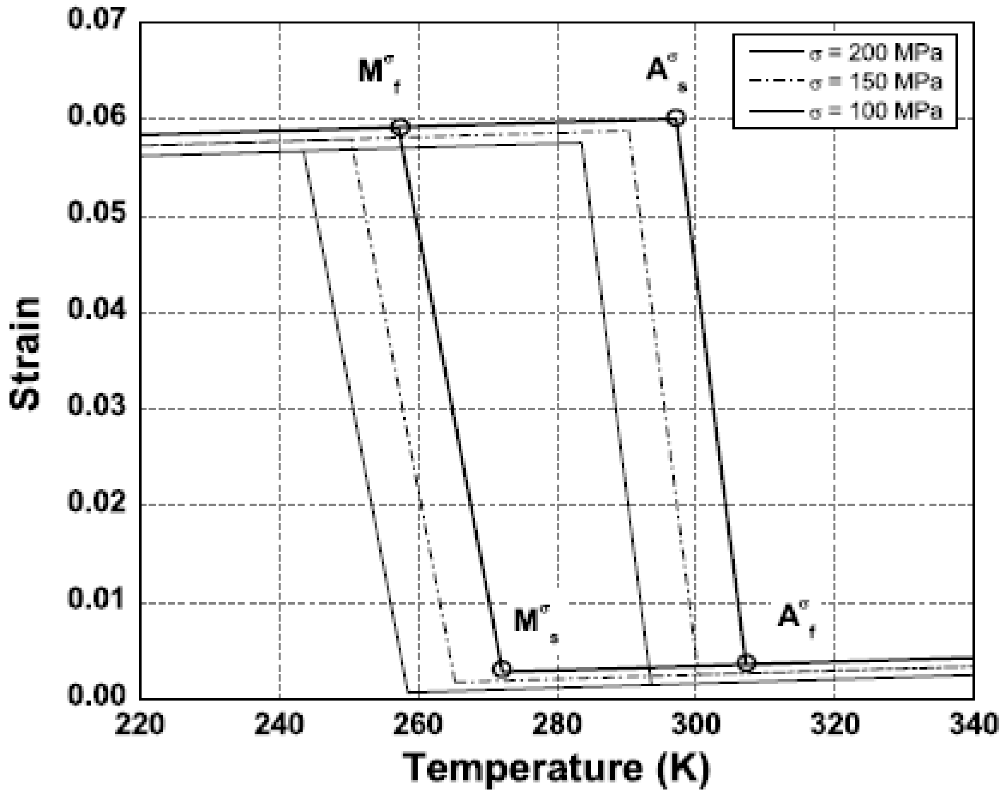


Figure 5.13: Variation of strain with wire temperature [195].

5.4.2 FEA of SMA actuator model

The behaviour of the SMA was modelled based on a steady-state FEA. From the first law of thermodynamics, the thermo-mechanical relation of the SMA is given by Equation 5.4.21; where ρ is the density of the SMA wire, u is the specific internal energy. The rates of heat generation, stress and strain rates are given by ($\dot{g}$); $\dot{\sigma}$, and $\dot{\varepsilon}$, respectively. The heat flux is represented by the parameter q [195].

$$\rho\dot{u} + q - \sigma\dot{\varepsilon} - \rho\dot{g} = 0 \tag{5.4.21}$$

Initially, the wire was loaded with the 1000g mass to model the displacements and subsequent deformations due to plastic strains. This mechanical load results in a de-twinned martensite state. Figure 5.14 shows the wire temperature contour plot. As expected, upon heating, the wire transitions into a full austenite phase due to resistive heating causing a uniform wire temperature of 121.43°C. With the wire being small in size, it quickly reaches thermal equilibrium throughout its whole structure when the power is applied. In this process, the phase transformation from the initial de-twinned martensite to austenite occurs because the power supply is sufficient enough to increase the wire temperature above A_s. This heating process either induces thermal stresses in the microstructure of the SMA or is dissipated to the environment. The wire deformation depends on the magnitude of the induced thermal stress.

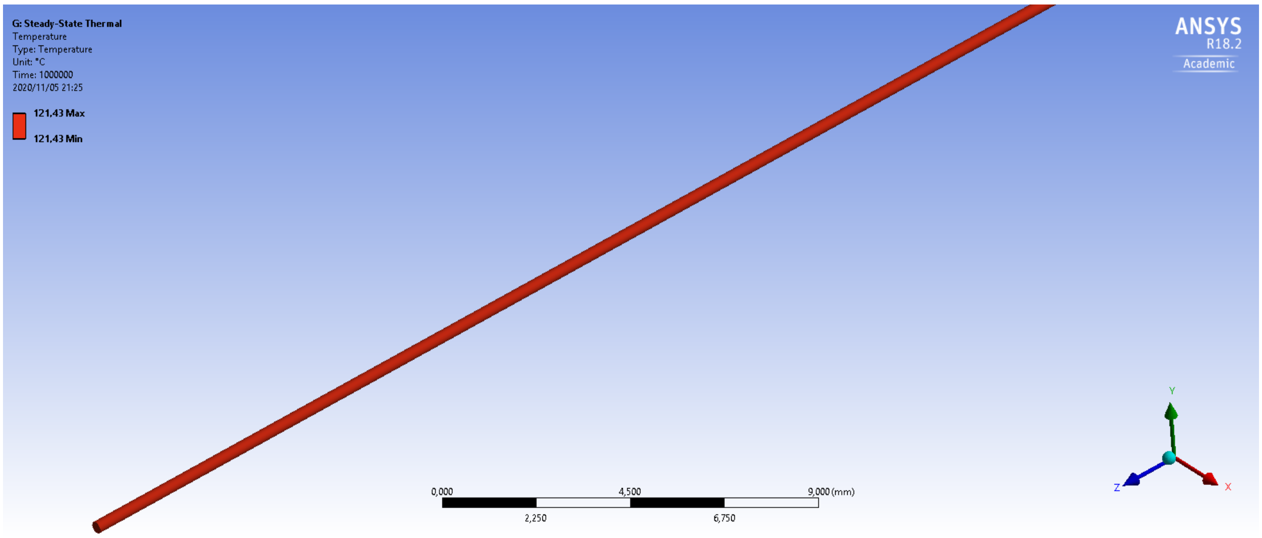


Figure 5.14: SMA wire temperature contour.

The coupled effect of the power input and the mass on the wire displacement is shown in Figure 5.15 contour plot. From this plot, it can be seen that the power input boundary has almost a zero displacement. This is because it has been constrained to a zero displacement by the remote displacement boundary condition. However, the slightly negative displacement observed can be attributed to wire deformation as a result of both the thermal and mechanical loads. These loads might induce a bending moment during wire contraction or expansion, which might lead to the observed negative displacement. With a maximum power input of 19W and a mass of 1000g, the maximum normal displacement of the wire was found to be 4 mm. For the case where only power was added, the maximum displacement of the SMA wire was found to be 4.35mm.

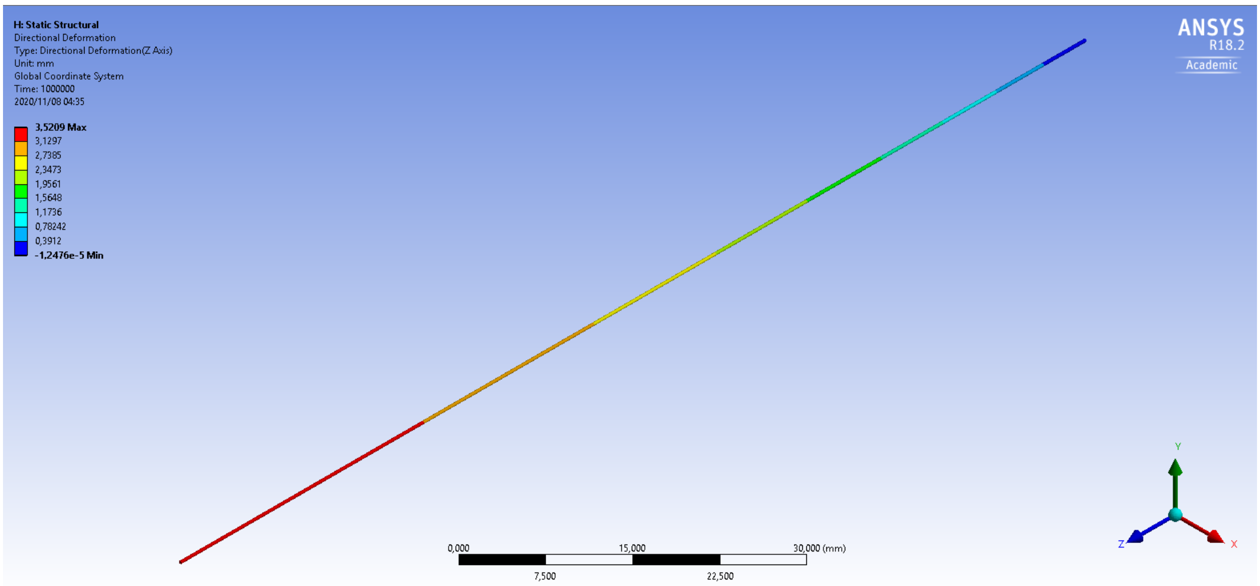


Figure 5.15: SMA wire displacement.

With the wire being subjected to two different loads, its response is thus a function of both the thermal and mechanical strains induced by these different inputs. Figure

5.16 shows the relationship between the power input and the displacement of the SMA wire modelled using ANSYS FEA. From the figure, it can be observed that as the power increases, so does the SMA wire displacement. This increase in displacement can be attributed to continuous dislocations of the SMA lattice structure as a result of an increase in thermal stresses [195]. It can also be observed that there is a drop in the rate of change in displacement at higher power inputs. The reason for this trend can be attributed to the energy being used to accomplish complete phase change from martensite to a denser austenite phase. This increase in density is associated with a decrease in volume, resulting in a reduction in strain rate with further increase in power input.

It can be observed that for the same power input, different mechanical loads result in different wire displacements when the graphs in Figure 5.16 are compared. From this figure, it can be seen that the mass results in a decrease in the SMA wire displacement. This is because the mass acts as a tensile load which causes elongation of the wire. This elongation opposes the contraction effect caused by the power input. The difference between these two responses gives the overall displacement of the wire. Also observed in Figure 5.16, the displacement response tends to flatten out at higher power inputs. For the response with the mass, this occurs at 10W, and 9.5W for the response with no mechanical load. The reason for these differences is due to the effect of the mass, which tends to increase the phase transition temperature [193]. Again, the flattening out of the displacements response is due to the wire being in a complete austenite phase which is denser and exhibits a larger strain resistance as opposed to the martensite phase.

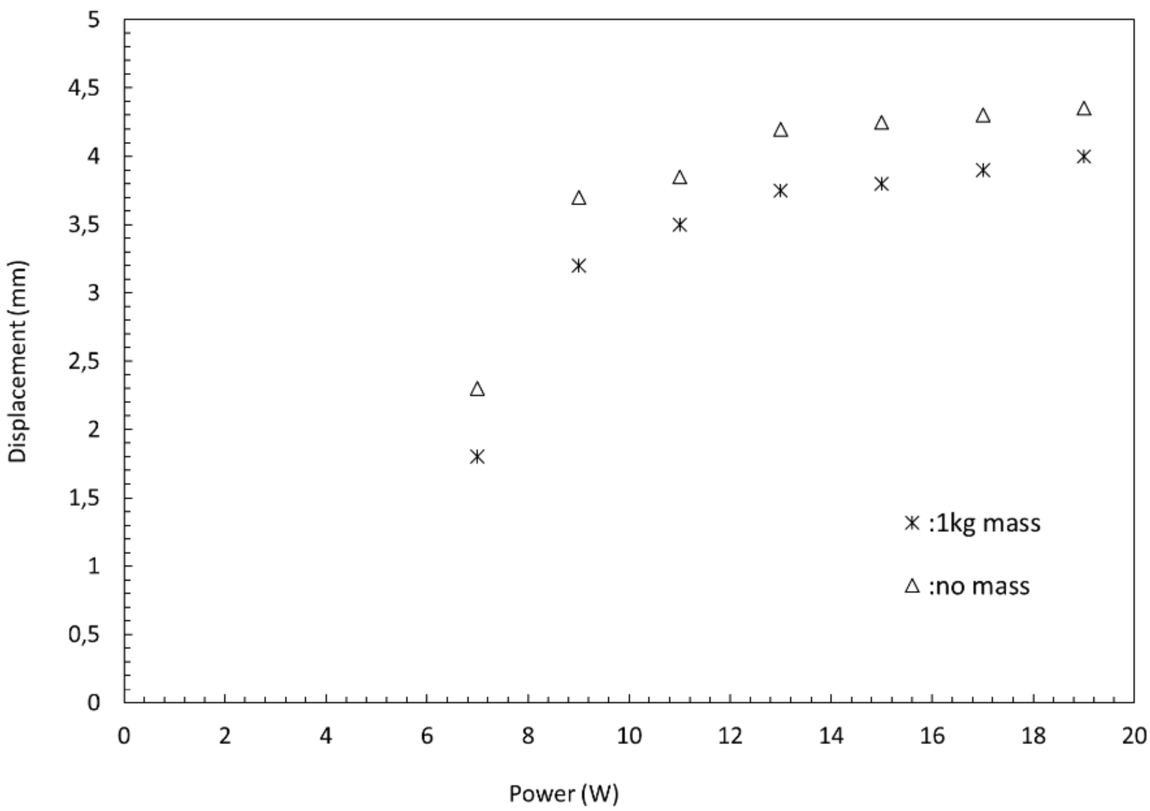


Figure 5.16: Variation of wire displacement with power.

5.4.3 NARX NN SMA actuator model

Computational black-box system identification requires minimal prior knowledge of the system and instead relies on the guidance of past successful models. This is because black-box models replicate the behaviour of the underlying system by mapping the input-output data with a function or finite system of functions, with non-specific parameters. Computational Black-Box models are particularly effective in representing multi-variate non-linear systems, where internal states are time-variant, highly-coupled and difficult to determine analytically.

As such, in this section, a novel NARX NN black-box model of an SMA actuator, comprised of a regulator, SMA wire and mechanical assembly is presented. Figure 5.17 presents the model structure used in this study, including the inputs and output of the NARX NN SMA actuator model as well as the expected hidden model states. In brackets are the variables that significantly influence the hidden states. Variables n and ν denote the number of cycles and velocity of the sliders and levers, respectively.

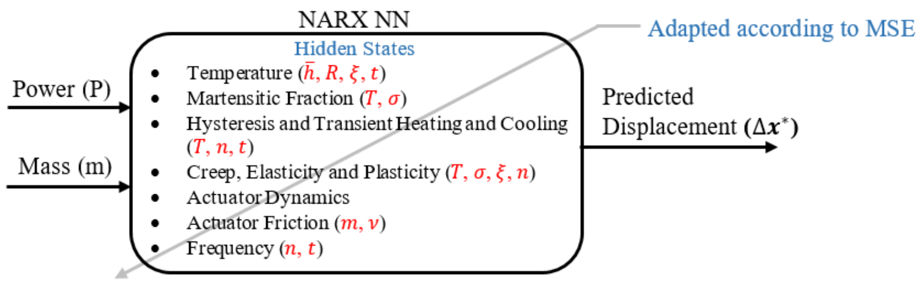
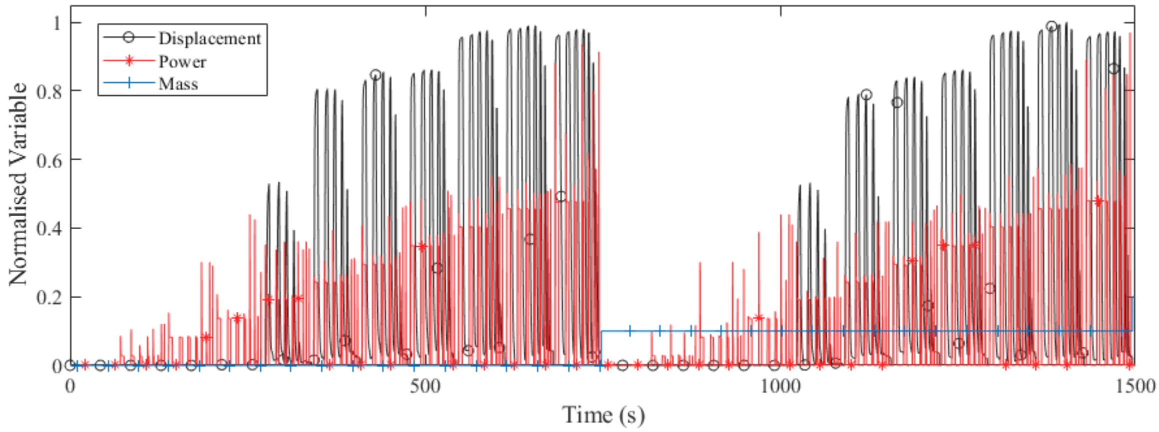


Figure 5.17: A multi-input, single output NARX NN black box model of an SMA actuator with identified hidden states.

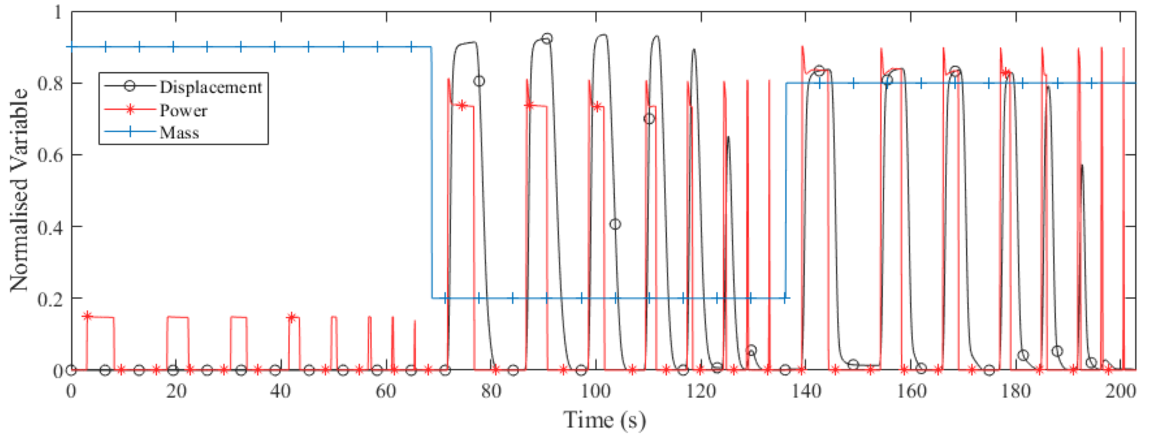
The system identification process typically consists of four steps—namely, (1) Data acquisition and pre-processing, (2) Selection of model class and structure, (3) Estimation of parameters and (4) Model validation [214]. Accordingly, this section is structured as follows: data acquisition and pre-processing, model structure, training and optimisation, model validation and the Simulink model. In the data acquisition and pre-processing section, the focus is placed on the preparation of raw data whilst maintaining temporal coherence. The model structure, training and optimisation section is guided by minimal training, testing and validation errors. The ability to generalise and capture specific SMA actuator characteristics is a prerequisite to this section. In the model validation section, the system response to various standard input signals is investigated; the inputs include step, pulse and sine signals of varied frequency. The development of the NARX NN SMA actuator model concludes with its implementation in a simulation environment.

5.4.3.1 Data acquisition and pre-processing

The raw data was constructed from 121 trials, where each trial was subject to a unique combination of power and mass; a segment of the raw data is illustrated in Figure 5.18a. Each trial lasted 70s and resulted in approximately 15740 samples. As mentioned (refer to Section 5.3), the power was locally varied in an ascending manner, while the mass was globally varied. This yielded the cyclic increase in the measured displacement due to power, while concurrently yielding a macroscopic decrease due to the increase in mass. In Figure 5.18a, the data is min-max normalised such that displacement variation, power and mass is scaled from zero to one.



(a)



(b)

Figure 5.18: Input-output data used in the feedforward and recurrent training of the NARX NN model. Figure 5.18a shows a portion of the raw data, while Figure 5.18b shows the data after pre-processing. In both figures the data is normalised for convenience.

Correspondingly, the normalisation factors for displacement, power and mass are 4.67mm, 19.26W and 1000g, respectively. Recall that the maximum strain of an SMA wire lies typically in the range of 2% to 6%. For the SMA actuator in this dissertation, with an original length of 200mm, this corresponds to a strain in the range of 4mm to 12mm. Notably, the maximum measured displacement of the SMA actuator lies towards the minimum of this range. This is due to the 25 preliminary trial cycles (refer to Section 5.3) which stabilised, conditioned and mitigated the initial stochastic behaviour of the SMA actuator. It is worth noting that the first few trial cycles yielded a displacement in the vicinity of 8mm.

It was apparent that the raw data needed to go through a pre-processing step after an initial test model demonstrated four key issues. The first issue relates to the spike in power at every instance power is applied. These spikes, which can be seen in Figure 5.18a, are due to the accommodation of the power supply to an initial peak current draw. Moreover, the spikes in applied power were inconsequential to the displacement of the

SMA actuator since they occurred over a very short interval and accordingly exhibited a frequency far greater than the upper bound of the SMA actuators bandwidth. Using the signal analysis application in MATLAB with a median filter and smoothing factor of 0.1, the spikes were removed, as shown in Figure 5.18b. The reasoning behind the removal of the spikes was to decrease the number of sudden changes in power, thereby reducing the risk of exploding or vanishing gradients during the training of the NARX NN model.

The second issue stemmed from the scale of the data involved in the training of the model. With 1904730 samples for each of the variables, it was noted that the use of the raw data for training would negatively impact the computation time. This decreases the chance of achieving a minimal global solution and increases the RAM requirements. Therefore, using a moving average with a window length of 80ms and no overlap, the samples were reduced to 126982 (see Digital Appendix for the code). The data was formatted as a time series to maintain temporal coherence. Evidence of the moving average can be seen when comparing the smoothness of the data depicted in Figure 5.18a to that of Figure 5.18b.

The third issue originated from the order in which the data was recorded. Because of the periodic nature of the recorded data, the trained NARX NN would likely learn to predict the periodicity of the data instead of learning the underlying characteristics of the SMA actuator. Also, the periodicity could increase the number of delays required, impacting training and the model size. The issue of periodicity is common in time series modelling and is often referred to as seasonality, with examples including day-night and seasonal cycles [215]. The periodicity in the raw data comes as a direct byproduct of the order in which the data was collated.

To remove the effects of periodicity, detrending is often employed, in which the moving mean is subtracted from the current state. This technique, however, can not be used in this application since it will result in unrealistic training data with negative displacements. The solution to this involved the pseudo-random shuffling of the data while maintaining the transient structure and synchronicity of the data. Specifically, this was achieved by splitting the dataset according to the troughs and then shuffling the segments (for code see Digital Appendix). The outcome of this process is shown in Figure 5.18b, where the displacement-power-mass segments differ from those shown in Figure 5.18a.

The last issue relates to the negative or non-zero offset of the SMA actuator displacement once the power is removed. The negative displacement comes as a consequence of the test-rig and the sampling speed of the Arduino. In particular, if the velocity of the slider mechanism (over a given distance dx) exceeds the time required for the execution of the *DigitalRead* function, then it is certain that pulse edges from the optical encoder will be missed. This could not only result in a negative displacement but also a non-zero offset, most notable in the tail-end of the displacement curve.

The inputs and outputs were duly segmented by troughs and shifted up or down such that the troughs were aligned with the $y = 0$ axis. This was done to remove the negative or non-zero offset. The discontinuities arising from segmentation and shifting was addressed through a Gaussian filter with a smoothing factor of 0.02 (MATLAB signal analysis application). The smoothing was subtle enough not to impact the transient displacement during the heating and cooling of the SMA actuator. In Figure 5.18b, it is seen that the offset is greatly reduced. Removal of the stochastic offset improved the training of the NARX NN model. Understanding the trends in the raw data helps to establish an intuitive feel for the data and consequently gives the capacity to make better model related decisions. As such, a plot of the input versus the output variables (Figure 5.19) provides insight into the overall behaviour and relationships present. In Figure 5.19, the displacement is a function of both the mass and the power. Here, it may be deduced that an increase in power equates to an increase in displacement, while an increase in mass equates to a small decrease in displacement.

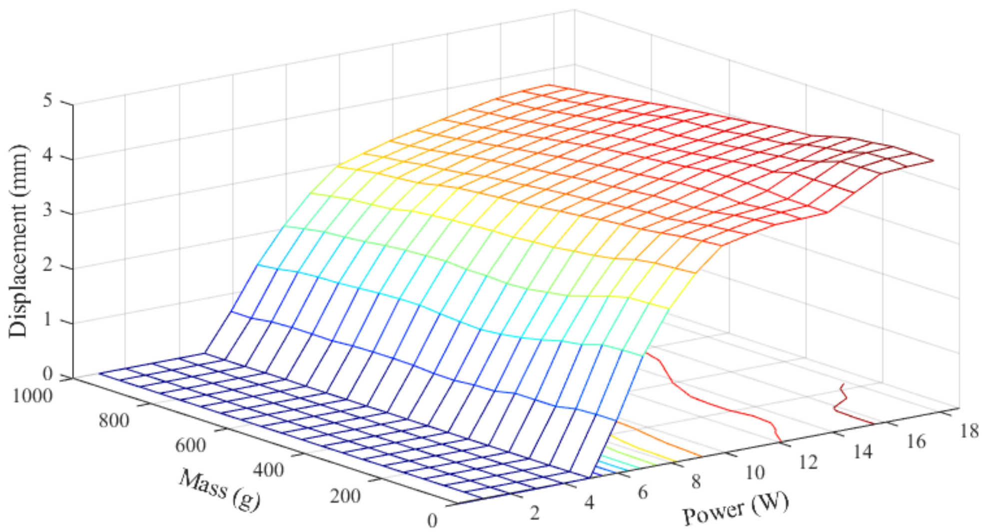


Figure 5.19: Surface plot of displacement as a function of both mass and power.

Understanding the trends in the raw data helps to establish an intuitive feel for the data and consequently gives the capacity to make better model related decisions. As such, a plot of the input versus the output variables (Figure 5.19) provides insight into the overall behaviour and relationships present. In Figure 5.19, the displacement is a function of both the mass and the power. Here, it may be deduced that an increase in power equates to an increase in displacement, while an increase in mass equates to a small decrease in displacement.

Before unpacking the relationship, it is worth noting that Figure 5.19 only provides details on the steady-state behaviour of the SMA actuator and excludes the effects of frequency. This is because the figure was constructed using the maximum displacement in each of the 121 trials. Some interesting observations can be deduced from Figure 5.19. Firstly, the displacement is saturated at zero for power less than 5W. This indicates that the temperature of the SMA wire never reaches A_s and that the heat generated seems to be dissipated by convection at a greater or equal rate. Also, for power exceeding 10W, the change in displacement decreases significantly due to a possible increase in the resistance of the SMA wire as the temperature increases beyond A_f.

Another observation is the almost linear relationship between displacement and mass. Explained by the inverse relationship between force and displacement for constant work, a lower mass yields a higher displacement, as shown in Figure 5.19. An irregular increase in the displacement is noted at 14W and 0g. Though this irregularity may be an error in the measurement of the maximum displacement for the trials in question, it is equally likely that it stems from the momentum of the slider. This may seem counter-intuitive given that a larger mass should yield a greater momentum, however, since the mass is not coupled in the direction of the momentum it has no significance and instead, the momentum is solely proportional to the velocity. From Figure 5.19, it is clear that there is an upper bound to the displacement of the SMA actuator.

The displacement-power cycle shown in Figure 5.20 is constructed using segments of data with varying power and fixed mass (1000g). Any conclusions derived from the figure is limited to a low-frequency response since the segments correspond to the longest applied power interval, which is fixed at 5s. From Figure 5.20, it is apparent that as the initial power increases, so too does the deviation in power. This seems to validate the fact that the resistance of the SMA wire plays a major role in the experienced and applied power. Where the former (experienced power) and latter (applied power) respectively refer to the power losses, as seen by the actuator and the power supply. Information related to the rate of transient heating is expressed through the gradients of the lines depicted in Figure 5.20. Here, the gradients of the lines of best fit vary slightly between the different applied powers, suggesting that the displacement-power relationship remains similar. In other words, more power does not necessarily translate to a faster response (recall that each cycle takes place over 5s).

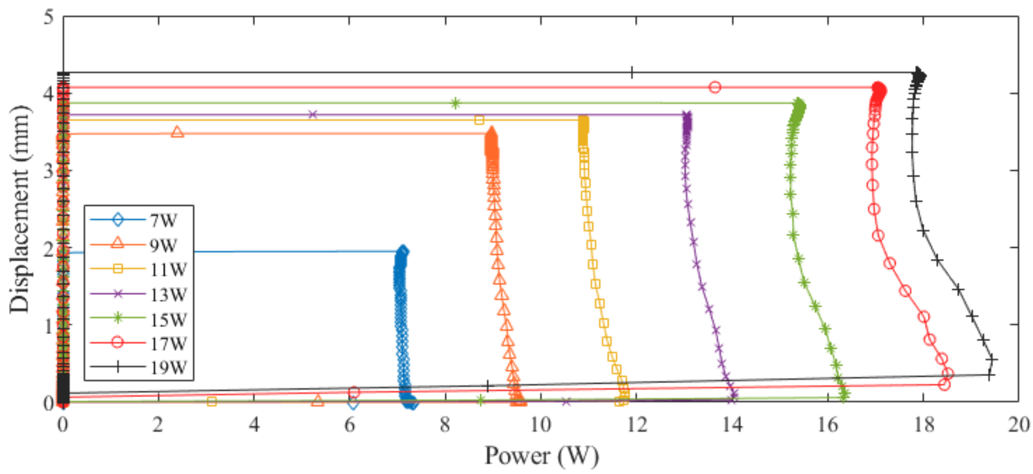


Figure 5.20: Displacement power cycle constructed using varying power and fixed mass (1000g) inputs.

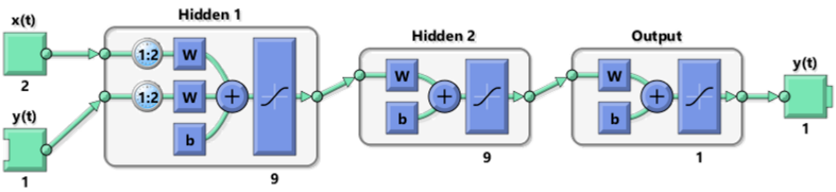
Intuitively this presumption may not make much sense, however, considering the existence of a power threshold for a given SMA wire diameter may prove useful. This means that for the wire diameter in this dissertation, has a negligible difference in the reaction time of the SMA is seen for a constant applied mass. It should be noted that the difference in reaction time should be more substantial for thicker SMA wires. In general, it is observed that the largest contributing factor to the heating and cooling time is the mass, where an increase in the applied mass results in an increase in the rise time and a decrease in the fall time. Another significant observation from Figure 5.20 is shown in the concavity of each of the cycles. The concavity increases with increasing power, possibly signifying the abrupt change from martensite to austenite.

5.4.3.2 Model structure, training and optimisation

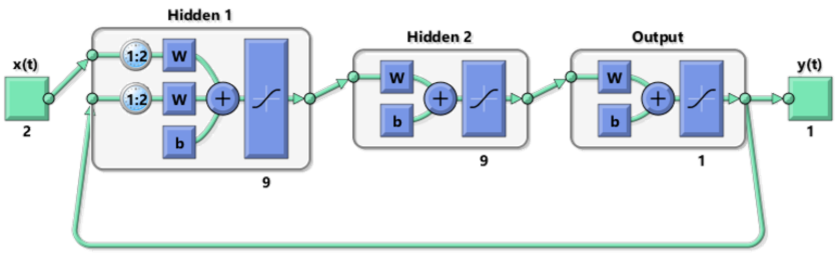
Figure 5.21a shows the network diagram of an open-loop NARX NN model used in the initial training phase. This is a two-layer feedforward model used to predict the output displacement. As can be seen, $x(t)$ represents the two exogenous inputs (power and mass) $y(t)$ is the expected output displacement. The number of neurons in both the hidden layers was determined after iteratively testing a maximum of ten neurons per layer to get hundred possible configurations. It was found that the configuration with nine neurons in each hidden layer resulted in the least MSE and was therefore selected as the model training configuration. The trained NARX model was then transferred to the closed-loop. In the closed-loop, by using the open-loop weights and biases, the model's performance dropped and was therefore retrained to get the configuration shown in Figure 5.21b with optimal weights and biases.

Furthermore, it can be seen in Figure 5.21 that the output variable is a function of both the exogenous inputs and the previously calculated output value. In this model, a tapped delay line memory containing the most recent outputs is formed. This means that after each cycle, a new output replaces the old one. Also, this has the advantage of saving memory since an older output is replaced in every new cycle, as opposed to when the new output is appended to the data set.

The strength of the relationship between the input signals and the output variable of the SMA actuator NARX NN model can also be obtained. This is accomplished through a statistical correlation analysis which gives a quantitative measure of how the input and output variables are related. The limits of correlation analysis are +1 or -1. These extreme values respectively indicate a perfect positive and perfect negative input-output relationship.



(a) Open loop.

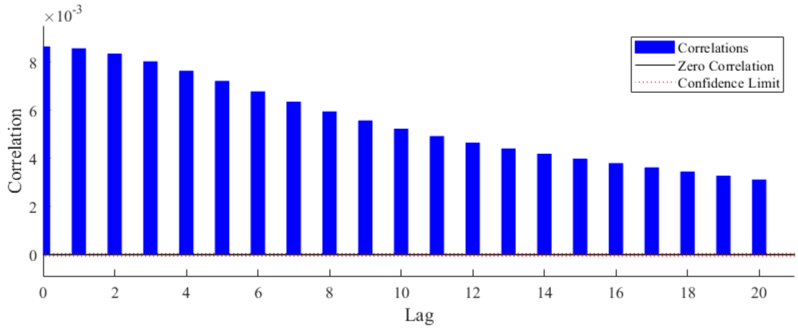


(b) Closed loop.

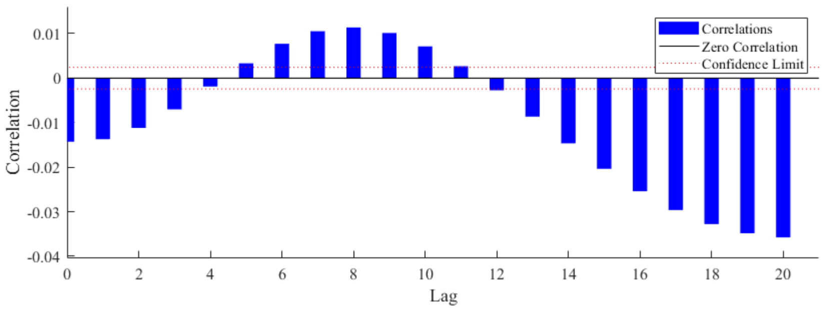
Figure 5.21: NARX NN model diagram.

Figure 5.22a shows a correlation function for the SMA actuator model. The dashed line represents the confidence limit of the correlation. The output data were de-trended before correlation to ensure that long term aspects that cause distortion are eliminated. It can also be observed that the results show a very small and decreasing correlation with lag. However, the correlations are outside the confidence limit, meaning that the model was well suited to predict the output response.

The cross-correlation of the SMA actuator model is shown in Figure 5.22b. In this figure, it can be observed that the cross-correlation between the two input variables relative to the output tends to fluctuate between positive and negative values. Though the cross-correlations are much less than one, they still are outside the confidence limit, implying the model was still well capable of predicting the output response.



(a) Direct correlation.



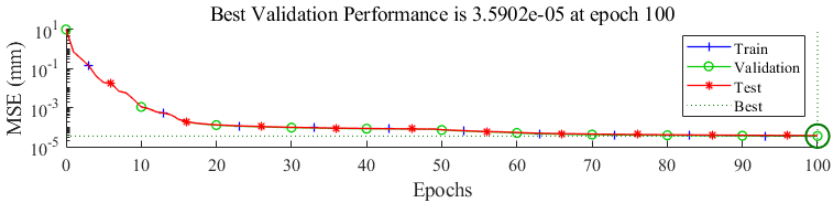
(b) Cross correlation.

Figure 5.22: SMA actuator model correlation function.

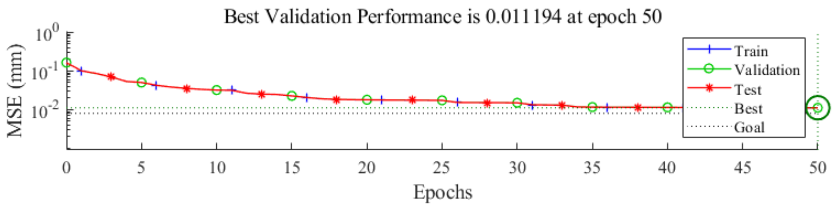
Figure 5.23 shows the variation of MSE with epochs for one open-loop and two closed-loop SMA actuator models during the training, testing, and validation processes. It can be observed from the graphs that there is a general decrease in MSE with increasing epochs for all three models. This decrease is in the form of exponential decay for the three models. It is also worth noting that the open-loop model has a higher initial MSE than the two closed-loop models. This is because, in closed-loop systems, the response of the model is relatively insensitive to internal parameters variations compared to the response in open-loop systems.

From Figure 5.23a, it can be observed that MSE drops by approximately 99% during the first 20 epochs. In this region, the MSE decays with epochs. From about 20 to 80 epochs, the relationship between MSE and epochs is approximately linear with a very small negative gradient. In this region, the drop in MSE is less than 10% of the MSE at 20 epochs. From 80 epochs to 100 epochs, it can be seen that the graph starts to flatten out with MSE approaching a constant value of $3.6e-05$.

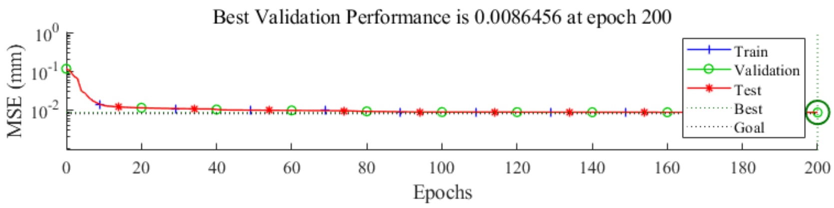
From Figure 5.23b, it can be seen that during the first 15 epochs, MSE is inversely proportional to epochs. With an increase in epochs beyond 15, this relationship starts to follow a linear relationship with a constant negative gradient. The minimum MSE is obtained at 50 epochs and corresponds to the best validation performance of the model. Analysis of the results in Figure 5.23c shows that for the first 10 epochs, MSE is inversely proportional to epochs. In this region, MSE falls by approximately 90%. After that, the graph asymptotically approaches an MSE of 0.01. At 200 epochs, the MSE reaches its minimum of 0.00865 mm, and this corresponds to the point of best validation performance.



(a) Open-loop performance.



(b) Closed-loop performance 1.



(c) Closed-loop performance 2.

Figure 5.23: Variation of NARX model performance with epochs.

Figure 5.24 shows the results obtained after training the NARX model. The first graph (from the top) shows the variation of the gradient descent with epochs. In this figure, it can be seen that there is a general decrease in gradient with the number of epochs. As epochs continue to increase beyond 140, it can be observed that the gradient descent is approximately periodic. From this graph, the training reaches a steady state at 200 epochs for which the gradient will be 0.63283. This point gives the best set of coefficients for the model. This gradient descent function is important for system optimisation of the SMA actuator model to obtain the best model coefficients that minimise the neural networks model's computational cost.

This process is typically used because analytical methods are not sufficient enough to calculate the bias and weight parameters. The second graph in Figure 5.24 shows the variation of MU with epochs for the trained model. This value is used to control neuron weights during the updating process. It can be seen that the MU spikes during the first few epochs (before stabilising around 40 epochs) and then drops to a local minima between 75 to 95 epochs. With further increase in epochs, the MU stays constant at 0.01 and starts to be periodic after 138 epochs. From this trend, it can be concluded that the further training of the model will not improve its learning performance as the

MU is now oscillating with the same periodicity. The third graph in Figure 5.24 shows the model validation results for the actuator. This is achieved by evaluating how well the network performs using held-out validation data set in each training loop. For this test process, the data was split into two groups, the training and testing sets. The model was then trained using the testing data to obtain a function approximator. This approximator was then used to predict outputs using the testing data set. From the figure, it can be observed that the number of failed validations is zero throughout the testing and training process. This shows that the model was well suited to predict the output displacement of the actuator. However, very few validation errors occur after 120 epochs. This can be attributed to continuously changing weights and biases beyond 120 epochs resulting in a less accurate prediction function. The validation set can also be used for tuning purposes in a neural network, such as selecting the number of hidden layers.

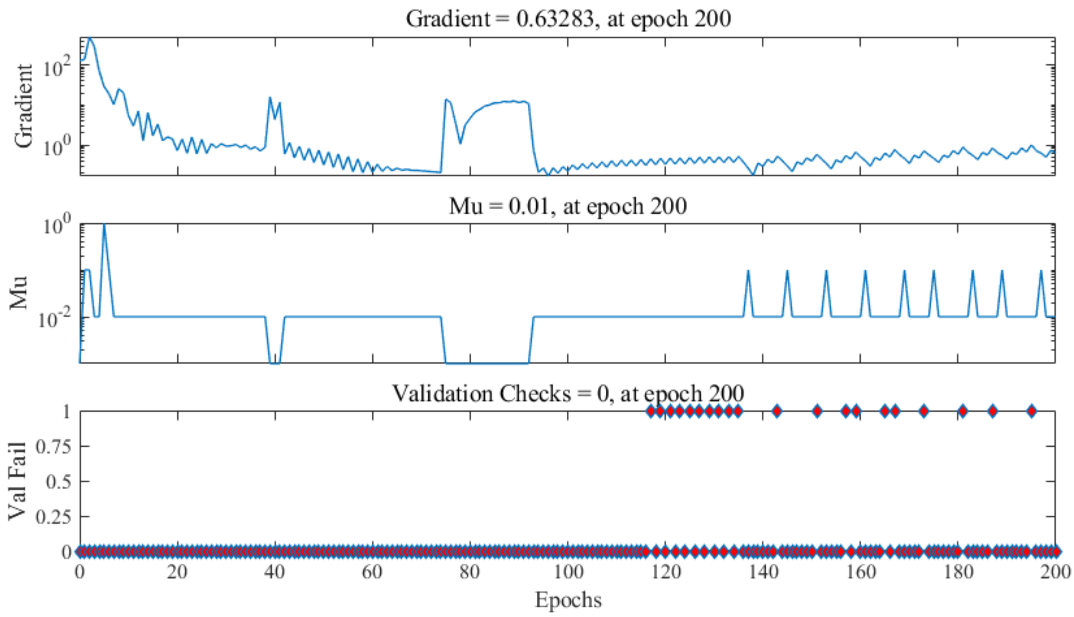


Figure 5.24: NARX NN model training results

Figure 5.25 shows the results of the actual and expected output displacements for the actuator during the training, testing and validation stages. As can be observed, the output displacement is periodic, due to the **ON-OFF** switching of the input power during the experiment. This resulted in the SMA having a periodic response. Also, the crests of the system response curves are much higher between sample 1000 to 2000, where a higher power input range was used. Between samples 2001 to 3000, the power input range was lower. This resulted in the model having larger displacements for larger power inputs than in the low-power range. Also, it should be noted that the first 1000 samples were used to condition the SMA so that its behaviour becomes more deterministic.

Figure 5.25 also shows the model error, which is the difference between the expected and actual outputs. It can be seen that when switching between heating and cooling (corresponding to turning power **ON-OFF**), the model fails to predict the actual output displacement accurately. These errors are a result of residual strain accumulation with increasing cycles.

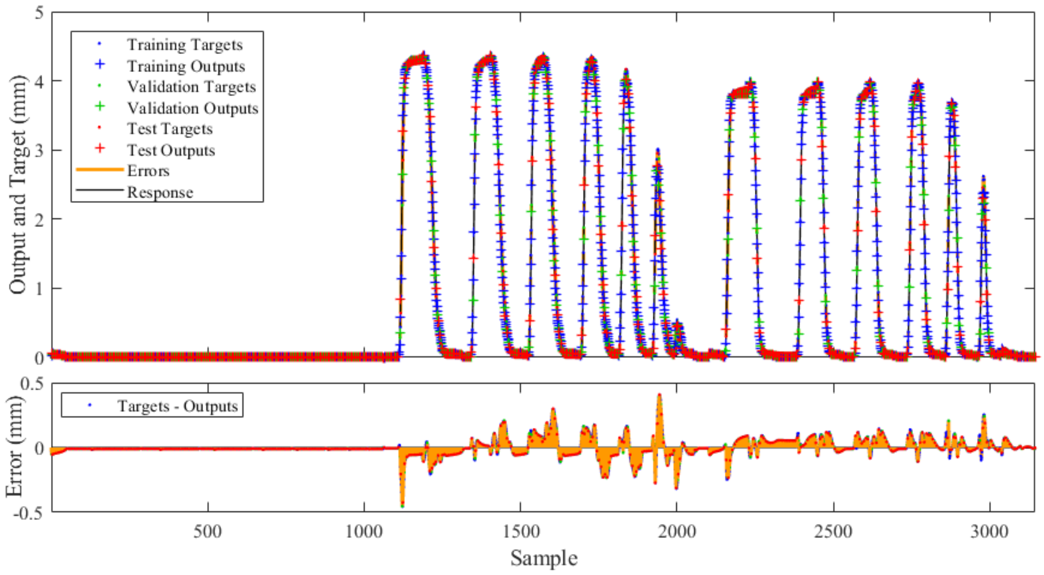


Figure 5.25: Actual and expected actuator output displacement during training, testing and validation

Figure 5.26 shows a histogram of how the neural network errors are distributed on the training, testing, and validation instances. It can be seen from the chart that the training process has the largest error contribution than both testing and validation process. Also, it can be seen that the testing and validation processes have similar error contribution. The main reason behind this observation is that the training process uses about 70% of the data, and thus its cumulative error becomes significantly higher than the testing and validation processes, which both use 15% of the data.

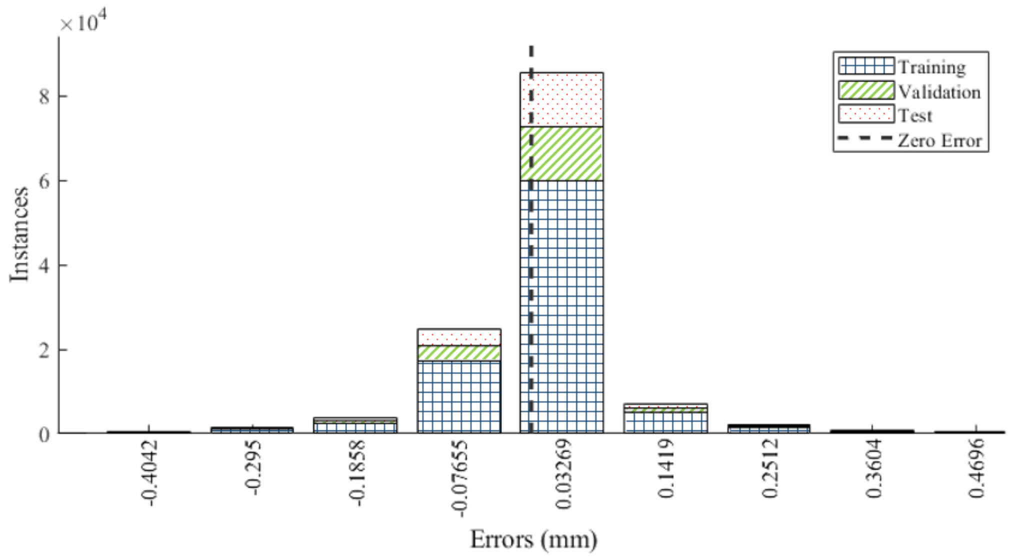


Figure 5.26: Model error distribution.

A linear regression analysis of the neural network outputs and the target outputs from the training, testing, and validation processes are shown in Figure 5.27. As can be seen in the plots, all the predicted model outputs are close to the actual target outputs and follow a similar trend. This is justified by the correlation coefficient, which is very close to one. From these results, we can conclude that the neural network model developed is well suited to predict the actuator displacement.

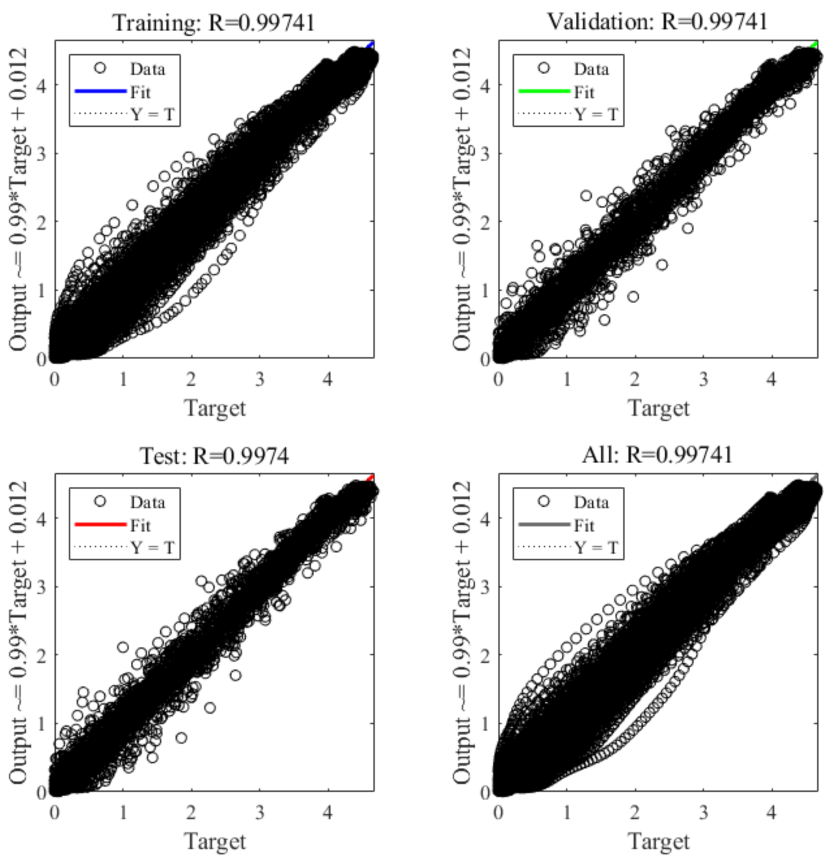


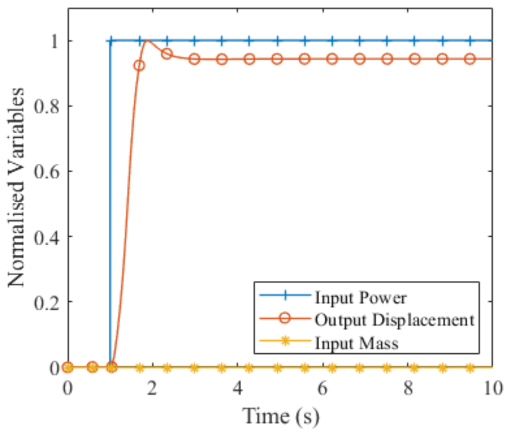
Figure 5.27: Linear regression analysis of the NN and target outputs.

5.4.3.3 Model validation

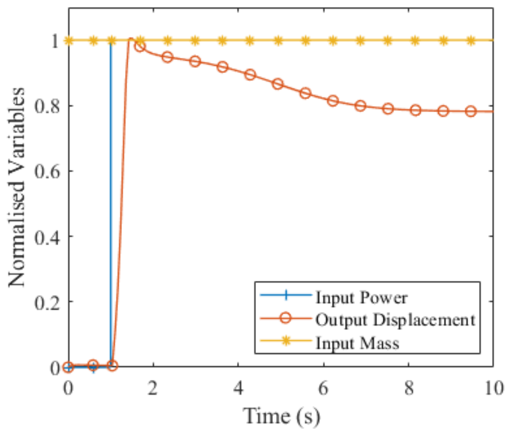
Figure 5.28a shows the model normalised response due to a step input in power signal without any external load applied. It can be seen that the model response has an initial overshoot which then decays as the output reaches steady state. Resistive heating from the power input causes the actuator to change phase from a stable low-temperature martensite to a stable high-temperature austenite phase. Being transient, this overshoot dies out upon complete phase transformation due to the effect of the SMA wire resistance in the full austenite phase. This response is similar to that of an underdamped system with initial overshoot.

Figure 5.28b shows the actuator response due to a step input in power and a constant load. It can be seen that the actuator system response shows a sudden peak when the power is switched on and then starts to drop continuously with time. In this process, the applied load causes an increase in the phase transformation temperature, and hence the input power is no longer enough to cause any phase change from martensite to austenite. The power input is then used to increase the SMA temperature and also induce lattice dislocations. As the SMA temperature increases, the proportion of lattice dislocations decreases due to an increase in SMA wire resistance. The opposing effects of the applied load and these lattice dislocations cause the displacement to drop with time continuously.

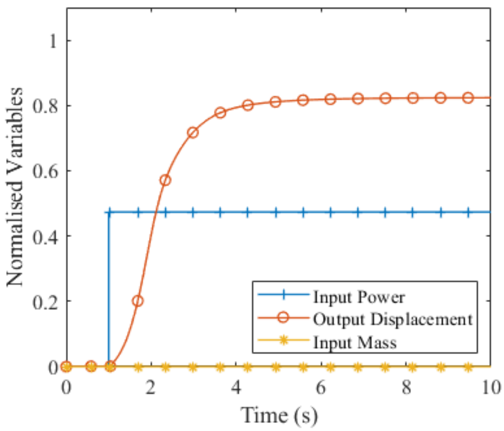
The step response of the actuator model with a lower power input and no external load is shown in Figure 5.28c. It can be observed that the response is quite slow to reach equilibrium as the heat generated is not sufficient to cause any partial phase change. In this process, most of the heat generated is dissipated by convection, resulting in less heat energy to initiate crystal lattice movements that result in the microscopic displacements. The response of the system with the load and lower power input is shown in Figure 5.28d. In this figure, it can be seen that the added mass causes the output displacement to be lower as compared to Figure 5.28c. This can be attributed to the fact that the extra mass input causes an elongation of the SMA, which opposes the displacement due to resistive heating from the power input. Also, this added energy is not enough to onset formation of austenite; hence the actuator structure will still be in the stable martensite region.



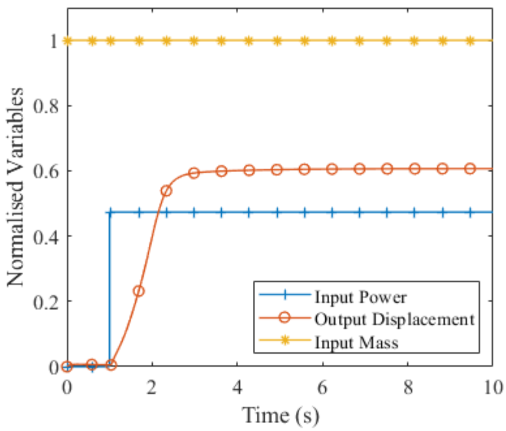
(a) Maximum power with zero load.



(b) Maximum power and load.



(c) Half maximum power with no load.



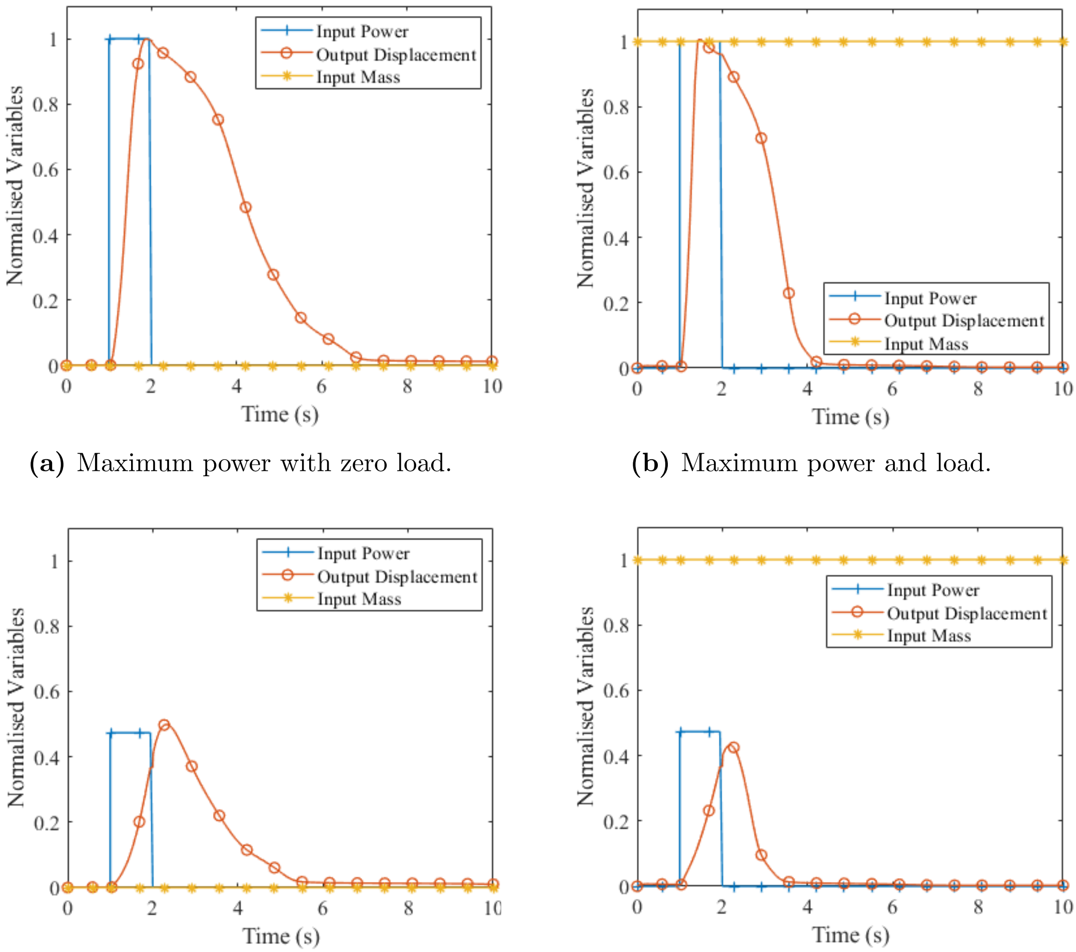
(d) Half maximum power with load.

Figure 5.28: Normalised response due to a step input.

Figure 5.29a shows the response of the actuator due to a pulse power input. It can be seen that the displacement reaches a peak value and then starts to decay to a constant asymptotic zero displacement with time. This drop-in displacement is due to an increase in the SMA resistance at higher initial temperatures, as well as energy loss due to natural convection after the input signal has been removed. The SMA will then try to return to its original un-deformed state influenced by the shape memory effect. Also, it can be seen that the displacement never goes back to zero, which can be attributed to the residual strains developed in the lattice structure. The response in Figure 5.29b shows a similar trend to that previously described. However, in Figure 5.29b, the extra load causes the SMA to return to its equilibrium position quickly.

Figure 5.29c shows a similar trend to the one exhibited in Figure 5.29a, but with a slight lag in system response. In this process, when the power is removed, most of the heat generated is dissipated by convection, resulting in less heat energy to initiate crystal lattice movements that result in the microscopic displacement of the SMA wire. Figure 5.29d shows the response due to a load and a lower power input. This response is similar to that shown in Figure 5.29b, but with a lower peak displacement due to the lower power input. Also, this response exhibits the same lag just as the response in Figure 5.29c and adheres to the same reasoning.

Furthermore, an important observation from these graphs is that the rise and fall time response respectively increase and decrease as the load is increased. This can be attributed to the fact that the applied load causes an increase in the inertia of the wire. These extra input loads cancel the effect of the residual strains in Figure 5.29a and 5.29c, causing the displacement to return to zero.



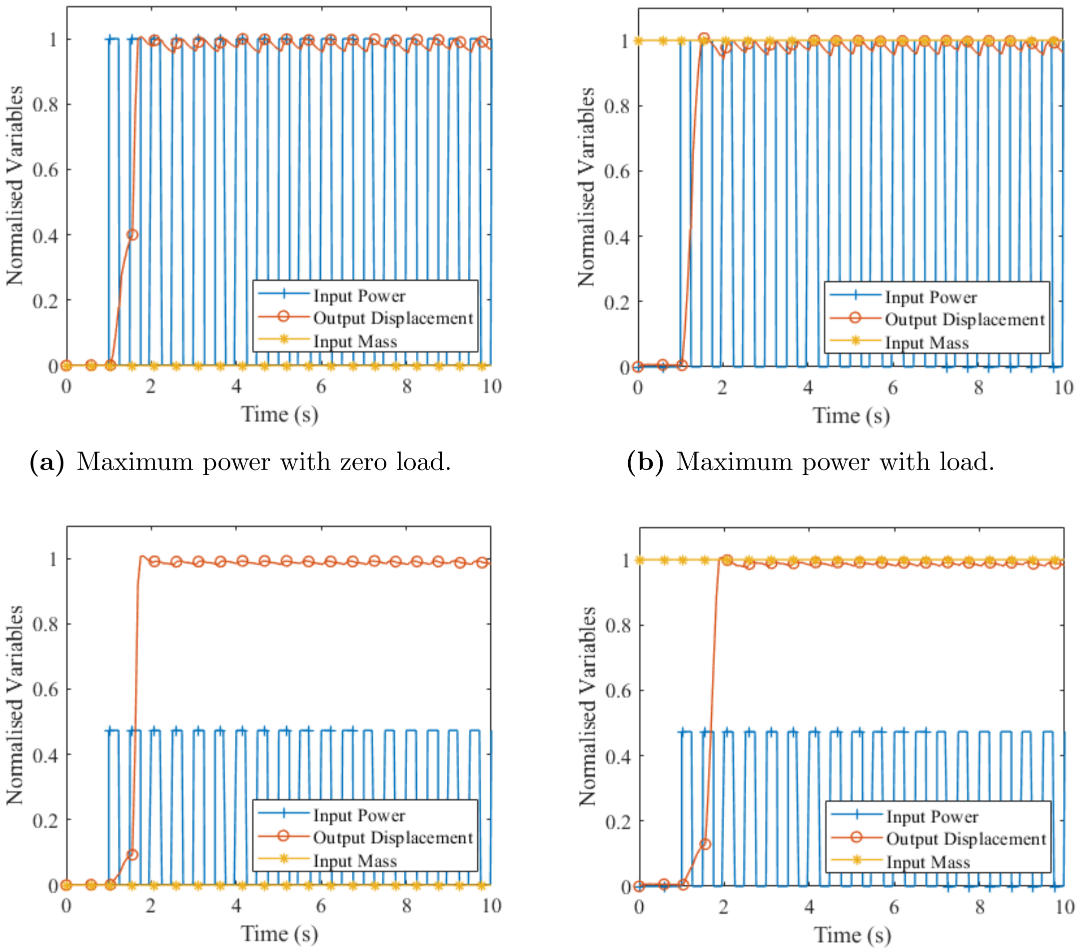
(a) Maximum power with zero load. **(b)** Maximum power and load.

(c) Half maximum power with no load. **(d)** Half maximum power with load.

Figure 5.29: Normalised response due to a pulse input.

Figure 5.30a shows the system response due to a pulse train input. As can be seen, the displacement is in the form of a periodic frequency response with peak displacement for each pulse input. Also, the displacement never reaches steady state even when the input is removed. This can be attributed to the fact that the time interval between successive pulse inputs is much smaller than the speed of response of the SMA, that is, the rate of cooling is very slow. Figure 5.30b follows the same trend as Figure 5.30a, but with an extra load as the input. The interval between successive pulses is still very small, such that the response of the SMA due to the load only is impossible to obtain before the next power input kicks in. This pulse input does not allow the system to reach equilibrium; hence the system response is transient.

The results in Figure 5.30c are similar to the results in Figure 5.30a, but with lower power input. The difference between the two graphs can be seen by observing the rate of change of displacements between two successive pulses. In Figure 5.30a, this change in displacement is much higher than that observed in Figure 5.30c. The reason being that, for higher power input, the SMA resistance increases as the temperature rises beyond A_f. The response in Figure 5.30d is similar to the one explained in Figure 5.30c.



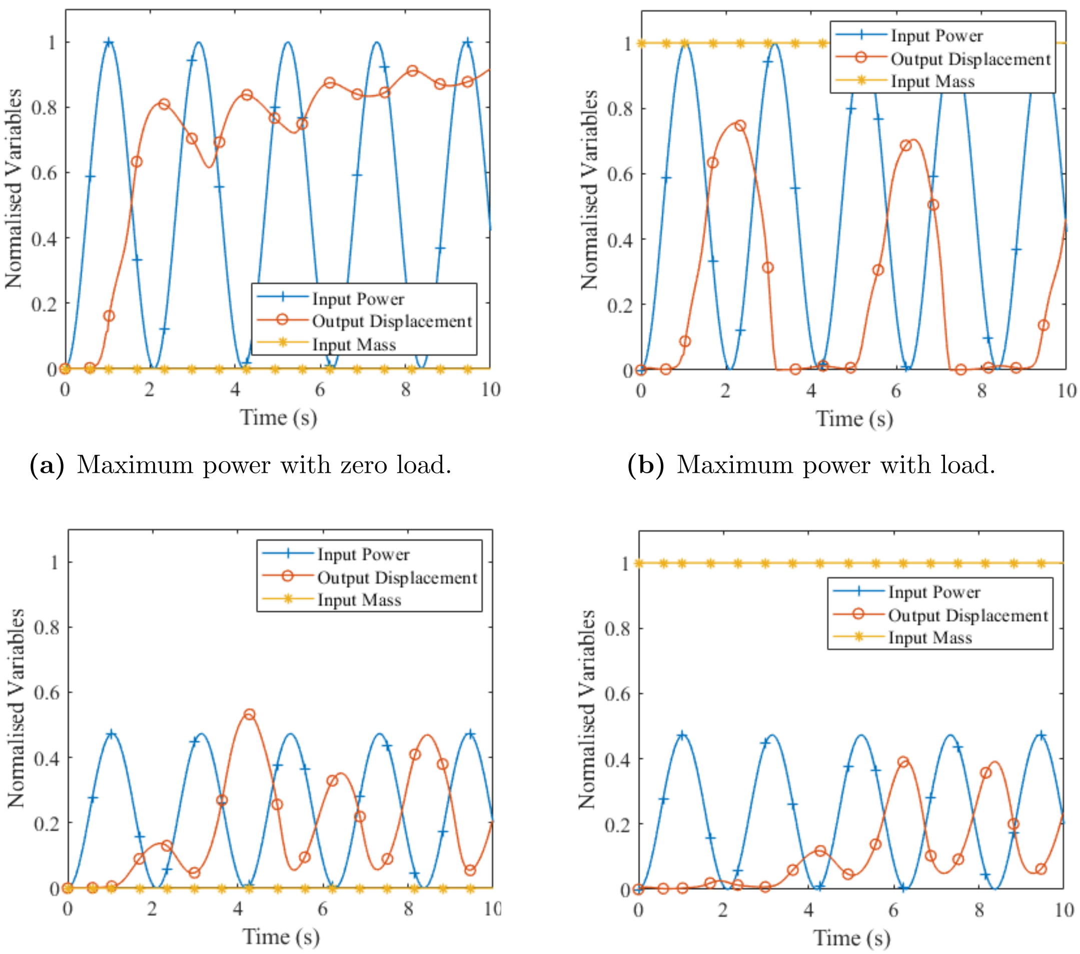
(a) Maximum power with zero load. **(b)** Maximum power with load.

(c) Half maximum power with no load. **(d)** Half maximum power with load.

Figure 5.30: Normalised response due to a pulse train input.

As can be seen, the extra load in Figure5.30d does not interfere with the response of the SMA. This can be attributed to the fact that the time between two successive pulse inputs is much smaller than the time needed for the model to register a response due to the load alone. Hence, a new pulse input kicks in before the system responds to the effect of the load alone.

Figure 5.31a shows the SMA response due to the sinusoidal power input. It can be seen that the response is periodic with the maximum displacements gradually increasing. This increase in maximum displacement can be attributed to the accumulation of residual strains with cycles. Also, it can be observed that the difference between the peaks of successive responses decreases with time.
The reason for this is due to hysteresis stabilisation, which besides reducing the energy dissipated, also improves tracking, and thus makes the SMA actuator more efficient. Figure 5.31b shows the sinusoidal response of the SMA due to the effect of load and the sinusoidal power input. The peaks of the response are due to the effect of the maximum power supply. When this power input starts to decrease, the effect of the load starts to play a major role, thus causing the SMA to extend. The SMA response is the difference between the constant expansion effect of the load and the periodic contraction due to the power input. Also, unlike the response in Figure 5.30b, this response can register the effect of the load on the SMA when the power is switched off. This is because the time between successive power input crests is greater than the system response time when only the load is applied. Also, there is a lag in the SMA response.



(a) Maximum power with zero load.

(b) Maximum power with load.

(c) Half maximum power with no load.

(d) Half maximum power with load.

Figure 5.31: Normalised response due to sinusoidal input.

5.5 Chapter Summary

In this chapter, extensive research was conducted in order to understand the mechanism in which SMA-based actuators respond to different stimulus. With this in mind, three types of models that estimate the displacement of a stabilised SMA wire-based actuator were developed. Based on a theoretical framework, the analytical models use the conservation of energy to model a transient behaviour of the SMA wire. This gives rise to a mathematical relationship between the input energy and stress, to the output strain. The developed FEA model used much more advanced numerical tools to model the SMA actuator by using more performance information. Unlike the analytical model, the FEA model could also simulate the SMA actuator within its plastic region. In the analytical model, this would require much more complex mathematical equations which would be too complex to manipulate. Finally, based on a computational Black-Box approach, a NARX NN model was developed to properly represent the highly coupled, and time-variant SMA actuator system. Results from the three models showed that the NARX NN model was the most accurate in predicting the input-output relationship of the SMA actuator.

Chapter 6

Simulation-Based Modelling of an Anthropomorphic Robotic Prosthetic Hand

Towards a Kinematic and Dynamic Robotic Prosthetic Hand Model

It is well known that accurate modelling of an RPH is foundational to the development of a simulation-based RPH control system. Traditionally, an RPH is modelled analytically without the consideration of trajectory planning and contact modelling. As such, this chapter aims to develop an accurate anthropomorphic RPH model that forms part of the RPH control system and takes into account the properties of motion and force, whilst remaining simple to implement. Using a combination of programs and algorithms, the resultant 10 DoF anthropomorphic RPH model effectively expresses the forward and inverse kinematics. Furthermore, the set-points and logarithmic spiral trajectories for various grasps are representative and presented alongside a linear contact model, link relationship and passive elements.

6.1 Theory and Literature Review

A human hand is a perfect example of a complex yet effective system consisting of 27 bones and joints divided between the wrist, palm and fingers. Figure 6.1a shows the skeletal structure of a human hand with 23-DoF [216], while Figure 6.1b shows a simplified 10-DoF RPH (this dissertation). Also, Figures 6.1a and 6.1b present the joints of a hand represented mechanically as hinge joints; however, other joints such as a saddle or ball and socket joint are also common [217]. Based on the types of joints and actuation, it is possible to devise an anthropomorphic RPH which is capable of much of the dexterity of a human hand. Apparent in Figure 6.1a is the fact that a finger consists of three joints except for the thumb. Primarily ensuring grasp stability, these joints can flex, extend and in the case of the MCP joint, abduct and adduct. The joints of a finger are composed of passive elements which give them a natural tendency to return to an equilibrium or rest state. In relations to an RPH, these passive elements are commonly mimicked with the inclusion of springs and dampers.

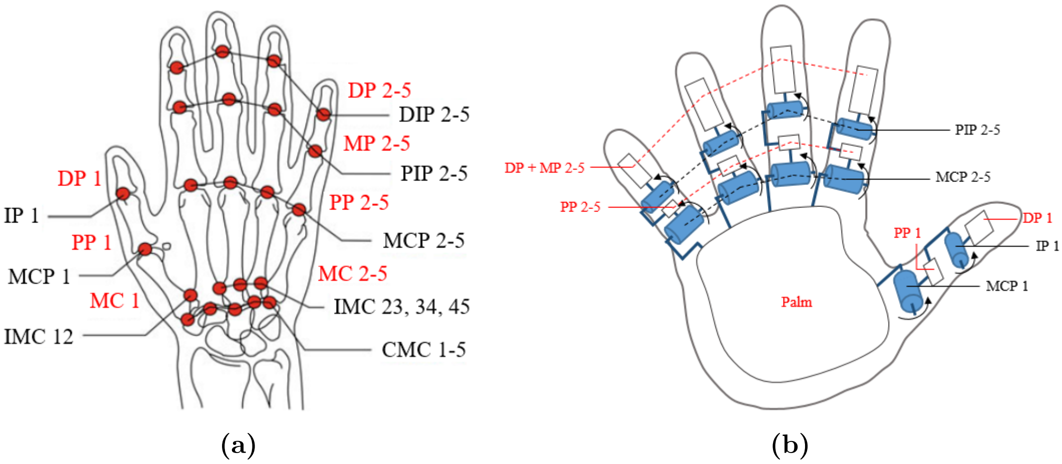


Figure 6.1: The representation of anatomical joints and bones in an anthropomorphic RPH.

The modelling of an RPH should not be limited to the movement of the various phalanges and should therefore include the associated complexities surrounding the contact between the fingers and GO. In this chapter, contact modelling is instituted at nine key locations (see Chapter 4) through primitive shapes using a linear Kelvin-Voigt model [218] and as such, this section details some of the associated concepts. Furthermore, the physical interpretation of contact sensing through an FSR is given. RPH models are conventionally reduced to a 2D problem in which only a single finger is analytically analysed. Subsequently, with increased difficulty, these models are extended to form complex multi-finger models. This complexity is compounded by stability requirements and the fact that most RPHs are underactuated.

6.1.1 Overview of existing RPHs

Mimicking biological structures has garnered significant interest from humans over the past millennia. This interest has led to the advancement of prosthetic hands over the years progressing from early works such as the prosthetic iron hand of Götz von Berlichingen [219] in 1505, to the dexterous shadow hand from 2005 [220]. The progression of prosthetic hands saw the improvement of physical, actuation and kinematic properties [221], with a drive towards electrically driven RPHs. The use of robotic hands is primarily focused on prosthetic and rehabilitation applications, though, it also appears in industrial and human-robot applications.

Judging by the review by Piazza *et al.*, [222] the number of RPHs has increased exponentially over the last century. This is linked to the increased uptake of rigid joints and actuation coupled with underactuated transmission properties. Piazza et al., also noted that there seems to be an increased exploration of simple soft-robotic hands which ensure robustness, efficiency, adaptivity, safety, design formalism, control simplicity and natural motion. As shown in Figures 6.2a to 6.2f, there have been several attempts at developing a dexterous RPH, both in the commercial and academic space.

One further example of this is the three-fingered RPH developed by Zollo *et al.*, [223]. The DC motor-actuated hand makes use of a thumb, index and middle finger to ensure triangular stability. This greatly reduces the control complexity and the number of actuators required, however, the lack of finger movement means that certain anthropomorphic qualities cannot be achieved. Each of the fingers consisted of three hinge joints with an added joint at the base of the thumb allowing for the adaptive grasping of various sized objects. Therefore, in total there were 10-DoF, with the thumb actuated by two DC motors while the index and middle fingers were each actuated by one DC motor. Lastly, Zollo *et al.*, [223] found that simulation-based optimisation of the trajectory planning of one finger led to a more human-like hand movement.

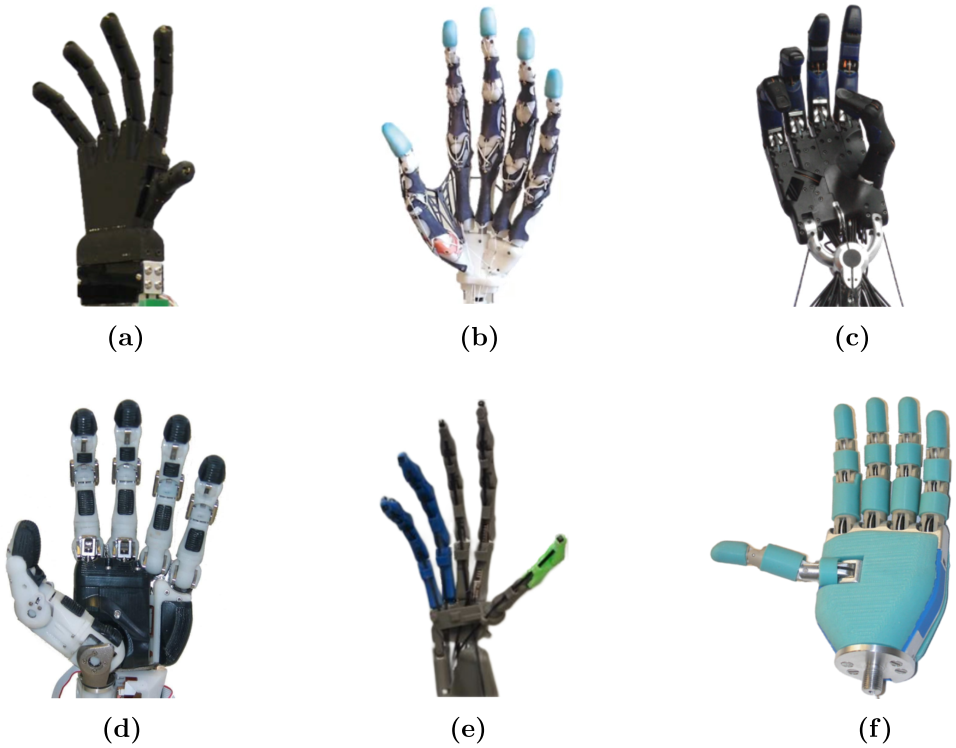


Figure 6.2: Existing state of the art humanoid hands. Figures 6.2a to 6.2f show the SMA actuated MIT hand [224], biomimetic University of Washington hand [225], Shadow C3 pneumatic hand [226], Elu2 anthropomorphic hand [227], hybrid actuated hand [228] and underactuated hand from Laval University [229].

It is argued by ten Kate *et al.*, [230] that the increased output of RPHs is due to the improvements of rapid prototyping such as 3-D printing. This factor, coupled with open-source RPH initiatives, has sparked interest in the broader robotics community. Apart from adopting rapid prototyping as a means of production, Adrianesis *et al.*, [231] used an unconventional SMA-based actuation system (similar to the one in this dissertation). The actuation system was linked to a network of cables which formed the transmission system. Interestingly, only the MCP and PIP joints of each finger were actuated, leaving the DIP joint to provide compliance via a bias spring located at the joint. Omitting the actuation of the DIP joint was based on an analysis of the workspace, where Adrianesis *et al.*, found an insignificant change in the ROM. Important to the work conducted in this chapter, Adrianesis *et al.*, [231] found that at lower angles of rotation the fingertip force remained at a low of 1.7N, while at higher rotation angles the force doubled. The fingertip forces were measured sufficiently by use of a PCB-mounted FSR; care was taken to ensure the FSR was mounted on a flat surface. A passive consideration in the modelling of an RPH was pointed out through the use of artificial skin (see Section 6.3). Although the study by Adrianesis *et al.*, was comprehensive, parameters such as hand pre-shaping, force closure and grasp stability and security [231] were not

considered. As highlighted by Price *et al.*, [232], a key consideration for single-actuated fingers lies in the coupling between the different links. This is substantiated by Xu and Todorov [233], who developed arguably the most realistic RPH by introducing polymer-based ligament and capsule derivatives.
Conversely, Parshotam [234] included leaf spring elements in the joints of the developed 10 DoF RPH, which forms the basis of the model presented in this chapter. The RPH uses two coupling links for transmission of the SMA displacement, with FSR tactile sensors located at the fingertips for force feedback. The DIP joints are fixed in a set position, and the thumb is orientated such that an additional joint is not necessary.

In summary, it is of significant benefit to reduce the complexity of an RPH through the number of links and joints. Furthermore, under actuation is an effective means of attaining multi-grasp performance, while compliance is of importance since it mitigates grasp pre-shaping and ensures a larger contact region between the RPH and GO.

6.1.2 Passive elements of a human hand

The human hand contains various anatomical structures that give rise to its passive behaviour. These structures have a natural stiffness and damping that stems from the elasticity, flexibility and compliance experienced as a force imparts a change in geometry. Of concern in this dissertation is the passive elements in the finger joints of a human hand, as shown in Figure 6.3. These primarily consist of ligaments and articular capsules, where each articular capsule is made up of a fibrous and synovial membrane. The fibrous membrane is composed of connective tissue which presents elastic properties and forms part of joints stiffness. On the other hand, the synovial membrane encapsulates synovial fluid which acts as both a passive fluid spring and damper whilst lubricating the articular cartilage.

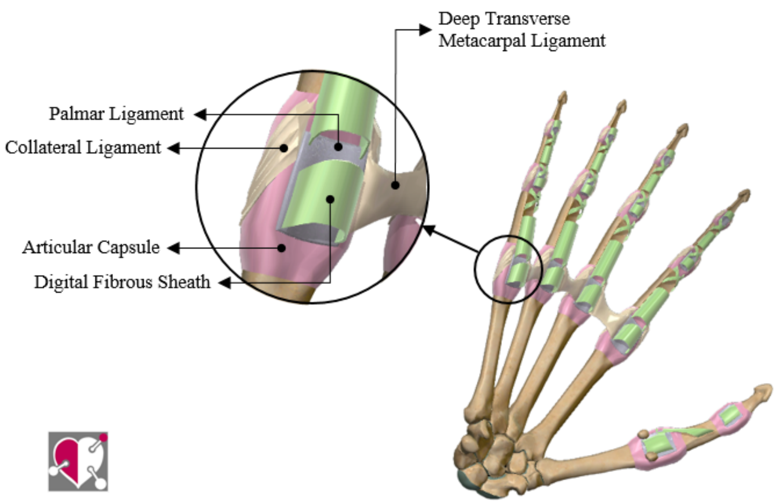


Figure 6.3: Passive structures in the joints of a human hand. Image courtesy of Anatomy Learning, Dr. R. Blanco Salado [235].

As shown in Figure 6.3, the deep, transverse ligaments serve to couple the motion of each finger. This means that flexion of any finger, excluding the thumb, provokes an involuntary flexion of the neighbouring fingers. The coupled motion that originates from these ligaments is important as it permits the natural cupping of the hand for ADL. Apart from these tangible passive elements, friction exists between the articular cartilage and meniscus of each phalanx. Extrapolating the results of Warnecke *et al.*, [236], the coefficient of friction is shown to be less than 0.045; however, due to large forces and a small contact region, friction may still be significant. Friction may also be exacerbated by conditions such as cartilage degeneration or osteoarthritis.

External passive factors such as the elasticity of skin and muscle length also play a role in the behaviour of finger joints. For example, when undergoing flexion, the skin on the palmar side of the hand is in compression, while the skin on the dorsal side is in tension. This collectively adds to the joint stiffness and damping. Muscle length is implicitly accounted for in the model developed in Chapter 4. Joint stiffness and damping are commonly modelled as a combination of torsional springs and dampers, as shown in Figure 6.4.

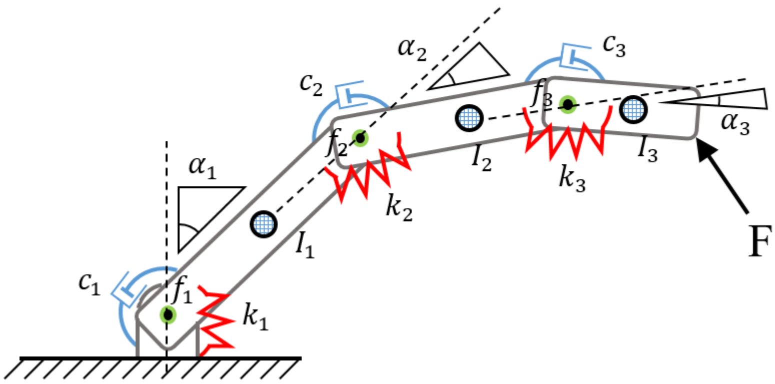


Figure 6.4: Kinematic diagram of the ROM and passive structures in the joints of a finger. [237].

These induce torques at each joint and is governed by Equation 6.1.1.

$$T_n = k_n \cdot (\alpha_n^f - \alpha_n^o) + c_n \cdot (\omega_n^f - \omega_n^o) \tag{6.1.1}$$

Without these passive elements, the simulation of the RPH would result in a perpetual rotation of the finger segments. A method for estimating the torsional spring and damper constants for a joint is presented in work by Oatis [238]. This study focused on the knee joint; however, the same concept can be applied to most joints. An electrogoniometer was used to measure the angular displacement of the knee after the leg was dropped from an extended and relaxed state. From the oscillatory angular displacement, the natural frequency and viscous damping factor were determined. Furthermore, the torsional spring and damping constants were calculated using Equations 6.1.2 and 6.1.3 and resulted in average stiffness and damping coefficients of 4.24 Nm/rad and 0.12 Nms/rad.

$$k = I \cdot \omega^2 \tag{6.1.2}$$

$$c = 2 \cdot \zeta \cdot \omega \cdot I \tag{6.1.3}$$

Highlighted by Oatis was the fact that these coefficients were significantly correlated with the initial angular position of the joint. This suggests that the passive structures change properties depending on orientation, force, velocity, etcetera. When springs and dampers are incorporated into kinematic chains such as a multi-link manipulator (robotic finger), it is subject to buckling if the external force is in a direction which causes uncontrolled deviation [238]. One such condition is seen when a force is applied normal to the fingertip of a finger in a horizontal position. For an RPH actuated in flexion only, buckling may also arise for a dorsally applied force while returning to rest.

Knez *et al.*, [239] investigated the vibratory response of a finger and subsequently performed system identification of a biodynamic finger. The resultant spring and damper coefficients were questionable as they were several orders of magnitude greater than those suggested by both Wright and Johns [240] and Franklin [241]. According to work by both Wright and Johns, and Franklin, the spring stiffness was suggested to lie between 0.01 and 7.25Nm/rad, depending on the joint in question. Chen Chen *et al.*, [242] reported similar values with the addition of a damping coefficient of 0.0225Nms/rad.

A finger model proposed by Sancho-Bru *et al.*, [243] highlighted the need to include the effects of passive force-generating structures for an accurate estimation of force propagation. Also, the passive structures were found to be critical to the behaviour of the finger during fast movements. In general, the passive joint elements serve to couple, transfer and return to equilibrium the forces and phalanx positions. These need to be mimicked in an RPH so that both realistic and functional motion is possible.

6.1.3 Contact modelling

Most RPH systems operate in multi-body contact conditions. Due to the existence of elastic deformation in the contact areas, the elastic properties of the different bodies in contact cannot be neglected. The geometry of the contact area also changes when there is a relative translation and/or rotation between the contact bodies. This condition also cause the load distribution and friction in the contact areas to vary. Therefore, understanding contact region dynamics aids in modelling the behaviour of different objects in contact and at the same time, assisting in controlling the stability and performance of different joint actuators.

When modelling the behaviour of contacting bodies, the elastic behaviour of the bodies is modelled as springs and dampers that act as the passive contact elements. These simplified single degree-of-freedom systems are modelled to investigate the effects of stiffness, etxernal loads, damping and friction coefficient on the dynamics of the multi-body contact areas. Figure 6.5 shows a simplified multi-body linear Kelvin-Voigt contact force model with friction between the contact areas [218]. As can be seen in the figure, both contacting bodies have their own spring stiffness and damping contacts. Also, due to the translation between the bodies on the contact areas, there exists a friction force that is perpendicular to the line of contact.

Figure 6.6 shows the schematic of the underlying physics and resistance analogy on an FSR at different loading conditions. As can be seen, the FSR is made up of two conductive surfaces, a polymer substrate, and carrier particles dispersed randomly in the substrate. Being a passive device, the electrical resistance of the FSR decreases when a compressive load is applied over the conductive surface. This phenomenon, termed quantum tunnelling as depicted in Figure 6.6b, allows for an increase in particle transmission paths through electron bridging within the polymer matrix. Figure 6.6c

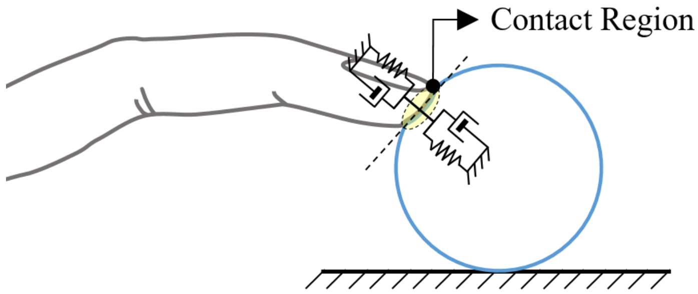


Figure 6.5: Multibody (sphere-sphere) linear Hooke's contact force model with friction [218].

shows the FSR when the load has been removed. As observed, the resistance of the FSR increases again due to an increase in the distance between adjacent carrier particles. However, it is noteworthy to observe that the resistance of the device does not return to its initial state. The reason being, the drift which causes the carrier particles to not return to their initial position results in a different resistance.

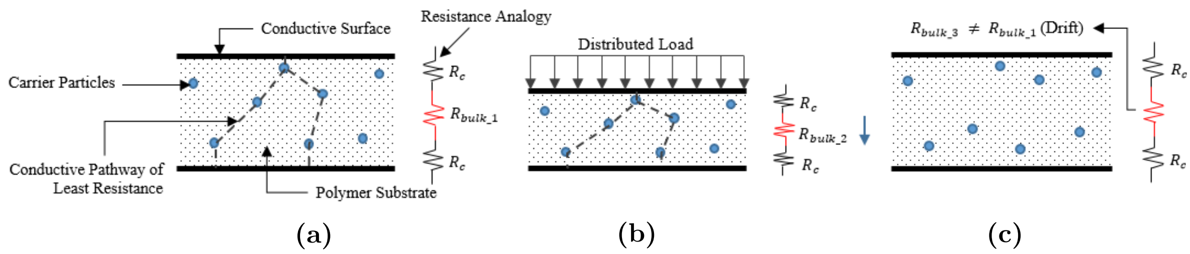


Figure 6.6: A schematic representation of the underlying physics of an FSR. Figures 6.6a and 6.6b show an FSR with no-load and load, respectively. Figure 6.6c shows the physical interpretation of FSR drift. Images adapted from [244].

6.1.4 Static and dynamic RPH models

Understanding the kinematic structure of an RPH is pivotal in establishing joint and contact point conformity during the grasping process. A detailed model of an RPH thus allows fine manipulation of hand postures through actuator force interaction and contact point planning. In these models, proper trajectory planning of the RPH might be computationally expensive as it is based on solving an optimisation problem and may also sometimes require knowledge of the object's shape. To avoid this problem, Yang *et al.*, [245] developed an underactuated robot hand that requires less computation. In their study, they developed a model that uses three actuators to drive fifteen joints of a robotic hand. The intuitive control of the actuators was accomplished by directly applying the EMG pattern recognition method from the human hand gestures to mimic the joint movements of the robotic hand. However, with these types of under-actuated models, some joints are not able to freely move, which may result in less functionality.

A model of an SMA-actuated MCP joint of an RPH is shown in Figure 6.7. This joint is actuated using two spring-biased SMAs that work as an agonist-antagonist muscle pair. This muscle pair is responsible for the actuation of the abduction/adduction and flexion/extension of the metacarpophalangeal joint. These SMA-based actuators are connected to the finger model by extensor and flexor tendon cables at points B and

C. The opposing contractile actions of the two SMA-based actuators enable the finger model to rotate about the MCP joint (point O).

A schematic of the external forces acting on the Proximal Phalanges is shown in Figure 6.7b. The forces due to the weight of the finger and the lower actuator enable the finger to rotate in a clockwise direction, whilst the upper actuator causes an opposing rotation of the finger. By applying the equations of motion and equilibrium relations to this system, together with both inverse dynamics and inverse kinematics, the position of the joint can be obtained and related to the actuator force. This allows for an optimal trajectory planning of the model.

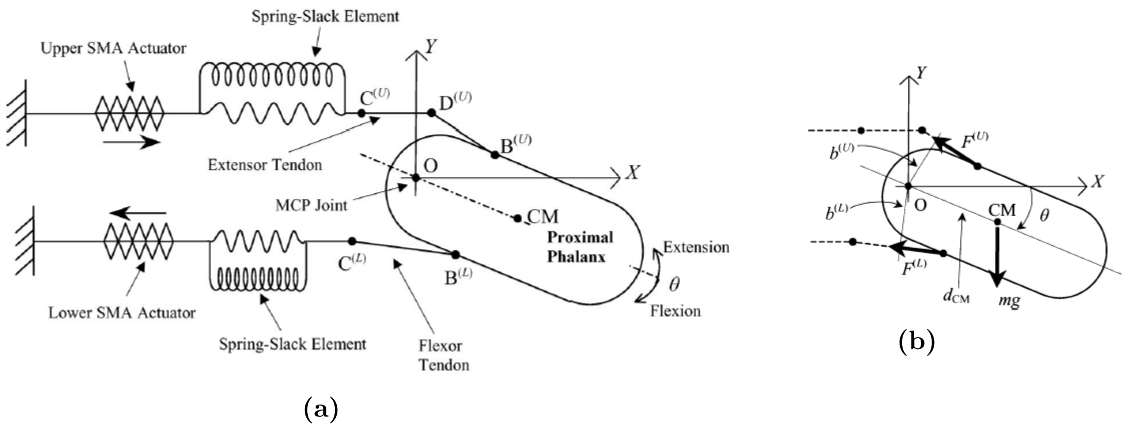


Figure 6.7: SMA-actuated MCP joint model for an RPH. Figure 6.7a shows the mechanical coupling between the joint and actuation system [246]. A free-body diagram of the PP is shown in Figure 6.7b [247].

Figure 6.8 shows the dynamic coupling of an SMA-actuated two degrees-of-freedom finger model. As shown in the figure, this model consists of a base plate onto which the finger base is connected through MCP joint 1. Just like the human hand, this base plate represents the palm. The finger base is then connected to the finger at the PIP joint. In order to accomplish flexion/extension, the finger base and finger are connected to the SMA-based actuator through link 1 and link 2. Link 1 connects the finger base to the SMA-based actuator. This link can accomplish both planar rotation and translation. This link displacement causes the finger base to rotate about the MCP joint. It can be observed that this rotation of the finger base causes the finger to rotate by just the same amount due to their connection at the PIP joint. The finger is also connected to the base plate through link 2. This link rotates about the fixed connection at the base plate, which results in further finger rotation in order to accomplish full flexion during grasping.

The coupling shown in Figure 6.8 is based on a slider-crank mechanism to enable movement of the finger. At any instant, the position of the finger is always a function of the angular positions of both link 1 and link 2.

Rotation of link 2 allows the model to be able to grasp objects of different sizes, by achieving dexterous grasping sequence. Bandara *et al.*, [248] developed an underactuated RPH model with self-adaption ability. In their study, they modified the cross-bar linkage mechanism by using length varying links in place of rigid links. These new links had a prismatic joint and a spring which allows them to vary their lengths. In doing so, the

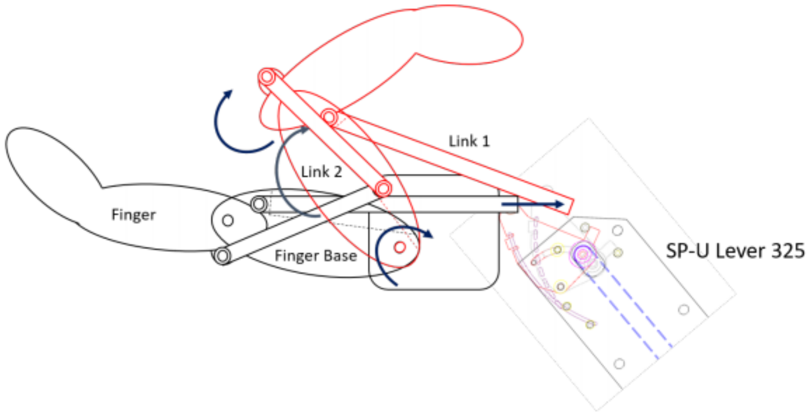


Figure 6.8: Dynamic coupling of an SMA-actuated two DoF robotic finger mechanism [234].

passive prismatic joints enabled the fingers to accomplish the different joint angles to grasp different object shapes.

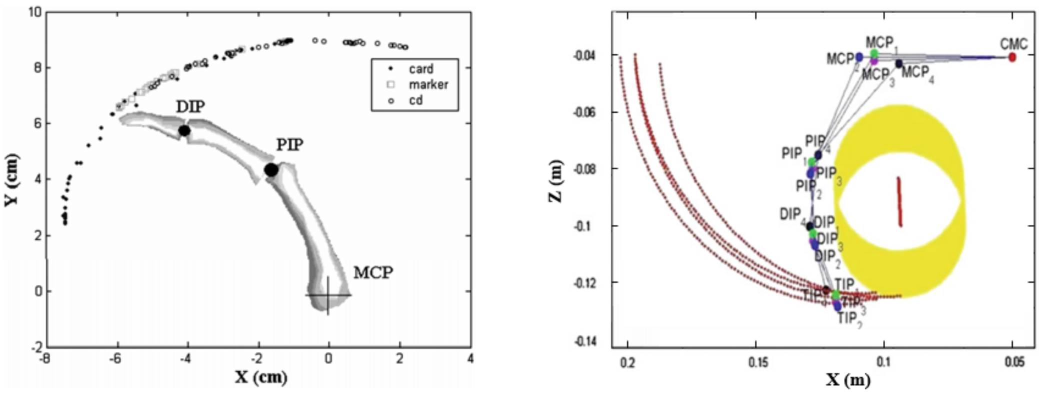
6.1.5 Grasp trajectory planning

The ability to accurately mimic the operation and trajectory features of a human hand has become a growing interest in the control of robotic hands. Several studies have been conducted to develop models that can predict the dynamic and kinematic behaviour of robotic hands. These studies are all based upon an understanding of the force transmission mechanism from the tendons to the skeleton. Armstrong and Chaffin [249] developed a model that predicts the relationship between joint finger positions and tendon displacements. In their study, they evaluated the Landsmeer joint-tendon mechanic behaviour during gripping and pinching using different hand sizes by developing a model to predict finger movement empirically.

The motion of the human finger can be seen as a combination of several joint trajectories occurring simultaneously to accomplish a certain pre-determined grasping task. Applied to a robotic hand, this mechanism increases model complexity due to the requirement of different control schemes for each joint movement. Figure 6.9a shows the fingertip trajectory for three different grasping tests. In the figure, the centre of rotation of the finger is located at the MCP joint. In this planar model, the x-axis is perpendicular to the palm, while the y-axis is aligned with the first metacarpal. The negative x and y values denote fingertip rotation in a counter-clockwise direction which is towards the hand palmer side. Figure 6.9b shows the logarithmic spiral trajectory of the fingertips during cylindrical grasping and the corresponding joint positions for the different fingers. All the fingers of the hand follow a similar path except for the thump, which will be pointing in the orthogonal y-axis during this grasping process.

6.2 Methodology

A multistage methodology was adopted in this section, with the first stage focusing on 3-D modelling of the RPH hand using Autodesk inventor. In this stage, joint types and model materials were defined, together with the associated constraints and boundary



(a) (b)

Figure 6.9: Logarithmic spiral trajectory of the finger tip during grasping. Shown in Figure 6.9a is the end-point trajectory of an index finger during a tip pinch [250]. A similar trajectory is shown for the cylindrical grasp in Figure 6.9b [251].

conditions. The next stage involved converting the 3-D Autodesk model into a corresponding Simscape block diagram, as shown in Figure 6.10. This is broken down into different sub-systems for simulation purposes.

Figure 6.10: Simscape model.

Before carrying out any simulations, all system parameters were defined, and the model initialised depending on the grasp to be performed. In the fourth stage of the methodology, contact modelling of elementary body pairs representing the RPH and the GO interaction was performed. Next, using joint angles for various grasps, together with the forward kinematic model, the path followed by each fingertip was established. In the final stage of the methodology, through the use of the FSR and optical encoders, the applied force and angular position of the fingertips were simultaneously recorded during the various grasps.

Figure 6.11 shows the full control system architecture implemented for the modelling of the RPH. Depending on the grasping being performed, the model parameters are initialised in the Simscape simulation environment. As can be seen, depending on the grasp to be performed, the target force at the contact and joint initial angular position are first calculated. Together with the output from the FSR and encoder, these are passed to the comparator, which then estimates the error before passing them to the FLC.

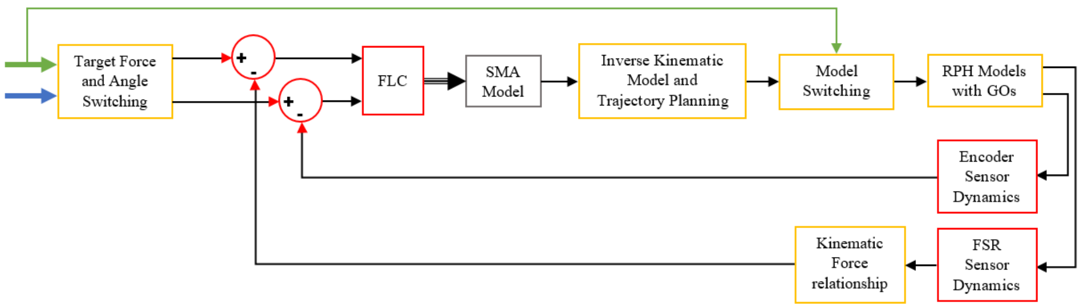


Figure 6.11: Control system architecture.

6.2.1 3-D model and constraints

The RPH 3-D model consisted of 12 parts related to the digits as well as the palmar and dorsal side enclosures. The parts were developed and assembled in Autodesk Inventor R2018 according to the mean hand dimensions specified by Chandra et al., [252] (see Appendix for working drawings). The grasp items presented in Chapter 3 were placed in the scene at a proximity reflective of grasping. Related to the work by Parshotam [234], the material of the entire model was specified as homogenous Polylactic Acid (PLA) with a density of 1.25g/cm^3. The 3-D RPH model excluded fasteners, however, maintained the effect of fasteners through the imposed RPH model constraints.

Revolute constraints were imposed on the following digit joints: MCP 1-5, PIP 2-5 and IP 1. Furthermore, the joints were restricted from translating in a direction perpendicular to the rotational plane. Constraints were used to fix the palmar and dorsal enclosures to one another. The addition of silicone or latex-based skin was excluded from the RPH model due to the associated complexity in modelling. Despite this, the passive effects of skin on the digit joints were included (refer to Section 6.1.2 for more details). As previously mentioned, the anthropomorphic RPH model was developed without distal joints in mind (see Figure 6.1b).

6.2.2 Simscape multibody representation

For simulation purposes, the 3-D RPH model was converted to a Simscape block diagram equivalent. This was done by firstly installing the Simscape Multibody plug-in for Autodesk Inventor and linking it to MATLAB via a command-line input. After installation, the 3-D model assembly was exported, resulting in the equivalent Simscape model. Notably, before exporting the 3-D model, it was orientated such that the hand was in a set position and relative to the global reference frame (see Figure 6.10). Upon conversion, the individual 3-D model parts were converted to a STEP format, and an XML data file was created. The XML data file hierarchically contained the orientation and material properties. The Simscape model consisted of 12 custom body elements interconnected with corresponding revolute joints and rigid transforms. After breaking the Simscape model up into various sub-systems and re-routing, the process of converting the 3-D model to a Simscape model was repeated. The repetitions were for each grasp separately, since, each grasp involved a different GO which needed to be placed in the scene. It is important to note that the control system revolves around the grasping of statically stable GOs. Therefore, each GO was constrained, which consequently equated to the GO being rigidly fixed to the world frame in the Simscape model. Apart from the world frame, a mechanism configuration block was present which stipulated the gravitational parameters.

6.2.3 Model parameters and initialisation

As previously noted, an XML data file was automatically created which stored the initialisation parameters including the mass, centre of mass, moment of inertia and the product of mass for each of the model parts. Furthermore, parameters such as the equilibrium angle of each joint and the orientation of each rigid transform were also stored. The loading of the parameters was done by using various structured variable names in the Simscape model and linking this to a model initialisation callback, which loaded the XML data file. The joints were set to be actuated according to the angular rotation, and the torque at each joint was automatically computed. This required further sub-system routing, since, the angular rotation was the input to the RPH simulation model. Additional to the actuation, the internal mechanics and limits were set. Specifically, this meant that spring stiffness and damping coefficient properties were analytically estimated respectively in Equation 6.1.2 and 6.1.3, after considering the passive components included in the RPH by Parshotam. Also, a trial and error approach was used to refine the values prescribed to the internal mechanics. The joint limits were initially adjusted according to the specifications of Gellman and McKellop [253]; however, they were subsequently revised due to the transmission link limitations (see Section 6.1.4). In mechanism configuration, the magnitude and direction of the gravitational constant were initialised. Lastly, the solver was set to ODE15s with a maximum step size of 0.05s to reduce the prevalence of a jerky simulation. Other parameters exist and are discussed in the subsequent paragraphs.

6.2.4 Contact model

To provide force feedback, it was necessary to perform contact modelling on the Simscape RPH model. This included the incorporation of elementary bodies pairs at specific locations noted in Section 6.1.3. These pairs consisted of sphere-tube, sphere-sphere and sphere-plane bodies, representing the interaction between the various GOs and the RPH. Using the Simscape multibody contact force plug-in, the elementary bodies were placed at the associated positions. The positions were established by creating a new reference frame at the pulp of each fingertip and four locations in the palm of the RPH.

Furthermore, to maximise the contact surface between the RPH and GO, the positions were chosen as per the touch test. Thereby reducing the closure force and improving grasp security. The contacts were linearly modelled according to the Kelvin-Voigt contact model with stick-slip friction. Following the recommendations of Flores and Lankarani [218], the contact stiffness and contact damping were set to 10000N/m and 100Ns/m, respectively. These parameters were iteratively lowered depending on the behaviour of the RPH model, i.e., for exploding variables. Some attempts at estimating the contact stiffness and damping were made, however, it remained challenging since these parameters depended on the current and deformed shape of the objects in contact, the thickness, surface finish and other factors. The static coefficient of friction used was determined experimentally by Howard [254] for PLA and was slightly reduced to represent the kinetic coefficient of friction. There were several outputs from the contact model, such as the friction force, penetration depth, etcetera; however, only the normal force was of interest. Apart from the contact model, the transient behaviour of the FSRs in the actual RPH model was accounted for.

6.2.5 Model simulation switching

The procedure of model switching involved the selection of an appropriate RPH model based on the grasp classification (see Chapter 3). After selecting the appropriate model, corresponding MCP and PIP joint angles were prescribed in a MATLAB function. These angles were determined from the 3-D RPH model in Autodesk Inventor. Specifically, with the contact solver on, the fingers were positioned such that they were in contact with the GO in a manner indicative of actual grasping. The MCP and PIP angles were measured using reference frames from both the joints and the global coordinate system. Rather than providing the final joint angles, these measurements served to locate the fingertip and were updated according to the recommendation previously suggested. Apart from the joint angles, the model switching block also allocated the target forces. As an example, for the 3 jaw chuck pinch, the target forces were specified for the thumb, index and middle fingers only. These changed with the different grasps as per the touch test undertaken in Chapter 4. Notably, according to the definition of the eighth class in Chapter 3, the RPH targets were set to a rest position. This was done such that any irregularities in the classification would not permeate further into the RPH model. Similarly, the grasp targets were set to rest for the grasps not currently active. In other words, for a 3 jaw chuck pinch, the joint angles and forces for the remaining grasps would be set to rest. By doing this, the task of trajectory planning was vastly simplified as it did not need to keep track of the previous hand state.

6.2.6 Trajectory planning

The trajectory planning of the RPH fingers superseded the determination of the MCP and PIP angles for the various grasps discussed in the previous paragraph. From the MCP and PIP angles, the coordinate of each fingertip was numerically solved using the forward kinematic model of a two-link manipulator. These coordinates were then passed through the inverse kinematic model, which yielded two solutions for both the MCP and PIP angles. The optimum solution was selected based on joint limitations. After establishing the end-point MCP and PIP angles, the fingertip coordinates were again computed, and a logarithmic spiral trajectory applied. The logarithmic spiral trajectory built on from work by Kamper *et al.*, [250] and was computed iteratively for a user-specified step-size. This resulted in the fingertip coordinates, which followed the trajectory and was used in the inverse model to compute the dual solution evolution of the MCP and PIP joint angles.

Unlike the previously selected angles, these considered the fact that the PIP joint angle can not be negative; however, this did not apply to the MCP joint angle. A major factor in whether the trajectory was practically possible came from the limitations imparted by the links which joined the phalanges. Two-link models were derived and numerically solved to produce the relationship between the linear translation of the SMA actuator and the rotation of the MCP and PIP joints. The relationship was described by the sum of sines using the curve fitting toolbox in MATLAB. Both the trajectory planning and link models formed part of the augmentation of the input to the RPH model. The code for both the trajectory planning and the link models are presented in the digital appendices.

6.2.7 FSR sensor model

The FSRs and encoders located at the fingertips, palm and joints were the only feedback sensors used in the RPH model. Since the RPH model had outputs that were strictly the same as those required, i.e., force and angular position, the existing transient models could not be used. Instead, it was assumed that the sensors only served to impart lag on the system. Based on the results from Saadeh *et al.*, [255], a time constant was estimated and used to produce a first-order transfer function reflective of the lag in the FSRs. This essentially behaved like a low-pass filter. The same could not be said for the optical encoders as they exhibited very limited transient behaviour; however, depending on the resolution incurred a noticeable period of delay was seen. This was simulated by establishing a buffer with a delay of one clock cycle of 80ms. Other than these modifications, saturation was applied to both data streams to limit the output based on the maximum force of the FSRs (10N) and angular velocity (31.42rad/s). These outputs were then fed back to determine the error, which was subsequently the input of the FLC discussed in Chapter 7.

6.3 Results and Discussion

This section dissects the simulation results from the multi-stage modelling of an anthropomorphic RPH model. The findings are based on the methodology described in Section 6.2. The results obtained by simulating the prehensile movement of the 10DoF RPH model associated with various everyday tasks are shown in Table 6.1. The RPH model in Figure 6.1b was used to simulate the hand movements in Figure 3.2. The different joints of the RPH model represent different joints types of a human hand and are named accordingly.

For the 3 Jaw chuck pinch, it can be seen that all the joints are inflexion, except for the thumb IP joint which is in extension. In Figure 3.2, it can be seen that for the proper balancing of the grasping action, the IP joint has to rotate away from the GO. Also, from Figure 3.2, it can be seen that during this grasping process, both the ring and pinkie fingers do not make contact with the GO. The flexion of the MCP joints and subsequently, the PIP joints of these fingers is due to the connection between the

MCP joints of all the fingers, which causes involuntary flexion. These connections act as the transverse ligaments that run across the palmer side of the finger connecting the heads of the MCP joints which are responsible for provoking involuntary flexion of neighbouring fingers. Also, the PIP joint of the ring and pinkie fingers are the largest because there is not an object to grasp that might limit or block their trajectory motion; hence they are in full flexion.

Again, just like the 3 Jaw chuck pinch, when performing the cylindrical grasp, all joints are inflexion, except for the IP joint of the thumb. The reason for this follows the same as the reasoning discussed in the previous paragraph. Furthermore, it can be seen that the MCP joints of both the thumb and pinkie fingers only rotate by a very small amount. The reason for this is that, during this grasping procedure, their MCP joints act more of pivot points for balancing the object. The other fingers then act to grip the object in place firmly and thus have large MCP joint angles. Furthermore, these high joint angles consequently lead to larger PIP joint angles.

When performing the Hook grasp, it can be seen that the MCP joints are almost in full flexion as they rotate towards the palmer side of the hand. Together with the rotation of the PIP joints, this coupled motion allows the finger to form a concave-upwards shape that enables the GO to hang from the RPH. The thumb finger is not necessarily required for full grasping of the GO, and therefore, does not have to rotate. However, the very small involuntary rotation of this joint is due to the connection between its' MCP joint and that of the index finger, which triggers rotation.

When performing the lateral pinch grasping, it can be seen from Figure 3.2 that only the thumb and index fingers are in contact with the GO. However, the other MCP joints are triggered to involuntarily rotate as they are connected to the index MCP joint. This then causes the PIP joint of the respective joints also to be inflexion. However, the IP joint of the thumb is in flexion to allow the GO to balance in the final grasping position. During the rest position, no joint is triggered to rotate; hence all the joint angles are zero. For the final tip pinch grasping, only the tip of the thumb and index fingers are in contact with the GO. During this grasping, the index finger mainly responsible for holding the object in place and thus has a large total arc of finger flexion. The thumb IP joint is in flexion to enable the GO to balance in the equilibrium position. Again, the connecting between the index MCP joint to the rest of the fingers cause involuntary flexion of the other fingers.

Table 6.1: Set-point Joint Angles.

Finger	Joint 1	Angle 1 (°)	Joint 2	Angle 2 (°)
3 Jaw Chuck Pinch Joint Angles				
Thumb	MCP	40.12	IP	-11.79
Index	MCP	37.29	PIP	36.29
Middle	MCP	42.66	PIP	31.11
Ring	MCP	34.44	PIP	50.94
Pinkie	MCP	29.17	PIP	43.09
Cylindrical Grasp Joint Angles				
Thumb	MCP	3.78	IP	-7.25
Index	MCP	21.05	PIP	41.79
Middle	MCP	23.17	PIP	52.76
Ring	MCP	10.02	PIP	57.59

Continued on next page

Table 6.1 – *Continued from previous page*

Finger	Joint 1	Angle 1 (°)	Joint 2	Angle 2 (°)
Pinkie	MCP	4.13	PIP	32.48
		Hook Grasp Joint Angles		
Thumb	MCP	0.81	IP	30.63
Index	MCP	71.71	PIP	53.95
Middle	MCP	70.47	PIP	64.77
Ring	MCP	70.29	PIP	68.91
Pinkie	MCP	71.80	PIP	40.70
		Lateral Pinch Joint Angles		
Thumb	MCP	35.29	IP	-3.44
Index	MCP	54.15	PIP	15.29
Middle	MCP	42.26	PIP	54.66
Ring	MCP	59.75	PIP	49.35
Pinkie	MCP	60.11	PIP	39.96
		Rest Joint Angles		
Thumb	MCP	0	IP	0
Index	MCP	0	PIP	0
Middle	MCP	0	PIP	0
Ring	MCP	0	PIP	0
Pinkie	MCP	0	PIP	0
		Spherical Grasp Joint Angles		
Thumb	MCP	4.06	IP	6.28
Index	MCP	56.87	PIP	27.43
Middle	MCP	52.16	PIP	27.99
Ring	MCP	49.67	PIP	28.17
Pinkie	MCP	55.42	PIP	28.68
		Tip Pinch Joint Angles		
Thumb	MCP	30.40	IP	-1.67
Index	MCP	40.35	PIP	42.30
Middle	MCP	42.43	PIP	47.81
Ring	MCP	11.18	PIP	36.25
Pinkie	MCP	7.94	PIP	7.40

Figure 6.12 shows the variation of the PIP angle with MCP angle. As can be seen, for every MCP joint angle, there exit four possible PIP joint angles. Of these four possible results, two of them occur during flexion, and the other pair is for when the joint is in extension. The hatched areas represent the range of PIP joint angles at a specific MCP joint angle. Accordingly, the upper and lower bounds represent respectively, the achievable maximum and minimum PIP joint angles at a given MCP joint angle. From this, it is worth noting that the maximum joint angles achievable by both joints indicates how large the total arc of flexion will be.

Furthermore, a larger range of possible PIP joints signifies high adaptability. This means that the RPH will be able to grasp objects of different shapes by varying the PIP joint angles within the prescribed limits to ensure full contact with the GO. However, in this dissertation, the RPH model was used to grasp only a single-shaped object. With this in mind, only one PIP joint angle was required for each MCP joint for full grasping. By examining the results in Figure 6.12, it can be seen that at an MCP joint angle of 37.29°, the required PIP joint angle for full grasping of the object will be 73.58°.

Figure 6.13 shows the ROM of the fingertip on the links' workspace when undergoing a single continuous flexion/extension cycle. Figure 6.13a shows the desired ROM with

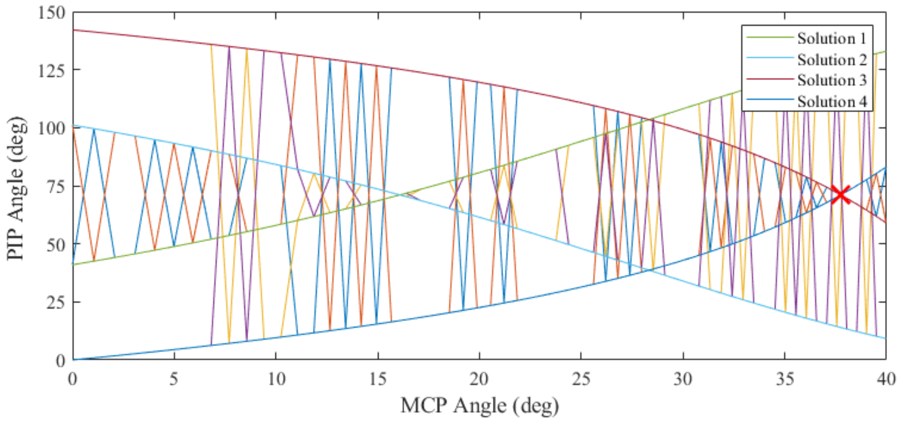
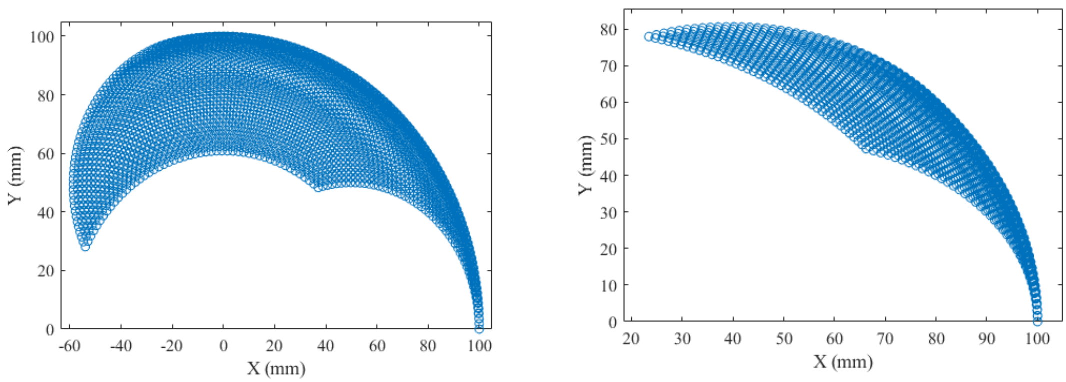


Figure 6.12: Variation of the PIP angle with MCP angle.

a large degree of adaptability. The ROM of the developed RPH model is shown in Figure 6.13b. It can be observed that the two results follow a similar loop, and thus the model developed in this dissertation was sufficient to perform ADL. However, the model developed in this dissertation could not exactly achieve the desired ROM depicted in Figure 6.13a. This was due to the model not having the extra DIP joint which enables extra flexion during grasping, thereby increasing the fingertip ROM. Though the addition of an extra linkage (distal phalange) would increase the ROM of the mode, the model would become too complex to model accurately. Also, adding an extra link would add extra complexity to the design of the model. With that in mind, by further examining the fingertip trajectory during extension only, both the results are almost identical. This is because, during extension, the angle of the DIP joint between the distal and proximal phalanges is zero. Hence, the human finger approaches a two-link model similar to the one developed in this dissertation. This then results in the observed extension trajectory making the model good approximation to mimic the movement of the human hand during grasping.

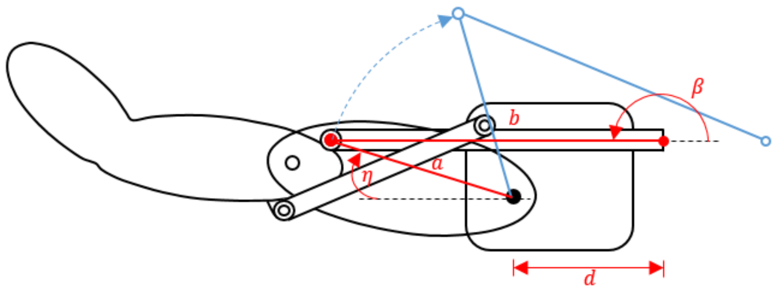


g

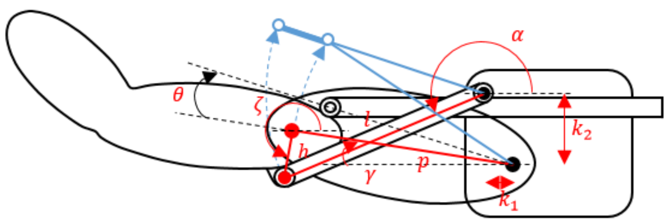
(a) Desired. **(b)** Actual.

Figure 6.13: Fingertip range of motion (ROM) during grasping.

Shown in Figure 6.14 is the relationship between the link position and the joint angles during grasping. At this stage, based on Figure 6.8, it should be noted that the angular position of the fingertip is a function of both the angular position of the MCP joint and the rotation of the finger about the PIP joint. These angular positions are determined from the kinematic study of link 1 and link 2. From Figure 6.14a, SMA-based actuator activation pulls link 1. Due to the connection of link 1 to the finger base, it then follows a slider-crank motion. Furthermore, the finger base will rotate about the MCP joint. Equation 6.3.1 shows the horizontal displacement of link 1 as a function of both its angular position and that of the finger base.



(a) RPH at rest position.



(b) RPH in flexion.

Figure 6.14: Kinematic mechanism of link position and joint angle during grasping.

$$d = a \cdot cos(\pi - \eta) - b \cdot cos(\beta) \tag{6.3.1}$$

$$\beta = \pi - arcsin(\frac{a \cdot sin(\pi - \eta) - c}{b}) \tag{6.3.2}$$

Figure 6.14b shows the relationship between the angular position of link 2 and the finger base during flexion of the RPH model. As previously described in Section 6.1.4, the position of the finger is a function of two mechanisms. Firstly, the rotation of the finger base about the MCP joint, which causes the PIP joint to rotate by the same amount since the linkage is a rigid body. Secondly, the rotation of link 2 about the base plate resulting in a relative rotation of the finger about the MCP joint. Equations 6.3.2 - 6.3.5 show the kinematic relationships involved during finger flexion of the developed RPH model. Using the regression parameters in Table 6.2, the performance of the RPH model was calculated using Equation 6.3.6.

$$h = \sqrt{(-k_1 + l_1 \cdot cos(\alpha) - l_2 \cdot cos(\pi - p))^2 + (l_1 \cdot sin(\alpha) - l_2 \cdot sin(\pi - p) + k_2)^2} \tag{6.3.3}$$

$$-k_1 + l_1 \cdot cos(\alpha) - h \cdot cos(\zeta) = l_2 \cdot cos(\pi - p) \tag{6.3.4}$$

$$p = \eta + \gamma \tag{6.3.5}$$

$$y = a_1 \cdot sin(b_1 \cdot x + c_1) + a_2 \cdot sin(b_2 \cdot x + c_2) \tag{6.3.6}$$

Joint	$\mathbf{a_1}$	$\mathbf{a_2}$	$\mathbf{b_1}$	$\mathbf{b_2}$	$\mathbf{c_1}$	$\mathbf{c_2}$	$\mathbf{R^2}$	**NRMSE**
MCP	1.3700	0.4524	0.3112	0.4909	-0.2316	2.3680	0.9992	0.0038
PIP	3.3230	1.1420	0.1333	0.2932	-0.1079	2.8260	0.9999	0.0015

Table 6.2: Performance of regression models.

Figure 6.15 shows the variation of joint angle with joint displacement for both the MCP and PIP joints. From the figure, it can be observed that as the joint displacement increases, so does the joint angles. The reason being that, during grasping, the actuation of the finger MCP joints cause them to displace through flexion towards the palmer side of the hand. This simultaneously causes a corresponding increase in the joint angle. Because of the linkage between the MCP and the PIP joints, displacement of the MCP joint will also cause displacement and subsequently flexion of the PIP joint. With the MCP joint assumed to be directly connected to the origin of the coordinate system and the links modelled as rigid bodies, the MCP joint thus exhibits the observed linear relation with joint displacement. However, the PIP joint is not directly connected to the origin of the coordinate system. Its displacement is a function of both the MCP joint displacement and the rotation of the fingertip phalanx about the PIP joint as depicted in Figure 6.15. That is, for every angular rotation of the MCP joint, the PIP joint rotates by the same amount plus an extra relative rotation about the MCP joint.

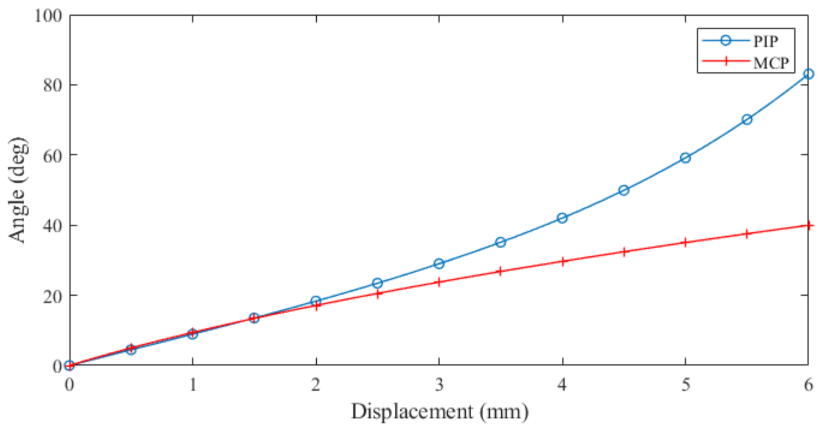
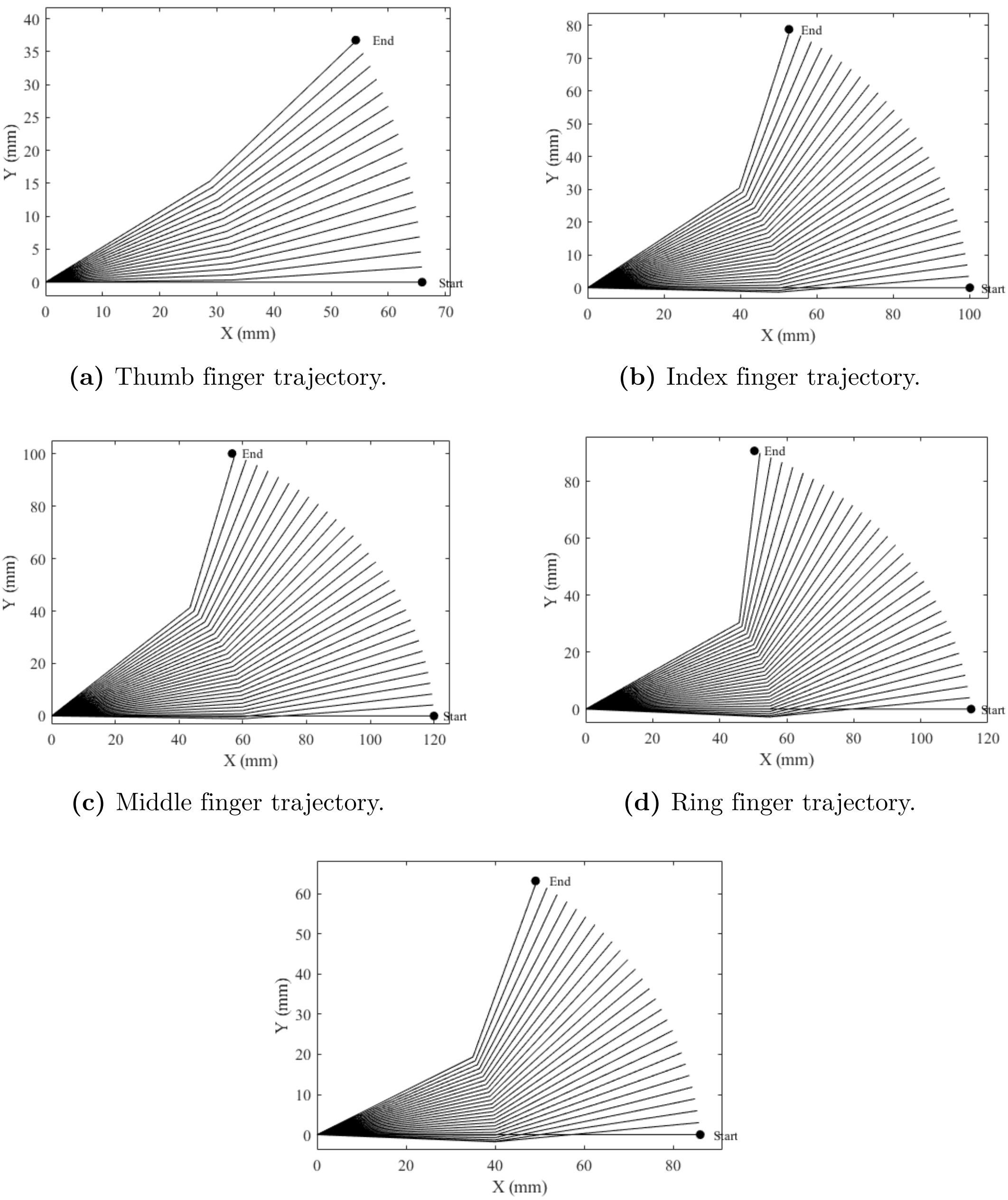


Figure 6.15: Variation of joint angles with displacement.

The fingertip positions of the RPH model when the joints are inflexion for the 3 Jaw chuck grasp are shown in Figure 6.16a - 6.16e. In these graphs, the origin of the MCP coordinate axis is at the MCP joints, with the hand position originally aligned with the horizontal x-axis. During flexion, the joints rotate clockwise towards the palmer side of the hand as they approach the vertical y-axis, which is initially perpendicular to the palm. From these results, it can be observed that the MCP joint does not translate as it is fixed to the centre of rotation. Furthermore, as the RPH model linkage is made of rigid bodies representing the phalanges, the fingertip is linked to the MCP joint through the PIP joint. From the kinematic analysis, its position is thus a function of the angular positions of both the MCP and PIP joints, as shown in Figure 6.16 results. Again, even though the model does not have an addition DIP joint, it still managed to perform 3 jaw chuck grasp shown in Figure 6.10 and was therefore used in the development of the FLC.



(a) Thumb finger trajectory.

(b) Index finger trajectory.

(c) Middle finger trajectory.

(d) Ring finger trajectory.

(e) Pinky finger trajectory.

Figure 6.16: 3 Jaw chuck fingertip trajectories.

Figure 6.17 shows the Simulink model responsible for actuating the RPH. As shown, the model inputs a signal from the SMA actuator, which in turn detects the angle that the MCP, PIP, and IP joints of the RPH will have to move to accomplish the desired trajectory.

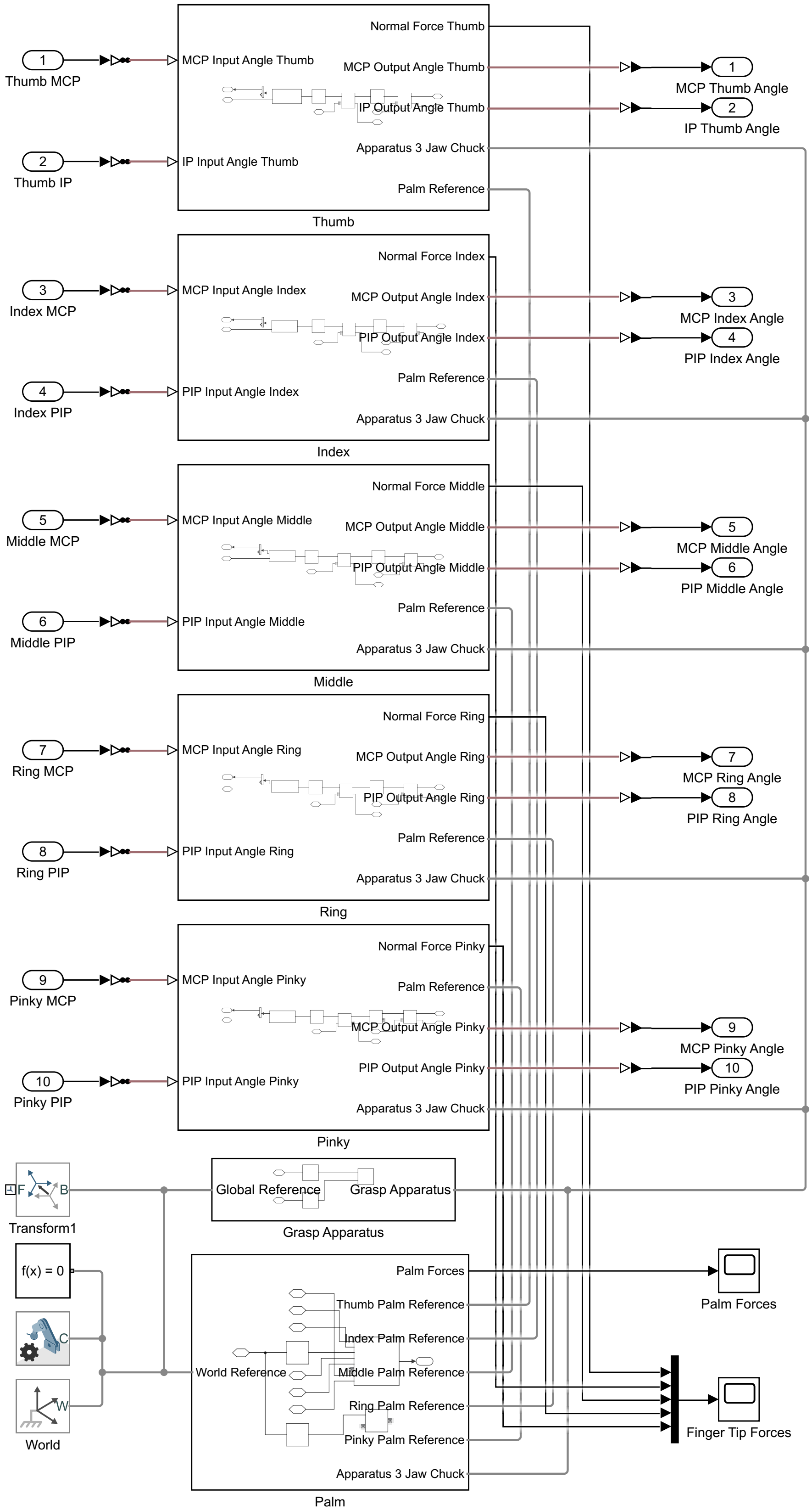


Figure 6.17: Simulink model 1.

The structure of the single finger and its constraints constructed in Simulink is also shown in Figure 6.18. The finger is made of 2 joints, with the MCP attached directly to the actuator. The IP joint is the closest to the fingertip onto which the finger makes contact with the GO. At the fingertip, a ball and socket contact is defined to enable contact force measurement during grasping.

When performing a certain GA, the fingertip comes into contact with the GO at an angle, and the 9-FSRs attached to the fingers and the palm of the hand measure the generated contact force between the fingertip and the GO. In addition, the optical encoders attached to each finger measure the fingertip angle. These 2 variables are continuously measured and form the basis of the input to the FLC of the RPH developed in Chapter 7 using feedback system until the required contact force and fingertip angle is achieved.

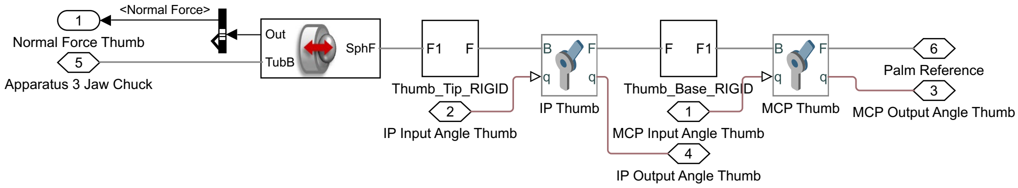


Figure 6.18: Simulink model 2.

6.4 Chapter Summary

In this chapter, a simulation based RPH model was developed and used to accomplish several grasps. The 3-D model of the RPH was developed using Autodesk Inventor, and simulated in Simscape. The six different grasping actions shown in Figure 3.2 were simulated so as to optimise the response of the SMA actuated RPH. Depending on the grasping action to be accomplished, the target positions were calculated and proper boundary conditions specified by the model. These were used to estimate the error from the measured MCP and PIP joint movements during the grasping process. Furthermore, using FSR, the contact force between the finger and the GO was measured and will be used in Chapter 7 to optimise performance of the RPH.

Chapter 7

Fuzzy Logic Control of a Smart Material Alloy-Actuated Robotic Prosthetic Hand

Towards a Stable and Robust Controller

In this chapter a FLC was developed to control the behaviour of the RPH used to mimic day-to-day activities of a human hand during different tasks. The FLC was designed to be able to isolate parameter variation effects by adjusting controller range of universe of discourse, degree of membership function overlap, the rules that map system inputs to outputs, and the fuzzy set modes. The simulation results from Simulink showed that both the steady-state and transient system responses of the system were affected by varying the controller inputs range of universe of discourse. Furthermore, given that a proper rule base is set, the system desired output could be achieved by adjusting fuzzy set output modes. The PID controller managed to track fingertip position with an initial 5.53% overshoot, whereas the FLC exhibited 0 overshoot. For the force control, both the PID and FLC exhibited the same trend, meaning their accuracies were within similar ranges. In addition, when both force and position FLCs were implemented, it was shown that the 2 controllers were mutually exclusive in controlling the variables. A 4.78% increase in the SSE for the FLC in position control was recorded, and the SSE in force control periodically varied between -14.28% and + 16.23%. These results were obtained after varying the output gains of the FLC force and position control to be 67%, and 33% respectively.

7.1 Theory and Literature Review

The increase in the use of multifunctional RPHs has prompted the need for advancement in signal control in order to improve functionality. Currently, research is being geared towards the development of technologies that enable RPH control systems to accurately mimic the range of human motion with minimum user input. The vast majority of intelligent RPHs require implantation of electrodes and sensors, which apart from them being overwhelmingly expensive, may be difficult to implant in individuals with limp deficiencies.

With that in mind, most people eventually resort to the use of less functional RPHs, resulting in reduced ability to perform day-to-day activities. When controlling RPHs, focus is centred on controlling the actuation force and the respective positioning of each finger during grasping. These parameters are usually measured using sensors connected to the complex sensorimotor pathways and are passed to the RPH control devices which determine how much these parameters need to be altered. These sensorimotor pathways not only coordinate muscle movement, but also process feedback and predict consequences, thus forming part of the control system.

In healthcare systems, an actuator-driven RPH enables the patient to achieve more dexterity through the use of advanced control strategies. Therefore, factors that affect human hand movements, such as the anatomy and physiology need to be properly understood when designing a control system for an RPH. These are responsible for providing signals that drive muscle movements. To achieve dexterity, neuromuscular synergies through the central nervous system allow for the control of a large number of joints and muscles using small sets of neural signals. These synergies, together with those derived from soft robotic technologies are used to develop highly functional simplistic control systems for an under-actuated RPH [256]. However, errors due to anatomic changes also affect the central nervous system, hence proprioceptive and tactical sensory channels are made use of to achieve somatosensory feedback. This enables the prediction of future actions through constant internal structure update of the hand and GO interaction.

7.1.1 Generic control systems and theory

The homeostasis designed autonomous and natural control process contained in the human body is responsible for maintaining human life. However, in some instances, the human control system fails to accomplish its natural function, thereby requiring external control systems such as in robotic hands to assist the body in regulating its natural actions [257, 258, 259]. These external controllers are usually based on adaptive, optimal, and predictive control strategies. The aim of these controllers is to stabilise physiological variables to the specified setpoints. This can be accomplished either through negative feedback to reverse perturbations or using anticipatory open-loop systems [260, 261].

When applied to RPHs, the homeostasis principle allows for the control centre coordination with the sensors and actuators so that specified grasping actions can be accomplished. The three most basic control strategies being the negative feedback control, feed-forward control, and adaptive control. In this dissertation, focus is placed around the control of force and the position of the RPH fingertip. As such force sensitive resistors were used to measure the grasping force, with optical encoders as position sensors. Finally, the SMA-based actuator described in Chapter 5 receives signals from the controller to adjust the movement of the RPH during grasping.

The block diagram for a negative feedback control process for a human body is shown in Figure 7.1 [262]. This control strategy involves the use of both force and position sensors attached to the RPH to detect both the applied force and the location of the RPH fingertip in relation to the specified setpoint. In the comparator, a comparison between the actual outputs and the setpoints occurs, after which an error signal is sent to the controller. The controller then calculates the corrective action and sends it to the actuators. The actuators respond to the corrective signal such that the physiological variable is maintained in a stable steady state [263].

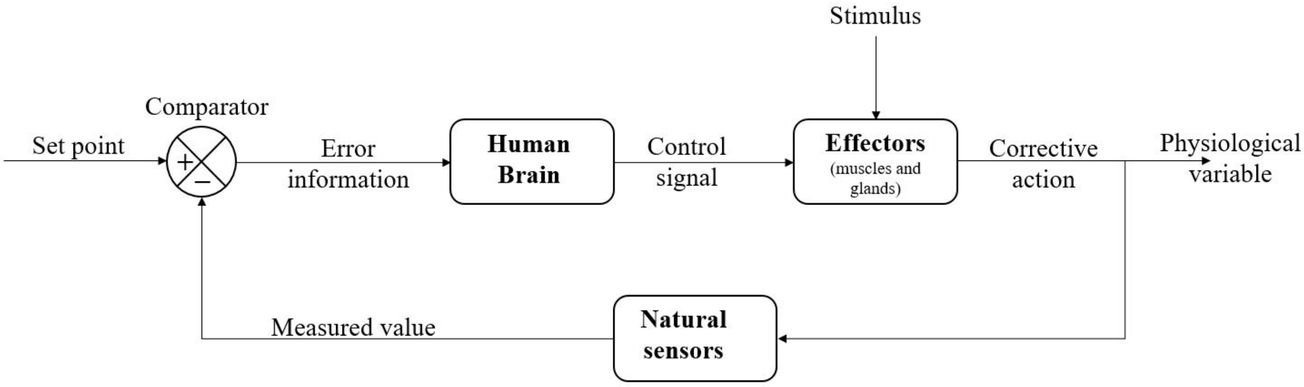


Figure 7.1: Negative feedback block diagram of the human body control process.

In a feed-forward controller, command signals are generated without the need of a continuous closed-loop, that is, feed-forward control strategy is open-loop [264]. Figure 7.2 shows the feed-forward control system with sensors used to both measure and predict extrinsic perturbations. Apart from being used as the direct forcing function, a feed-forward controller output can also be transmitted as a command signal to a feedback controller. These input commands being the setpoints of the overall control system [263].

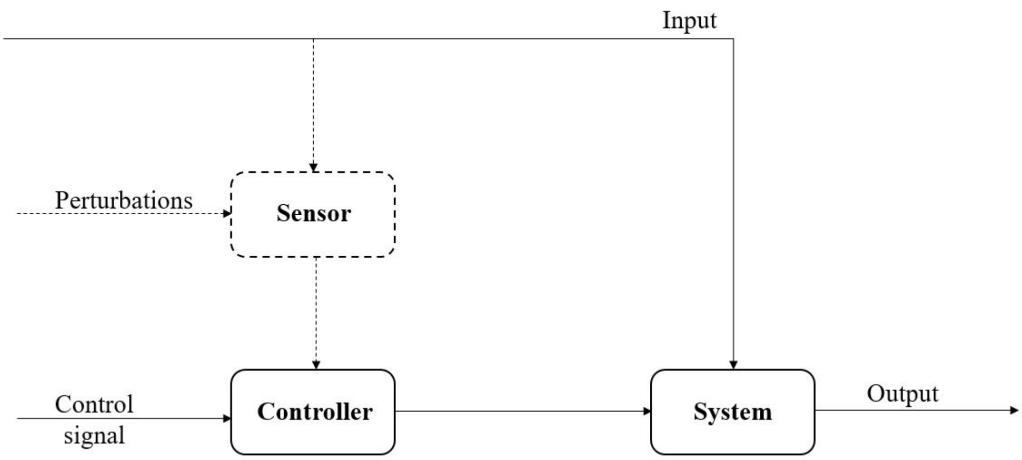


Figure 7.2: Schematic representation of a feed-forward control system.

As opposed to negative and feed-forward control where there are immediate changes in the output, adaptive control modifies system elements. In adaptive control, feedback information is used to enhance beneficial alterations in the controlled variable over time [262, 265]. Figure 7.3 shows the model reference of an adaptive control system. In this model, the actual and setpoint outputs are compared and an error signal is passed to the dynamic evaluator, which modifies the feed-forward controller characteristics. An adaptive controller modifies the systems response to future input signals, and therefore does not produce real-time changes [263]. This is the basis of most robotic hand controllers due to the need to constantly move the hand during grasping.

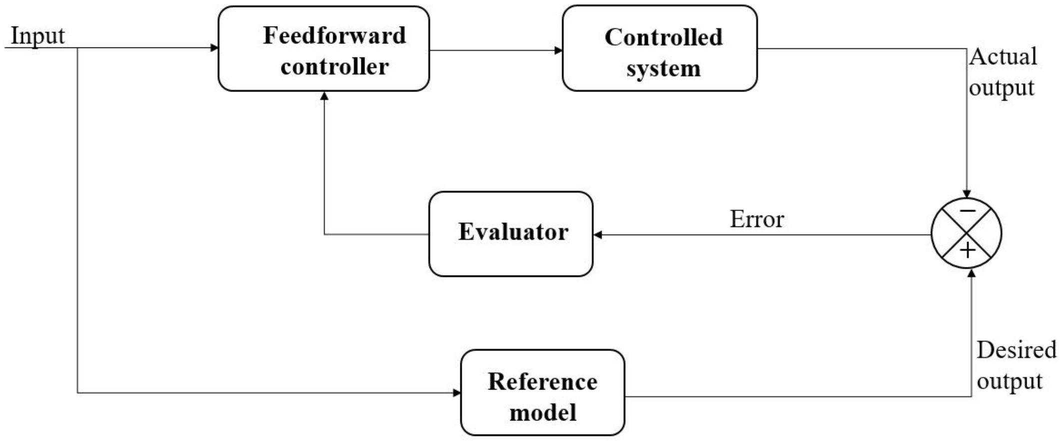


Figure 7.3: Adaptive model reference control system.

7.1.2 RPH control systems

In order to achieve precise dexterity, RPH users would need to control finger and thumb movements to generate the required grasping force. This movement control would require either a mechanical, electromechanical, or an electronic interface. Through muscle synergies, signals generated by the user can be sent to the actuator to accomplish the intended action. Apart from grasping, the user might simply want to open and/or close the hand [266]. The simple feed-forward or open-loop control strategy shown in Figure 7.2 may be implemented in order to accomplish such action. However, for precise control of the robotic hand fingers, there is need for comprehensive mathematical modelling of all the control systems' building blocks [263]. This requires a complete determination of all the model parameters through measurements. However, backlash and friction are variable parameters in finger mechanisms which make modelling complex and difficult to control the output variable to the given setpoints [266].

The negative feedback control system shown in Figure 7.1 is a much better idea than the open-loop system for RPH control. A controller with negative feedback provides better performance by enabling automatic grasp compensation and precise movement of the RPH fingers. Although the open-loop control system works well in force and position control, there is a need to monitor the hand continuously which sometimes has a cognitive burden on the user [266]. To reduce the cognitive burden on the user, a hierarchical control structure shown in Figure 7.4 can be adopted to implement a control strategy. Also if Figure 7.4 is implemented with automatic electronic systems used in the lower control levels, the user can signal the controller to a specific task and observe any deviations of the RPH from the setpoints. Haptic feedback can be used to allow sensory feedback to the RPH users [263, 266].

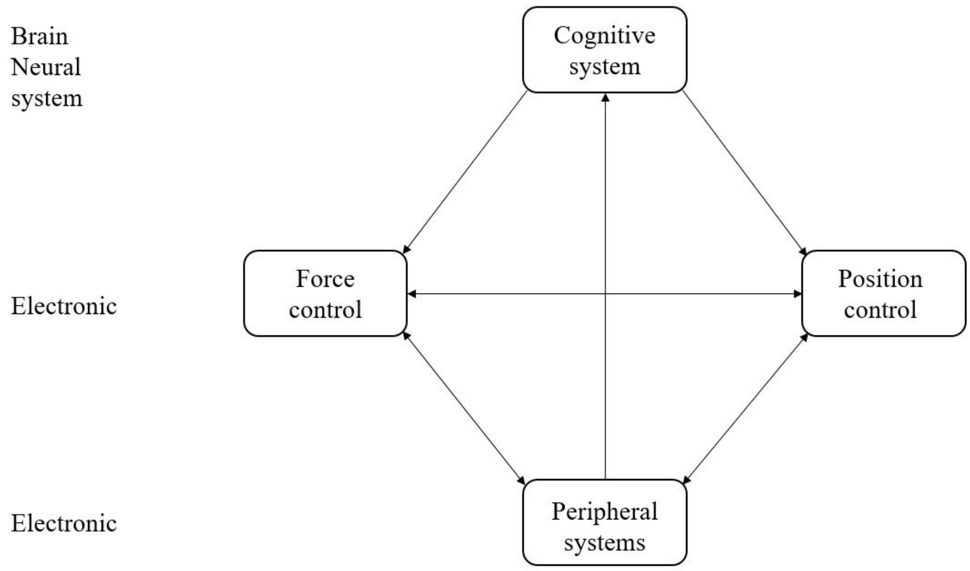


Figure 7.4: Hierarchical control structure of the human control process.

In RPHs, several control algorithms are implemented in the controllers for both feedback and feed-forward loops. Due to the RPHs being nonlinear dynamic systems, no general control algorithm is suitable for all robotic hand functions. The most common control algorithms include feedback linearisation, linear control, and adaptive control [267]. Feedback linearisation aims at achieving a linear differential equation based on the error command using a defined control law. Since the control law design is based on a nominal robot model, this control technique can be thought of as being model based. However, due to disturbances and uncertainties in parameter variations, robustness is not guaranteed when using this control technique [268]. The linear controller, designed from linearisation of the equations of motion about a setpoint is the simplest technique.

This method determines root stability locally, of which the most practical forms being the Proportional, integral, and derivative control methods [269, 270]. Lastly, adaptive control technique is mostly used for the control of time-varying setpoints. This method is more effective than both the feedback linearisation and linear control techniques. In addition, adaptive control can be made more robust by designing it based on the system nominal model and taking into consideration some degree of uncertainty in parameter variations [271]. This method is the basis of FLCs that will be developed in this dissertation since the RPH is required to achieve grasping of objects with different shapes, hence the need for adaptive control.

7.1.3 Force and position control

Application of robotic hand technologies present challenges regarding size, appearance, weight, and control. All these are critical to achieve a suitable and robust robotic hand that enables the user to perform day-to-day activities. Most prosthetic hands have a few degrees of freedom, limited humanoid appearance and make use of a single actuator that can exert large forces during grasping [272]. These robotic hands are thus easy to control. However, an increase in the DoFs requires the use of more actuators, and hence new control strategies thus need to be incorporated. Given that, many researchers are focusing on the improvement of control algorithms for the different contact forces exerted due to the action of different actuators.

A typical example in force control is by taking a control action as being proportional to the sEMG signals from antagonistic user muscles during grasping [273, 274]. This is a neural-based control approach, which can be integrated using a mechanical design approach to provide a feasible RPH force control technique. Zhao *et al.*, developed an EMG motion classifier that used a neural network based on variable learning rate with wave parametric autoregressive (AR) model. This pattern classifier uses signals from sEMG to identify the type of intended hand movements [274].

The core of the RPH designed in this dissertation is based on an under-actuated cable mechanism for each of the five fingers. This robotic hand used a single force sensor on each fingertip and four sensors on the palm of the hand to measure the contact force between the hand and the grasping object. To improve force control on this bio-inspired RPH, force regulation and stability during grasping must be accomplished. Direct approaches to the control of grasp force such as compliance control, hybrid position or force control, and impedance control can be implemented [272]. These controllers can be implemented in both the feedback and feed-forward control schemes presented in Section 7.1.1. Their aim is to provide accurate control on the robotic hand, by rejecting noise, disturbances and minimising the error signals.

The most conventional controller used in RPH systems is the Proportional-Derivative (PD) controller. This controller can be used to obtain the angular positions of the finger joints and contact forces given the required trajectory of the RPH fingertip. When implementing this type of controller, the input forces to the controller must ensure that the angular acceleration of the joint linkages satisfy Equation 7.1.1:

$$\ddot{q}(t) = \ddot{q}_{bar}(t) + K_D(\dot{q}_{bar}(t) - \dot{q}(t)) + K_P(q_{bar}(t) - q(t)) \tag{7.1.1}$$

Here K_D is the derivative gain and K_P is the proportional gain, and the variables q_{bar}, $\dot{q}_{bar}$, and $\ddot{q}_{bar}$ are respectively the desired reference joint trajectory, velocity, and acceleration. The actual measured joint trajectory, velocity and acceleration are given by q, $\dot{q}$, $\ddot{q}$ respectively [275]. The error ($e = q_{bar} - q$) can be made to approach zero by selecting the gain K_D and K_P , such that the roots of the characteristic equation have real parts that are negative.

This control model has been shown to exhibit better convergence when implemented in a decoupled control system where each joint is controlled by an independent single PD controller. However, for a multi-fingered RPH with a large number of DoF, the use of PD controllers makes the system computationally expensive, and also increases the steady-state error [276]. In addition, during fast grasping where there is a drastic change in the inertial loading terms which makes the system nonlinear, the PD controller with constant gains fails to perform well [275].

To reduce the effect of system nonlinearity of the controller performance, the Active Force Control (AFC) can be implemented. This control technique ensures that stability and robustness is maintained in the system even in the presence of both noise and disturbances [277]. The AFC method relies on the measurement and estimation of the number of parameters that affect its ability to compare signals and the reduction in the cost of computation. The error signal from the compensator is used to calculate a corrective signal that is fed to the actuators.

This method can be used for nonlinear systems due to its fast decoupling ability

[275]. In order to obtain more accurate results, an iterative approach proposed by Uchiyama and developed by Arimoto [278] can be implemented together with the AFC. In their study, Arimoto *et al.*, proved that the convergence, robustness, and stability of the learning algorithms improved with the increase in the number of iterations [275]. However, the AFC with iterative learning produces large initial tracking errors at the onset of contact between the fingertip and the grasping object [275].

With advancement in signal processing techniques, recent force and position control methods have been developed based on PID controllers. These controllers greatly improve the control of robotic hands. Studies by Yuden *et al.*, [279], showed that by using a PID controller the position control of an RPH can be accomplished with steady-state errors less than 0.11°for a 30°reference angular position. Their study used a master data glove (MDG) with flex sensor and XBee module for signal transmission to the robotic hand.

In order to receive the signal from the MDG, the robotic hand was also equipped with an XBee module, together with a DC-geared motor. Even though greater system performance was achieved, the system transfer function was only confirmed using system identification tools, instead of mathematical modelling. Reconfirmation using mathematical modelling would provide performance by allowing easier trial and error modifications of the transfer function [279].

The use of PID controllers for positioning of robotic hands can also be extended to higher DoF systems. Shauri *et al.*, [280] developed a method to fine tune the parameters of PID controllers for the feedback position control of a 7-DoF 3-fingered RPH. Their RPH model was actuated by a DC motor, with magnetic encoders as sensors, and a motion control unit.

A trial and error method was used to improve the RPH transient response. It was observed that the robotic hand managed to follow the prescribed trajectory motion that was pre-programmed using the Motion Manager Application with acceptable precision. The PID controller managed to reduce the RPH rise time, overshoot, and the steady-state error [280]. However, this method only proved accurate when used to grasp round and rectangular objects.

7.1.4 Fuzzy logic control

The use of fuzzy logic systems has gained popularity in the design of accurate and robust control systems. Due to their accuracy, FLCs are increasingly being used in the control of robotic manipulators to enable accurate mimicking of the human hand motion and grasping capabilities [281, 282]. Several researchers have shown that FLCs have an added ability to interpret linguistic data other than the commonly used control techniques which normally work with only data presented in a mathematical framework [283].

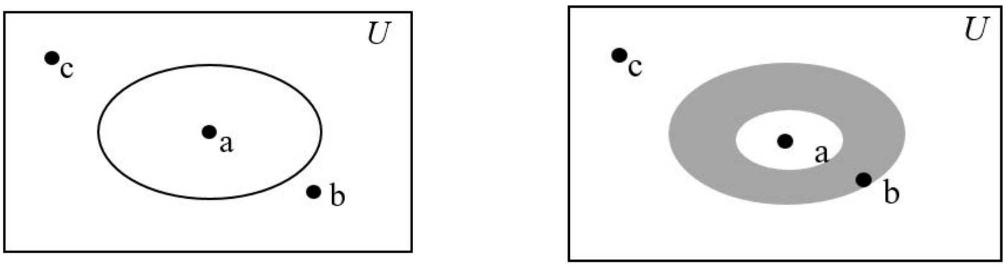
Mathematical models can be sufficiently implemented in the control of systems with time-varying outputs or continuous disturbances. However, as the system complexity increases, mathematical models become less effective for in the design of controllers. This unreliability of mathematical models to precisely control system behaviour continuously increases until a point is reached were precision and relevance almost become mutually exclusive, such that it becomes almost impossible to implement mathematical-based control strategies. Most day-to-day activities such as grasping are very complex and therefore inherently fuzzy [284].

Unlike mathematical models, FLC systems have shown to include skill, reasoning, and heuristics in the determination of future model events. Therefore, FLCs are much more effective when implemented in systems with ill-defined models, which are not reliable enough [281]. These FLCs use multivalued logic sets to describe vagueness or probability of occurrence of a certain decision or variable.

This theory was first introduced by Zadeh [284], who created fuzzy sets of different degrees based on the multivalued logic theory established earlier by Lukasiewicz [282]. Unlike the classical Boolean logic (1 or 0; True or False), fuzzy logic sets are multivalence from 0 to 1. Based on the described multivalence fuzzy logic by Zadeh [284], Mamndani developed the first fuzzy logic system to control a steam engine and traffic lights, which prompted further development in FLC due to its improved precision monitoring abilities and greater accuracy in tracking system variables [281, 282, 283].

7.1.4.1 Fuzzy sets and fuzzy logic

The use of fuzzy-sets is based upon a definition of collective objects that have the same characteristics known as the universe of discourse [281]. A collection of these objects/elements is called a classical-set, in which all member elements exclusively belong to this set. On the other hand, elements that are not in the member set are not related in any form to the set. Figure 7.5 shows a definition boundary of both classical and fuzzy-sets [263].



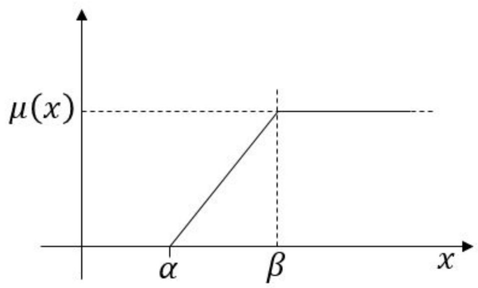
(a) Classical-crisp set boundary. **(b)** Fuzzy-set boundary

Figure 7.5: Definition boundary for classical and fuzzy sets.

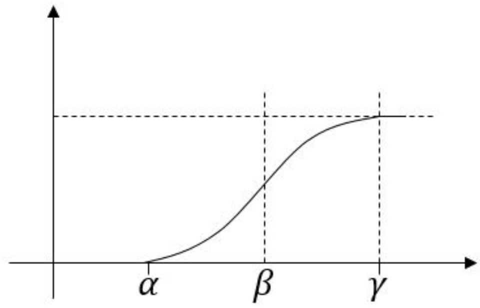
The fuzzy-set theory extended the characteristic function used in mathematical modelling into a much more generalised or vague form called the membership function that ranges continuously from 0 to 1. The fuzzy-set is thus defined by such an extended membership function. In contact, the two-element characteristic function is used to describe the crisp-set. In fuzzy-sets, the theory of vagueness is used to determine the degree of membership of a variable to a set. This degree of membership becomes important to obtain the overall effect of a specific variable to the output [263, 285]. Referring back to Figure 7.5b, point a is encompassed in the fuzzy-set, while point c is not. Since point b falls on the boundary, its membership to the fuzzy-set described is ambiguous.

7.1.4.2 Membership functions

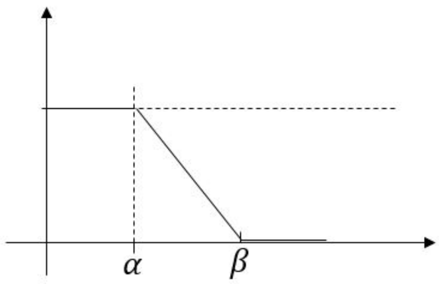
The fuzziness of FLC is best described by its membership function which gives an indication of the degree-of-truth of a certain variable. The use of membership functions is based on experience as opposed to knowledge in solving for the effect of a specific variable on the system output [286]. The most commonly used types of membership functions are shown in Figure 7.6, and the choice of which type of function to use is application dependent.



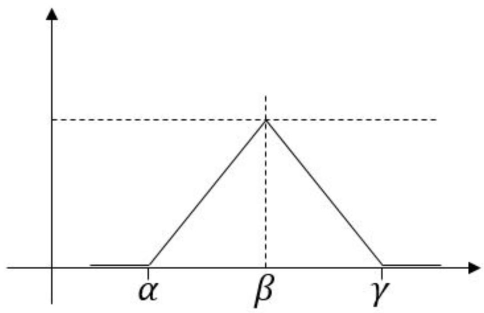
(a) Γ-function



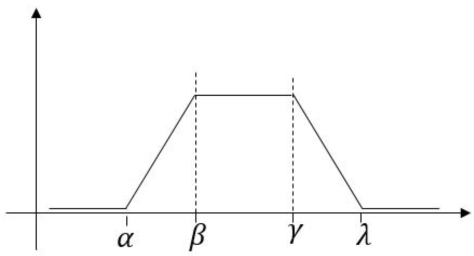
(b) *S*-function



(c) *L*-function



(d) Λ-function



(e) *Pi*-function

Figure 7.6: Common types of membership functions.

In addition, in most fuzzy control systems, a simple triangular or trapezoidal function is sufficient in achieving accurate and precise system control. This limits the use of the more complex Gaussian and S-type membership functions. The Γ-function, L-function, and Λ- function memberships can be defined as follows:

$\Gamma - function,\ \Gamma : U \to [0, 1]$

$$\Gamma(x; \alpha, \beta) = \begin{cases} 0, & x < \alpha \\ (x-\alpha)/(\beta-\alpha), & \alpha \leq x \leq \beta \\ 1, & x > \beta \end{cases} \tag{7.1.2}$$

$L - function,\ L : U \to [0, 1]$

$$L(x; \alpha, \beta) = \begin{cases} 1, & x < \alpha \\ (x-\beta)/(\alpha-\beta), & \alpha \leq x \leq \beta \\ 0, & x > \beta \end{cases} \tag{7.1.3}$$

$\Lambda - function,\ \Lambda : U \to [0, 1]$

$$\Lambda(x; \alpha, \beta, \gamma) = \begin{cases} 0, & x < \alpha \\ (x-\alpha)/(\beta-\alpha), & \alpha \leq x \leq \beta \\ (x-\gamma)/(\beta-\gamma), & \beta \leq x \leq \gamma \\ 0, & x > \gamma \end{cases} \tag{7.1.4}$$

α , β and γ are the specific points of interest that define the properties of the different membership functions.

7.1.4.3 Fuzzy control systems

Standard controllers require a mathematical model that describes the system behaviour using differential equations. However, FLCs make use of fuzzy sets described in section 7.1.4.1, together with membership function rules to describe the system behaviour [281]. Figure 7.7 shows a typical block diagram that represents an FLC. This block diagram is made up of 5 main elements namely, the fuzzifier, knowledge base, rule base, inference engine, and the defuzzifier [263, 285].

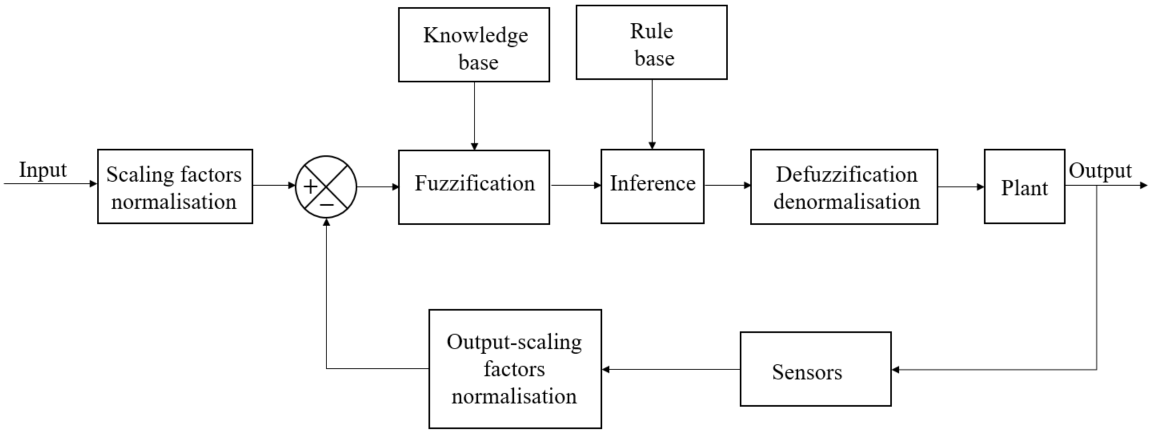


Figure 7.7: Fuzzy logic control block diagram.

All the parameters of the FLC block diagram elements can be set such that they automatically change during the control process, thereby creating an adaptive fuzzy logic controller. FLCs that have fixed element parameters are non-adaptive, and are commonly used as they are easy to develop and implement [281, 287]. In the fuzzification module, input crisp data is converted into fuzzy values. This fuzzification process allows data to be suitable for representation in the rule base [288].

The complete plant database is stored in the knowledge base in which all fuzzification commands are processed. These include the membership functions, representation of fuzzy sets input-output variables, as well as functions for mapping between the crisp and fuzzy domain [281]. In the rule base, expert knowledge is used to define the system control strategy. This is achieved using fuzzy inferences and consequences together with linguistic variables. This can be represented as shown below:

IF $error(e)$ is $BigPositive(BP)$ THEN $output(u)$ is $BigNegative(BN)$

rule antecedent   rule consequent

Here the error and output are the linguistic variables, and Big Positive, Big Negative being the linguistic values. In FLC, the Mamdani fuzzy implication technique is usually used to interpret these rules [281, 286]. The membership functions for a typical fuzzy logic controller are shown in Figure 7.8.

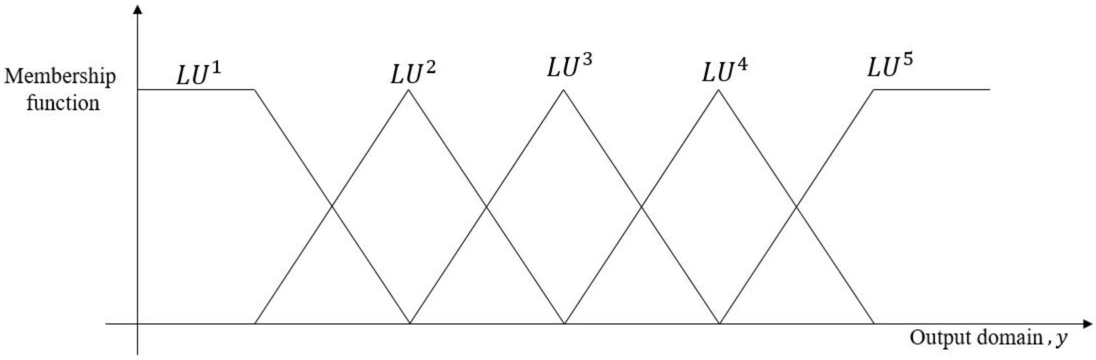


Figure 7.8: Membership functions of the output linguistic values.

These membership functions are used to obtain the output variable of the controller over the defined universe of discourse. That is, the linguistic values $LU1$ to $LU5$ are a mapping of the input data. At this point it is worth noting that the ranges of the membership functions can overlap between each other depending on the expert conditions defined. For the case with clipped membership functions, the output value will thus be the union of the two functions as shown in Figure 7.9.

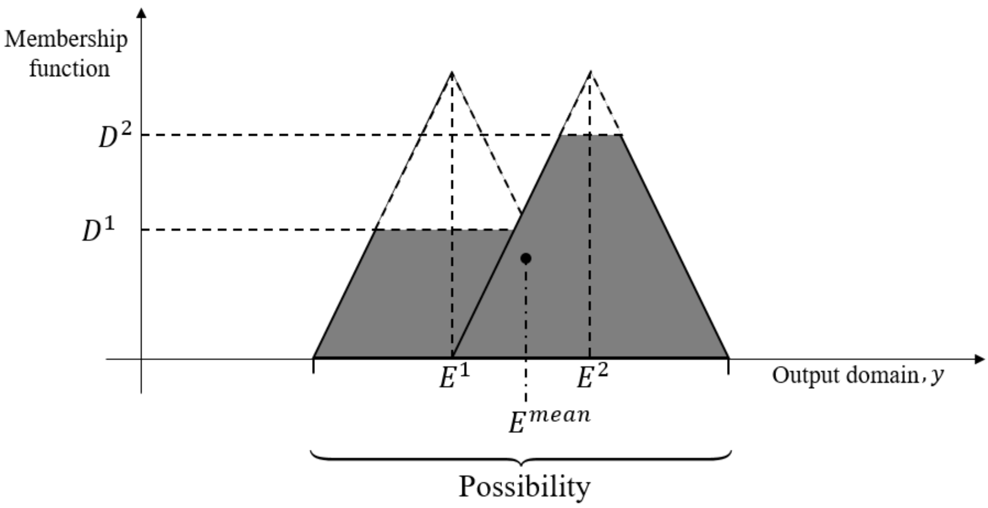


Figure 7.9: Distribution of an output condition using the weighted area method.

The output value of the inference engine on its own has no practical significance as it is a fuzzy output. For it to be used as a control command, this result needs to be defuzzified into a crisp value that can be passed to the actuator or plant. Several defuzzification methods can be used depending on the FLC application and processing power available. With that in mind, the weighted average defuzzification technique shown in Figure 7.9 is commonly applied and yields acceptable crisp values with minimum processing power requirements [281, 286, 289].

7.1.4.4 FLC control of RPH

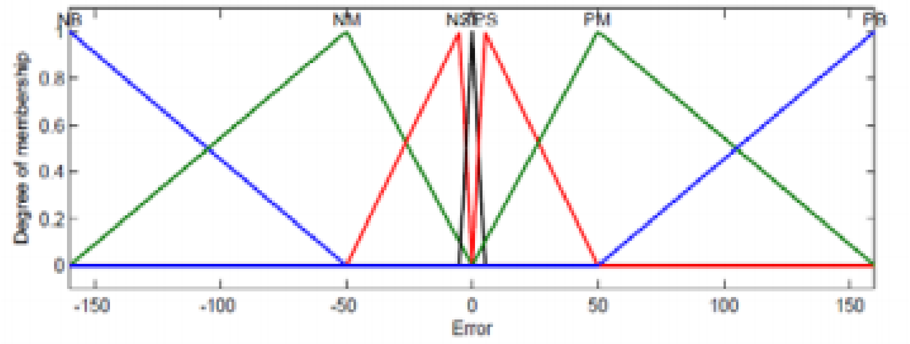
The use of fuzzy logic greatly reduces the complexity in the control and classification of multi-signal patterns, which usually occur in robotic hands that have multiple DoF. Although other algorithms have been developed to accomplish the same goal, they usually have a longer data acquisition window and decision making times, which in most recent cases might not be acceptable to the end user [287, 290, 291]. Fuzzy systems on the other hand, provide fast response times and are easy to implement in the control of RPHs. Furthermore, once the membership functions have been established, very little computation occurs in the fuzzy control system. The implementation of FLC can therefore be accomplished readily using simple 8 bit microprocessors [287].

In addition, the ease of membership function inference rule generation using simple signal statistics means that data clustering can easily produce a prosthetic user-transparent fuzzy system. In this case, the user only has to accomplish the easy commands, while the fuzzy system takes care of the more complex computations. The increased use of sensors in RHP systems also reduces cognitive burden on the user.

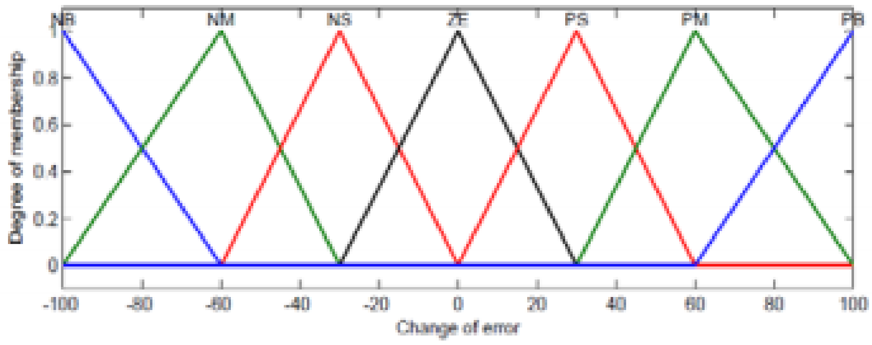
Ajiboye and Weir [287], developed a heuristic FLC for multi-functional robotic control with multiple EMG pattern recognition. In their study, they used basic statistics to construct membership functions together with data clustering for automatic construction of the inference rule base. 4 membership functions (OFF, LOW, MED, HIGH), to cover the input range were generated using the EMG RMS value during contraction. This ensured a 50% overlap between adjacent membership functions. Also, 4 triangular membership functions were also generated for the output condition. The trapezoidal membership functions were specified for both the OFF, and HIGH states in the input range. The rule base was generated using an automatic fuzzy clustering method in Matlab.

Their system showed good update rate with minimum processing time and high classification rates. This model allowed control of multi-DoF robotic hands during both steady and transition motion states [287]. However, their control method was not able to accomplish parallel control, and thus still requires modification to incorporate systems with mutually exclusive control variables.

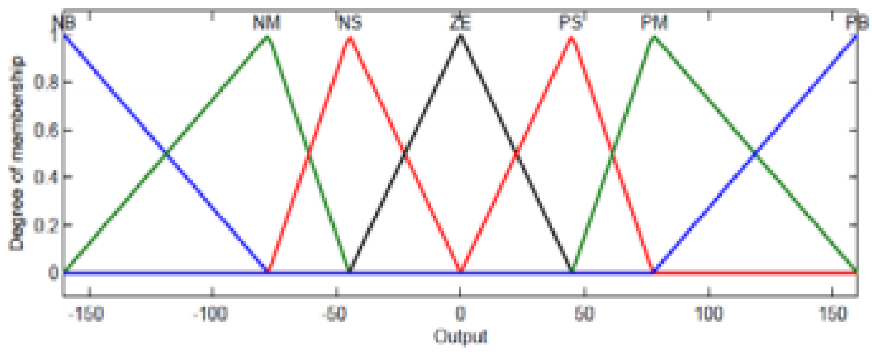
Alassar *et al.*, [292] developed both FLC and PID controllers to manipulate joint movement of a 5-DoF robotic arm using Simulink. In their model, they compared the time responses of a Lynx6 robotic arm due to a step input signal. Their PID controller was tuned using both the Ziegler-Nichols, and root locus methods to obtain the respective PID controller parameters. In addition, a MISO FLC with 2 inputs namely, error and change in error was developed. Figure 7.10 shows the membership functions of the input error, change in error, and the output variable u. Each variable has 7 membership functions that span across their respective ranges.



(a) Membership function of input error e.



(b) Membership function of input change of error Δe.

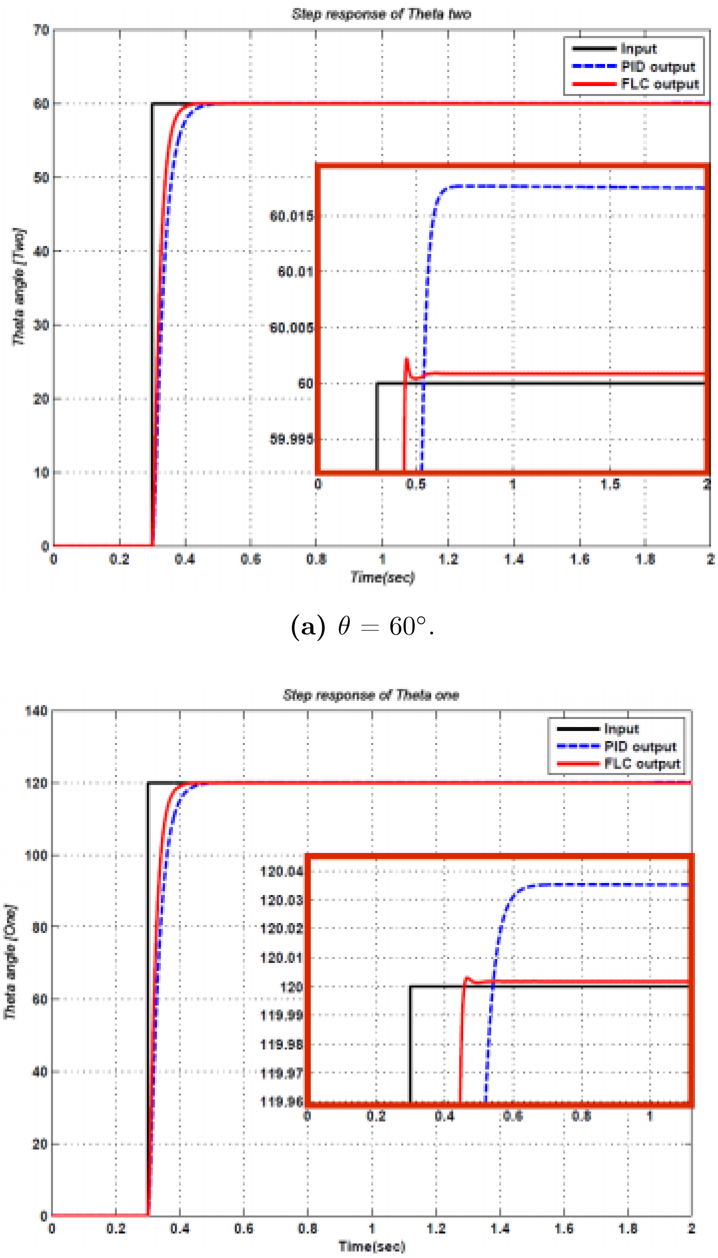


(c) Membership function of output u.

Figure 7.10: Fuzzy logic control membership functions.

Figure 7.11 shows the response of the PID and FLC for a 60°and 120°step input. These results show that the FLC controller had better performance than the PID controller. A rise time of 0.14s and 0.15s was recorded for both angles of the FLC, whereas the PID controller produced respectively, 0.22s and 0.23s rise time for the 60°and 120°angles. Also, both input signals, their FLC had a nearly 0 SSE, whereas the PID recorded SSEs of 0.01 and 0.003 for the 60°and 90°input signals respectively [292].

Most fuzzy control systems do not require intensive memory, this then allows for the generation of high-level fuzzy logic commands. Bowers and Lumia [293] developed an expert fuzzy logic mechanical hand to accomplish grasping of objects that are randomly placed on a surface using an intelligent vision system. In their study, vision data was



(a) $\theta = 60°$.

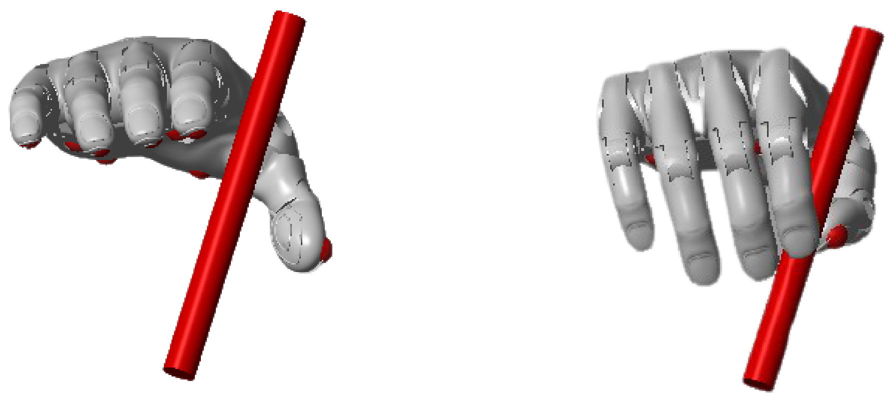
(b) $\theta = 120°$.

Figure 7.11: Fuzzy logic control and PID controller step responses.

used to generate a nonlinear mapping between the grasping object and the configuration of the hand. Using computational geometry in a post-grasp stage, they managed to ascertain grasp quality, thereby validating the fuzzy logic-generated hand configuration [293]. Although this work managed to provide a reliable technique for grasping using limited power, the need to include a learning process that modifies the membership function rules makes low processing power computers unreliable.

7.2 Methodology

This section outlines the methodology undertook in modelling the control and behaviour of the RPHs. Simulink was used to model the behaviour of the RPH during grasping with the aid of different controllers. Both PID and fuzzy logic controllers were used to control the power and force input signals passed to the actuator in order to accomplish the different grasping actions as outlined in Figure 3.2. A Simscape model of the robotic hand in its initial state before grasping commences is shown in Figure 7.12.



(a) Before grasping. (b) In contact with GO.

Figure 7.12: 3 jaw chuck grasp, before and after grasping.

All the 7 initial pre-grasping positions are the same for all the grasping models, thus the rest position of all grasping actions are identical. The robotic hand is a 5-fingered model with 2 joints at each finger making it a 10-DoF system. Each of these fingers is connected to the SMA-based actuator, with each finger being able to accomplish a maximum of 90°flexion during grasping. Also, a ball and socket joint at the tip of each fingertip and at the 4 locations of the palm make the attachment to the force sensitive resistors. These resistors, as explained previously in Chapter 4, measure the contact force between the hand and the grasping objects, which is fed back for control purposes. Figure 7.13 shows the block diagram of the RPH control process developed in this dissertation.

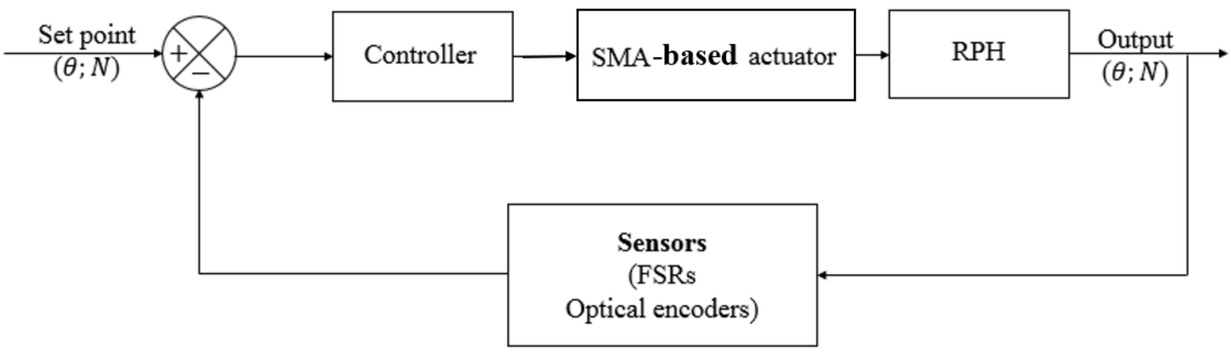


Figure 7.13: Schematic diagram of the RPH control process.

In order to perform a grasping action in the Simulink model, the SMA-based actuator sub-system which is connected to the RPH, extends and contracts as shown from the FEA model in Section 5.3.1. This in turn causes the robotic hand to accomplish the prescribed grasping action intended by the user. The model is also incorporated with optical encoder subsystems attached to each finger for joint angle measurements, which are also feedback signals for control purposes in order to accomplish fine grasping actions.

7.2.1 Feedback system

Before a signal is passed to the actuator for grasping, the position, orientation, and the shape or size of the grasping object needs to be established. This is accomplished by the optical encoder system described above together with the signal from the grasp classification model. Part of the grasping actions involves determining the geometry of the object, this is when cognitive burden arises, as the user would need to have an accurate visual interpretation of the grasping object before classification of which grasping action must be undertaken. The grasp classification technique developed in Chapter 3, Section 3.3 acts as the input to the full control system.

7.2.2 Grasping

The grasping action procedure uses grasp classification data to generate a nonlinear mapping which then pass a signal to the actuator to cause finger movement into the set grasping position. In order to accomplish fine grasping, finger adjustment through control of input power and force to the SMA model is accomplished by the controller. This adjustment is made possible by measuring the angular position of the fingertip and the contact force between the finger and the grasping objecting. Fuzzy logic rule base is used to provide the nonlinear mapping between the grasping object and the current fingertip position. In this model, several fuzzy rule bases and mappings were developed and evaluated for better system performance. The FLC designed was used to establish the quality of the grasping action through continuous adjustments of both power and force inputs to the SMA-based actuator that drives the plant.

7.2.3 Fuzzy mapping rule bases

There exist several methods for vision data transformation to the hand signals. The easiest way being the basic Single-Input-Single-Output (SISO) rule base. This method maps 1 input to 1 output signal. However, the model developed in this dissertation uses both power and force as input signals to the SMA-based actuator. In addition, 2 signals in contact force and fingertip angular position act as negative feedback to the system. Based on the setpoint angle and force, 2 error signals are passed from the comparator. These signals together with their derivatives are used as the input signals to the FLC.

However, in order to minimise complexity, 2 separate FLCs for both the angle and force control are developed. Thus, each FLC would have 2 inputs (i.e., error and change in error) and power as the output variable in both cases. The developed FLC is a Multi-Input-Single-Output (MISO) FLC. Figure 7.14 shows the fuzzy fis system used to develop the membership functions and rules for the control of the fingertip angle during grasping. A similar approach is followed for force control and the fis and membership functions are shown in Appendix 10.

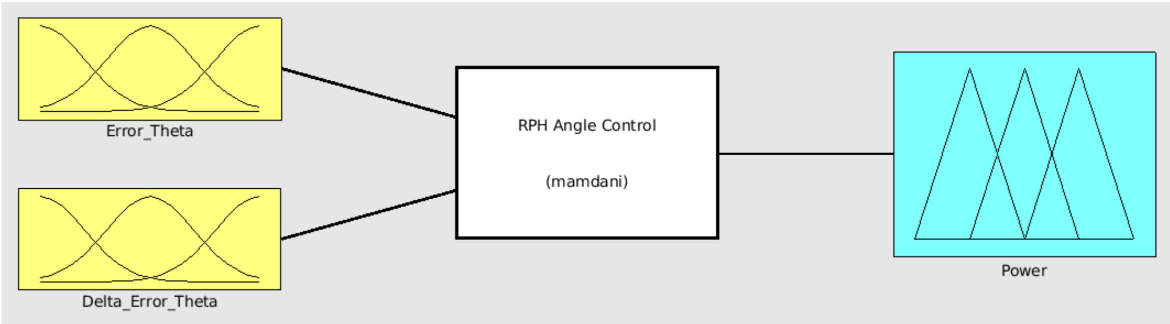


Figure 7.14: Fuzzy fis system for fingertip angle control.

7.3 Results and Discussion

This section presents and discusses the simulation results from the control of the behaviour of the anthropomorphic RPH model using Simulink. The findings are based on the methodology described in Section 7.1.1. The results obtained by simulating the prehensile movement of the 10-DoF RPH model associated with various everyday tasks using the developed FLC are also compared with the PID controller results. The RPH Simscape model in Figure 7.12 was used to simulate the hand movements shown in Figure 3.2 with the different grasps named accordingly in the Simulink model.

The Mamdani-type fuzzy logic method was selected for the development of the FLC. Being more intuitive than the Sugeno-type, the Mamdani method assumes a fuzzy-set output variable. These fuzzy-sets are easy to convert into crisp output variables using the defuzzification process. On the contrary, the Sugeno- method has singleton output variables in the form of consequent equations. This makes it much more mathematically inclined resulting in added complexity without any improved efficiency compared to the selected Mamdani-type FLC. In addition, the Mamdani-type FLC better preserves the local and global interpolations.

The most crucial part in the design of a fuzzy logic controller is the selection and definition of which membership functions to use. In this study, for all the input and output variables, the triangular and trapezoidal membership functions were selected. These membership functions and rules were continually adjusted to improve the performance of the FLC. Several approaches in adjustments of these functions exist such as fuzzy evolutionary algorithms, and fuzzy clustering methods. This study used the generic-based approach which still yields acceptable accuracy at low computational cost. Also, during the different grasping actions, there is notable dynamic variations in system parameters within small amounts of time. These variations can be adequately captured using the selected membership functions. Figure 7.15 shows the chosen membership functions for calculating the error in the fingertip angle.

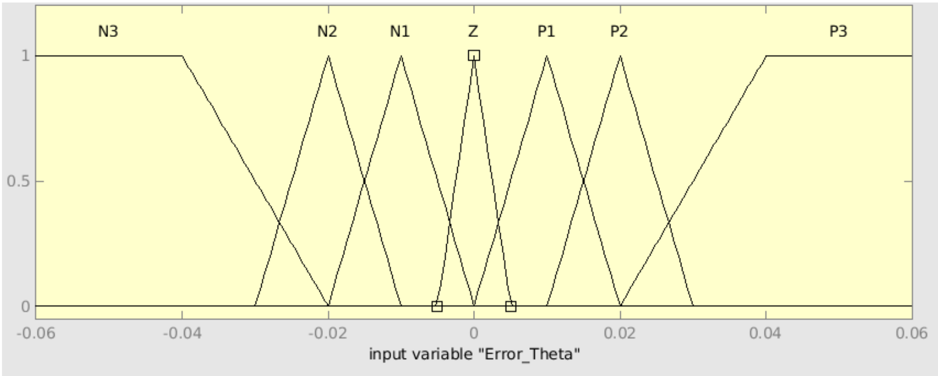


Figure 7.15: Membership function of input angle error.

As can be seen, a total of 7 membership functions were chosen. These membership functions include 2 outer trapezoidal membership functions at the ends of the error range, and 5 differently-sized triangular membership functions spanning almost 40% of the error range. The choice and number of membership functions used in the design of the FLC were based on expert knowledge through iterative adjustments and continuously monitoring the system response. In addition, an overlap of 0.5 was allowed between adjacent membership functions. This overlap was also iteratively adjusted beginning at 0.25, and also provides corresponding linear interpolations between the rules and consequences. Also, the use of triangular membership functions improves the convergence of the controller output as compared to other membership functions.

In a study by Bose [294], on the application of an expert system fuzzy logic approach in motion control of power systems, they used overlapping triangular membership functions crowded at the origin. This ensured better precision control of the system variables near steady state without the need to have a large number of membership functions. This also reduces complexity in that only a few rules will need to be defined [294]. The membership functions of the change in error are shown in Figure 7.16. The change in error assists in developing the possible controller output using a lookup table.

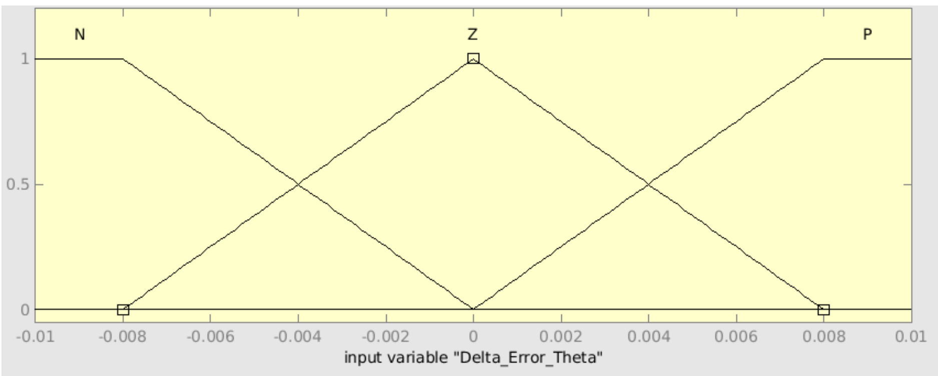


Figure 7.16: Membership function of change in input angle error.

Table 7.1 shows the look-up table used to generate the rule base for the control of the joint angle during grasping. This table uses the 7 membership functions from the error signal and the 3 membership function for the derivative of the error to obtain the different outcomes of the controller. The Mamdani fis system uses the "OR" and "AND" logic rule base to generate control outputs. In this dissertation, the "AND" rule base was used so as to achieve precise control and tuning of the output variable. By continuously changing the number and range of membership functions, the possible control outcomes in the rule base were also being changed. This was done up until the best rule base combination was obtained that provides the most accurate force and position control during grasping.

Table 7.1: Fuzzy logic output lookup

e \ Δe	**N**	**Z**	**P**
N3	W2	W2	W2
N2	W2	W2	W2
N1	W2	W2	W1
Z	W1	W4	W5
P1	W4	W4	W5
P2	W4	W4	W5
P3	W4	W4	W5

The possible results from the lookup table are obtained from the output membership functions shown in Figure 7.17. These membership functions have a 50% overlap, which was reached after careful iterations and observing the response of the controller output during grasping.

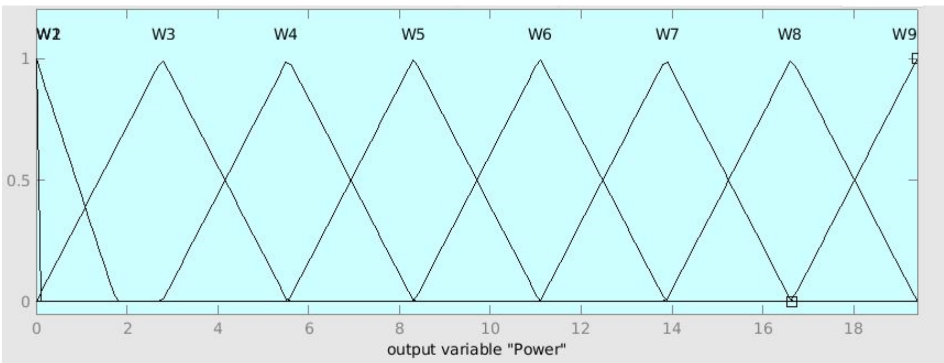


Figure 7.17: Membership functions of output variable.

Although there were continuous changes in the fingertip angle setpoint and the grasping force for the negative feedback control system, the FLC developed model was able to adapt to these variations. This is further made possible as the RPH user plays a part in the feedback control system. This adaptive rule base control system is also able to adjust to disturbances making it more reliable than the conventional PID method that is sensitive to parameter variations.

The use of adaptive FLC in improving control strategies was also investigated by Mahfouf *et al.*, [295]. This study focused on the utilisation of FLC in medicine by first identifying that the human mind operates by extracting meaningful information from approximate data, thereafter producing crisp outputs. With this being the basis of FLC development, Mahfouf *et al.*, utilised an adaptive fuzzy control rule base to control the trajectory of arm movements. Figure 7.18 shows the graphical representation of the defined rules.

These rules can be said to contain information of the system domain and the intended control targets. All the defined rules consist of both a data base and fuzzy control rule base. A combination of any of the rules enable the graphical mapping of inputs to outputs as shown in the Figure 7.18. This graph also gives information to obtain the linguistic control rules and fuzzy logic data, used to manipulate the outcome of the controller. Furthermore, the rules give a characteristic of the control goals and policy of the system.

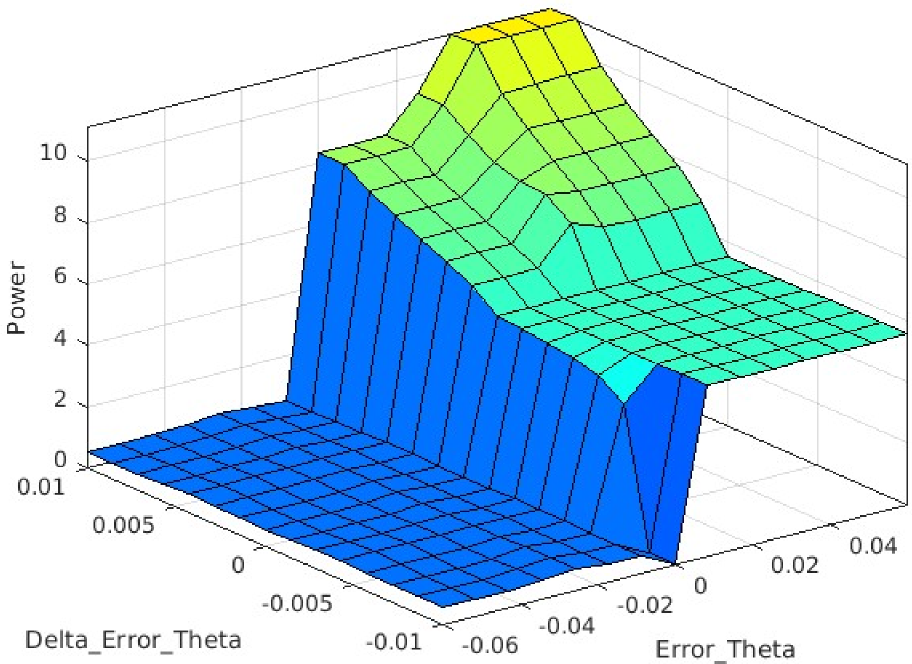


Figure 7.18: Surface plot of the fuzzy logic control defined rules.

Figure 7.19 shows the RPH angle control response when a PID controller is used. The graph shows the variation of joint setpoint angle, trajectory path of RPH fingertip, and the error between the 2 signals. These results are for the 3 jaw chuck during the first 40s for the thump finger. As shown, when the time starts, the RPH is at rest for the first 10s. Thereafter a signal is passed to the RPH to perform the 7 grasping actions sequentially starting with the 3 jaw chuck. At 10s, the fingertip setpoint angle raises steadily from 0 to 0.38 rad in about 2.3s. The fingertip is then set to stay in this position for an extra 27.7s after which the setpoint returns to 0 on completion of this grasping action.

By changing the fingertip setpoint angle, it can be seen from Figure 7.19 that the actual measured fingertip angle follows a similar path to that of the setpoint for the duration of the grasps. However, there is a slight delay in the PID controller response at the onset of grasping which is at 10s. This delay affects the control signal's rise time as seen from the graph, resulting in a 0.02% delay in the systems rise time when the

PID controller is employed. Also, the internal dynamics of the PID controller cause the actual fingertip angle to exhibit the observed overshoot resulting in an initial oscillation and system instability.

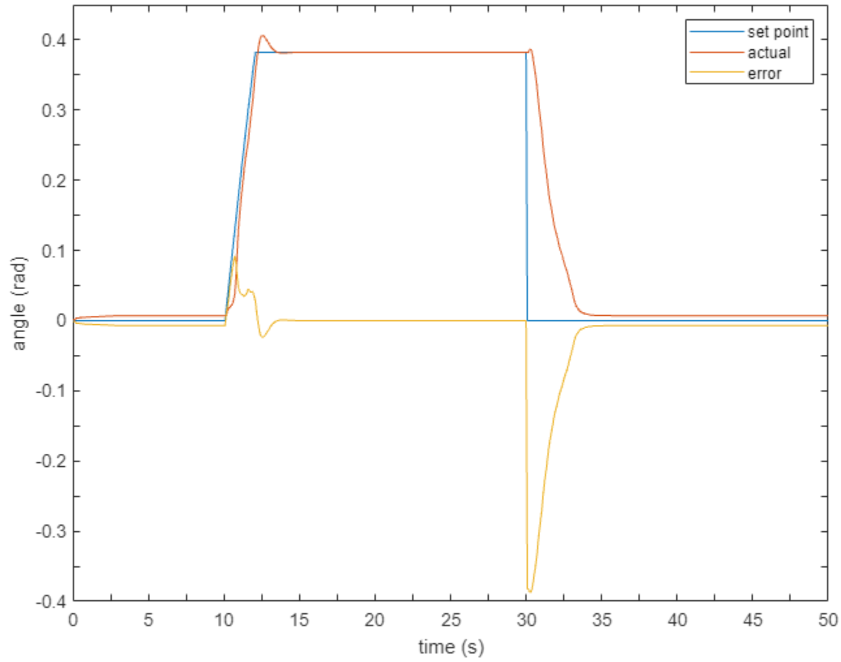


Figure 7.19: PID control system response for angle control.

The derivative gain was increased steadily to minimise overshoot until a point was reached where further increase in the derivative gain caused unexpected system response. From 14.5s, it can be seen that the PID controller follows the same trajectory as the setpoint angle. In this region, the PID controller accurately adjusts to the joint setpoint, however, a slightly high controller gain was responsible for a 0.1374% error being recorded.

Reducing the proportional gain further would cause the output to have a higher error magnitude than the one measured. Also, this error could be attributed to some degree of parameter variations, and thus making it difficult to accurately track the fingertip trajectory. At 30s when the setpoint joint angle for the 3-jaw chuck grasp is switched back to 0, there exists a further delay in the response of the PID controller. This can be attributed to the lack of the PID controller to quickly adjust to variations in system parameters.

Figure 7.20 shows the response of the RPH when the PID controller is used for force control. As can be seen, at the onset of grasping, the force setpoint becomes unsteady and oscillatory with a 0.9837N difference between the maximum and minimum peaks. This oscillatory behaviour is due to the continuously changes in the contact surface areas between the fingertip and the GO. These rapidly changing contact pressure forces cause the controller to continuously try to adjust its outputs resulting in the observed trend. In this region the PID controller overestimates the system response for the first 10s of grasping by about +0.238N to +0.485N. During the last 10s of grasping, this SSE falls to within a range of ±0.0837N. This can be attributed to the PID having a better control ability with each feedback loop.

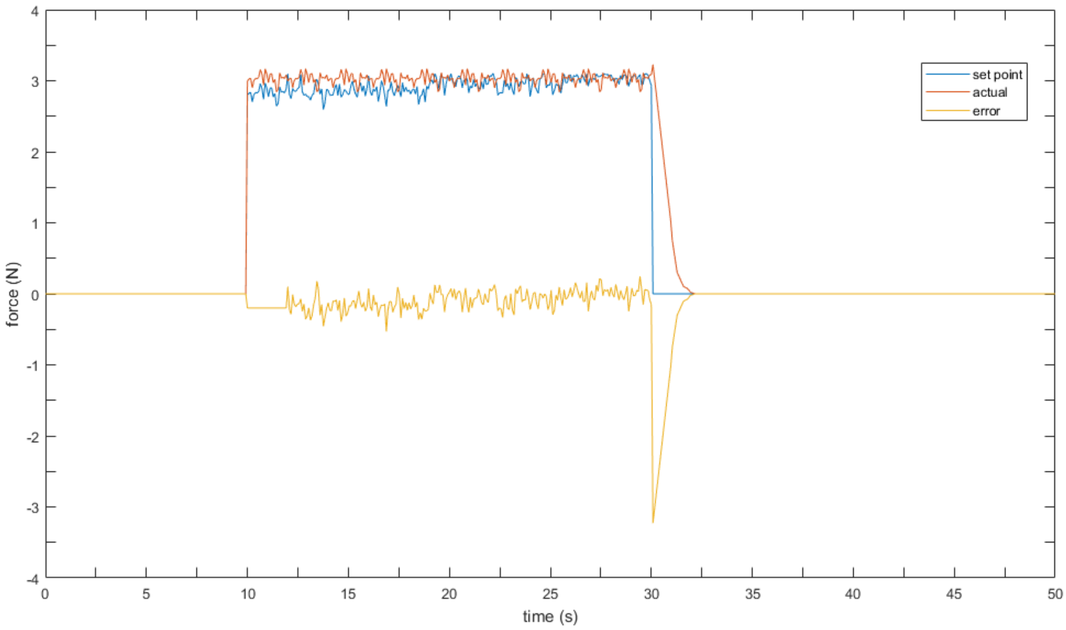


Figure 7.20: PID response for force control.

The criteria used to compare the performance of the FLC is similar for all the grasping actions performed by the different participants. It should therefore be noted that the simulation conducted in Simulink aimed at providing a critical examination of the dexterity of the developed RPH when used to perform human day-to-day activities. Figures 7.21 and 7.22 show the system response from the implementation of an FL controller, for both the angle and force respectively.

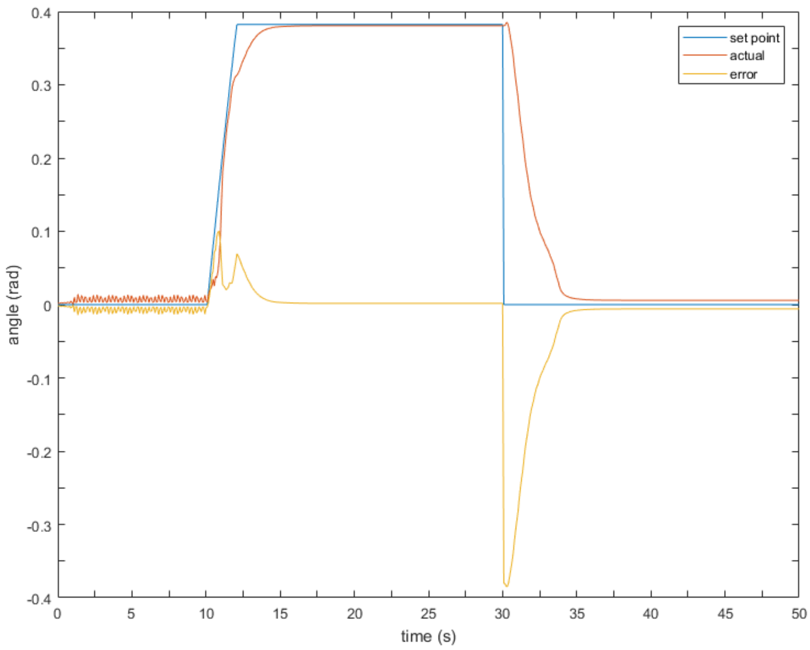


Figure 7.21: Fuzzy logic controller response for angle control.

As shown in Figure 7.21, the FLC accurately follows the set trajectory of the RPH during the 3-jaw chuck grasping action. Based on the methodology described in Section 7.2, the FLC used expert knowledge and iterative tuning. From this, initial results of the FLC with little adjustments in membership functions and rule base gave large steady-state errors. The range of universe of discourse was expanded iteratively for both angle and force input error signals and was equivalent to the reduction of a PID controller gain.

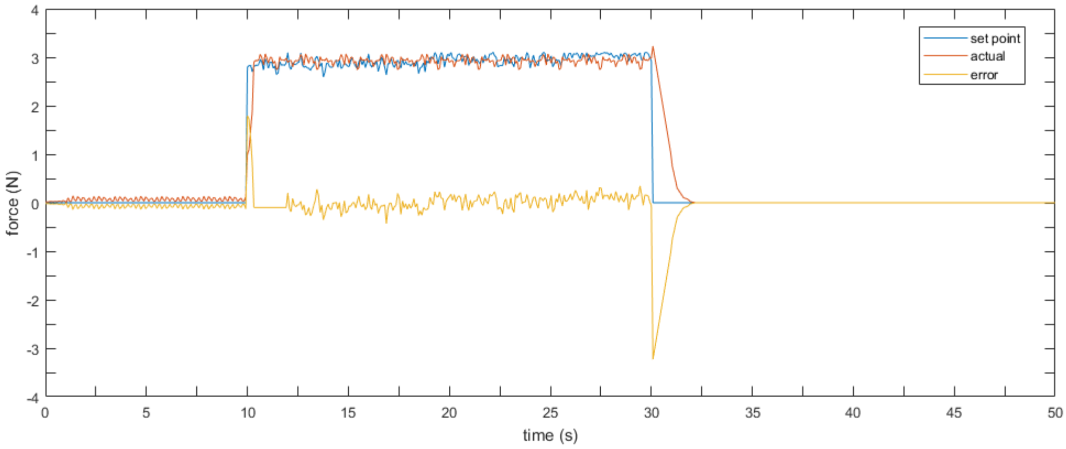


Figure 7.22: Fuzzy logic controller response for force control.

This was indicated by a drop in the system response resulting in high initial steady-state error on the onset of grasping as the controller became slow in responding. In observing this, the controller response was adjusted to reduce this lag by reducing the input error signals' range of universe of discourse. This improvement in the speed of response resulted in the response getting closer to the setpoint, meaning a reduction in the steady-state error. However, this came with an unwanted oscillatory behaviour, which is analogous to an increase in the PID controller gain [296].

As shown in Figure 7.21, the controller managed to accurately track the setpoint angle of the RPH. By further adjustments of the FLC, it was found out that overlapping of membership functions eliminated the oscillatory behaviour of the controller response, thereby resulting in a smooth curve. However, before initiating the first grasping action, the FLC was recording a constant oscillatory fingertip output angle. This constant oscillatory behaviour can be attributed to the robotic hand user not being fully steady, and thus passing small signals that try to initiate grasping even when the setpoint joint angle is still set to be at 0 or at rest.

As explained in the previous paragraph, increasing the speed of response of the FLC was enabled through reduction in the input error range of universe of discourse. For a case of a PD controller, this is analogous to reducing the derivative term. The developed controller is responsible for the adjustment of 4 input signals, with the other 2 being the rate of change of both the setpoint angle and grasping force errors. It was also shown that by reducing the width of the membership function of the derivative of the error signals, there was a marked increase in the steady-state error, and the system response became very slow.

Based on knowledge of a PD controller, this corresponds to an increase in the PD controller derivative term [297]. This is a major drawback in achieving dexterity when using robotic hands. To get rid of this slow system response, the membership functions of the derivative of error were gradually narrowed. Although the system still exhibited a transient response, this became steady over time. Since the smoothness of the response was observed to be a function of the degree of overlap, this reduction in the width of membership function resulted in the system response to still exhibit oscillatory behaviour.

In addition, the system response can sometimes be greater or less than the setpoint. This causes the error signal that is passed to the FLC to be either greater than or lower than 0. The FLC at some point showed results similar to the results for the PID controller in Figure 7.19 with either positive or negative errors due to overshooting. It was observed that these errors are a result of adjustment of the controller power output membership function ranges and not necessarily the rules. One other major factor affecting the performance of the developed FLC was the rule base.

Also, it was observed that by changing the output modes of the fuzzy-sets, the behaviour of the controller changed drastically. Even in the event that the best rule base had been selected, the fuzzy-sets also had to be adjusted to get the best controller response during grasping. As described in the methodology section, the fuzzy logic rules were defined based on expert knowledge of what the system must achieve and tested to see if the desired response was achieved.

During the development of the FLC, defining the rules which enabled mapping of the inputs to the output of the MISO fuzzy logic controller seemed to be the most difficult task. This is because elucidation of which input cause which corresponding output is largely an issue of experience than knowledge. This challenge in FLC, termed the Feigenbaum bottleneck, was emphasized by Feigenbaum [298].

In this study, Feigenbaum did an extensive research on FLC design and found that the degree of strength of an FLC is based on the experience of the expert rather than the strength of the fuzzy inference system [298, 299]. Therefore, the rule base can be thought of as the brains that map the inputs to the outputs, and thus proper rule definitions play a critical role in the strength of the FL controller.

Initially, during the first trials of the simulation with a less adjusted FLC, the system response was quite similar to a response in which a PD controller is implemented. This type of controller has no integrator, which results in the large steady-state error that is initially recorded. Usually, the inclusion of an integrator can eliminate these errors.

For these types of control problems, addition of an integrator means that a certain control action is maintained in the system even when there is no error in the response. This is important during the time at which the robotic hand is in contact with the grasping object, to maintain the hand in the grasping position. In light of this, the rule linked to the 0 error was further adjusted, such that a control action which outputs a power signal was achieved. This power signal being responsible for keeping the hand in the grasping positions shown in Figure 7.21, ensures that the error in the system response stays at 0.

The robustness of the developed FLC for angle control, was ascertained by the small margin in the variations of the power output used in this dissertation. Wide variations in output power results in a system that is difficult to tune that is highly oscillatory. Furthermore, proper iterative tuning ensured that the controller exhibits zero overshoot, that is, the controller will tend to have characteristics of a critically damped system, meaning that it is more stable than the PID controller which has an overshoot.

The developed FLC used overlapping membership functions to ensure that tuning becomes easy and that the system response becomes smooth once it reaches the desired setpoint. This FLC shown exhibited the same behaviour to the FLC developed by Alassar *et al.*, reported in literature [292]. Further improvements in FLCs can be done by introducing self-tuning or adaptive methods, such as discrete-time adaptive control, and the FLC adaptive sampling method [300]. However, these are difficult to develop for FLC of such robotic hands developed in this dissertation that are required to perform multiple tasks of varying dexterity. The developed FLCs was used to simultaneously control the fingertip angle and contact force in order to ascertain their effectiveness in real applications. Figures 7.23 and 7.24 respectively, show the RPH angle and force responses.

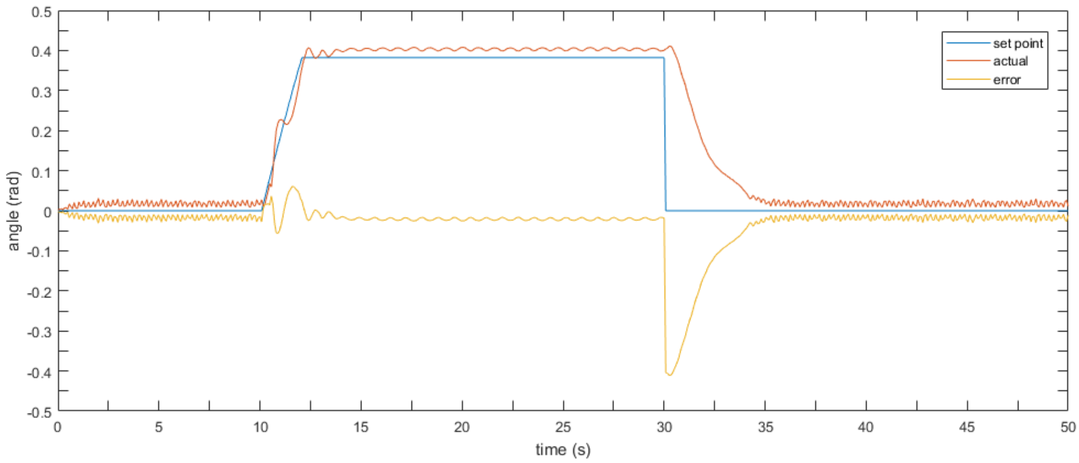


Figure 7.23: Fuzzy logic controller response for angle control with force compensation.

As shown in Figure 7.23, the FLC shows an initial overshoot that does not die out with time. The controller then exhibits an oscillatory response with an SSE ranging between -3.28% and -5.16% of the setpoint joint angle, during the 20s grasping period. In Figure 7.24, the FLC does not perform well for force control when used in conjunction with the FLC for angle control.

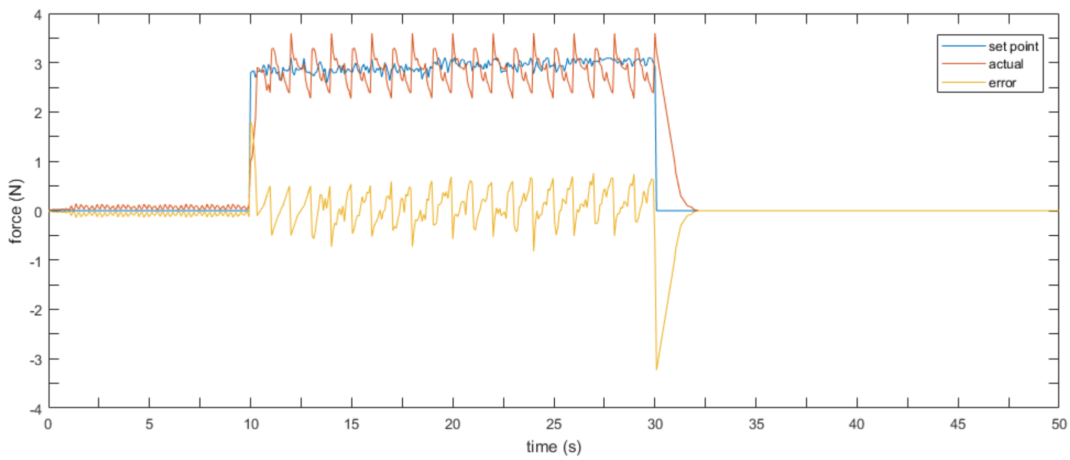


Figure 7.24: Fuzzy logic controller response for force control with angle compensation.

The Simulink block diagram in Figure 7.25 shows this, which results due to the 2 variables being mutually exclusive causing the controller outputs to counteract one another. The observed FLC force response was achieved after carefully changing the output gains of each controller such that a generally acceptable combined angle and force response was observed.

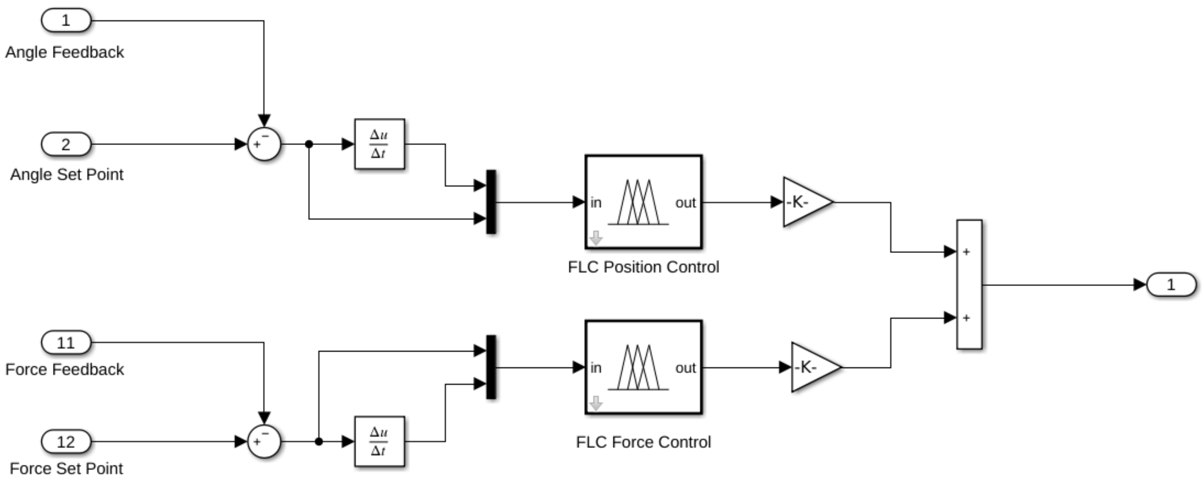


Figure 7.25: Simulink block diagram for fuzzy logic control of RPH fingertip position.

7.4 Chapter Summary

An extensive research was conducted to reflect the effect of different parameters and their variation on the control of the performance of RPHs. Several controllers found in literature proved to have accuracies that are affected by variations in the input parameters. In addition, most of the advanced control techniques are computationally expensive and impractical as they make robotic hands very expensive. The PID controller proved to be affected by parameter variations as it exhibited an initial overshoot.

Therefore, an FLC that uses less computer memory and that is easy to develop without any need for mathematical analysis was implemented. This Mamdani-type FLC was developed to control the power output sent to the SMA-based actuator in response to angle and force setpoints. In the development of the FLC, varying the range of discourse of the inputs signals affected both the system transient response and steady-state error. The most critical factor in the development of the fuzzy controller was in defining the rules that map the inputs to the outputs. The developed FLC for angle control proved to be robust and was able to attenuate initial grasping error. This FLC managed to accurately follow the prescribed setpoint and showed similar behaviour to the FLCs reported in literature.

In addition, the FLC for force control managed to precisely follow the continuously changing force setpoints with an error margin that is less than 6%. However, when used simultaneously, the developed FLCs worked against each other resulting in poor force estimation. Such as the FLCs found in literature, the developed FLCs are not able to accurately perform parallel control of different variables. However, with proper adjustments of controller output gains, a satisfactory response could be obtained using the current system. Also, with a proper rule base being set, adjustment of fuzzy output modes would improve the response of the controllers.

Chapter 8

Conclusion

In this work, a fuzzy logic-based controller was designed, simulated, and tested for application in a highly non-linear grip-compensated RPH actuated by means of an SMA. Though welcomed efforts have been made in recent years with respect to the advancement of robotic hands, these have in large been attributed to designs focusing on actuation achieved through either DC motors or servos, each having several drawbacks. In addition, prior studies often considered and presented singular aspects of the RPH system, typically not drawing attention to the integration of the often-complex sub-systems.

The development of an RPH, with qualities such as grasp recognition, grip compensation, realistic actuation and a robust control system bares significance to the field of medical prosthetics. To this end, this thesis systematically explored the development of several sub-systems, including sEMG-based grasp recognition and grip compensation, SMA and RPH model development and the fuzzy logic control of the overall system. Overarchingly, the developed RPH showed good performance, accurately decoding the sEMG signals in respect to the intended grasp as well as grip strength. Furthermore, the NN SMA model showed good agreement with the experimental results and expected displacement-power behaviour. The RPH model incorporated trajectory and force planning with the fuzzy logic control system governing the power supplied in order to achieve these goals. The control system was found to be robust and effective against enabling the grasping of various objects in a simulation environment.

The results of the grasp recognition model showed that a BPANN with time and frequency-based features including RMS, MAV, MAVS, SSC, WL and ZC, resulted in a highly effective sEMG classifier. Though, it was found that features such as SSC and ZC yielded the lowest correlation. The classifier was able to classify sEMG signals acquired from the forearm of 5 willing participants as either 3-jaw chuck, cylindrical, hook, rest, lateral, spherical or tip pinch. Overall, the developed NN model, consisted of two layers with 40 and 42 neurons respectively. The optimum neurons were found through iteration, yielding a model with a classification accuracy of 92.3%, and was improved to 100% after post-processing filtering techniques were applied. The 92.3% accuracy was directly impacted by the similarity of grasps, with the model confusion attributed largely to the 3-jaw chuck and tip pinch. The developed BPANN-based classifier outperformed several classification models including the DT, SVM and KNN models and was slightly better than the majority of models in previous works. This model has the potential to be used in the development of robotic hands in the prosthetic, medical and manufacturing fields.

The developed model may also be adapted to meet some of the requirements of virtual reality systems.

Furthering the lineage of development, grip compensation played a pivotal roll in granting the ability to directly control the amount of force applied while grasping an object. This was achieved through a NN regression model that mapped sEMG signals features such as RMS and MNF to output force. Data was collected from 5 participants using a constructed force sensitive glove and tailor made GVis system. Through analysis of the results, it was found that the MNF feature, though yielding a direct correlation with force, was not as effective as using RMS. This is partly due to the fact that MNF is closely coupled with physiological factors and as such differed significantly between the participants. Linear curve fitting clearly indicated that the relationship between sEMG and force is non-linear and grasp specific, with grasps requiring more force related to higher non-linearity. The regression model yielded an acceptable R2 of 0.73, which could be improved upon with refinement of the force sensitive glove and better-quality data as well as data pre-processing and additional model training.

In order to develop and test the performance of the fuzzy logic controller, a physical or model-based representation of the SMA actuator was required. In this thesis, an experimentally based model was developed, adding value to a gap in the field of SMAs. This model formed the basis for the actuator in the simulations of the RPH that followed. The use of an SMA-based actuator as a substitute for conventional actuators such as servos, DC motors or pneumatics was found to be valid in its approach. A test rig was developed to record the SMA-based actuator displacement/stroke as a function of input power and mass (axial stress proxy). Through collected input-output data a NARX NN model was trained to behave similarly to an SMA. The model accurately imitated and exhibited characteristics of shape memory effect, pseudo-elasticity, hysteresis, resistive heating and natural convection. Furthermore, the model consisted of a closed-loop structure with nine neurons in both the first and second layer. The number of neurons and associated hyperparameters were found through an iterative search of the solution space. The overall performance and RMSE of 0.04 for the model were a testament to the models likeness to the SMA actuator. The model was explored with pulse and sinusoidal inputs of varying frequency and was found to be robust and retain the characteristics of an SMA. Furthermore, the NARX NN model was compared to both a mathematical and FEA model and was found to be a better interpretation of an SMA, with a lower degree of error.

The anthropomorphic RPH model consisted of several parts including the mechanical model, setpoint estimation and trajectory planning, link and contact modelling. For the mechanical model, an RPH was designed in a 3-D modelling environment and exported to Simscape as parts. The PIP/IP and MCP joints were revolute joints with damping and ROM set based on previous research. Setpoint joint angles were defined for each grasp based on analysis conducted within the 3-D modelling environment, while the trajectory planning followed a logarithmic spiral path. The trajectory planning of the force setpoint was directly influenced by the switching of grasps and it is envisioned that the trajectory may be planned more effectively in order to facilitate a gradual ramp-up of the force to reach the required setpoint goal. The use of a single source of actuation at the base of each finger coupled the movement of the MCP and PIP/IP joints and as such the linkage needed to be accurately modelled. This was undertaken analytically and numerically solved. The Kelvin-Voigt contact model was found to adequately capture the static and dynamic behaviour of the hand sensing elements, allowing for the momentary impulse in force experienced upon contact. Overall, the RPH model provided a sufficient anatomically equivalent for which the other sub-systems could integrate into.

Following the development of the RPH system, a fuzzy logic controller was introduced with a weighted pair of Mamdani fuzzy logic controllers for each finger respectively. The controllers input was the error between the angular trajectory setpoint and actual angle, as well as the error between the force planning setpoint and measured force. The corresponding output was power which served as the input to the SMA-based actuator. The controllers were tuned iteratively through trial and error and were compared against the performance of a PID controller. It was found that the fuzzy logic controller outperformed the PID controller, however, still exhibited issues related to steady-state offset, oscillations, and overshoot. These were compounded when both position and force controllers were found to act in opposition to one another nearing contact between the hand and grasp object. To overcome this a weighted approach was used resulting in the positional controller having hierarchy over the force controller. The proposed controllers exhibited displacement and force errors typically less than 0.05rad and 0.1N in isolation, with the error increasing to 0.15rad and 0.25N respectively once combined.

The overall RPH system was found to be successful in achieving 3 jaw chuck pinch, cylindrical grasp, hook grasp, lateral pinch, spherical grasp and tip pinch with minimal error in both position and force. Components of this thesis may be extended to other medical and robotic systems.

Chapter 9

Recommendations for Future Works

1. Extend the number of grasps/pinches that the RPH is able to achieve, focusing in on those used during daily activities.

2. Develop alternative hand-force measurement system with greater sensitivity and force range. It is envisaged that improving the system will lead to better quality force-sEMG data and ultimately lead to a better regression fit.

3. Optimize the data cleaning process used in the force-sEMG model development. Remove statistical outliers and potentially apply a lowpass filter.

4. Redesign the RPH MCP-PIP linkage mechanism to improve the ROM by augmenting the link geometry to match the desired logarithmic spiral trajectories.

5. Use optimization techniques such as Particle Swarm Optimisation (PSO) and Genetic Algorithm for the tuning of the FL control system. Also, consider combining both force and position controllers into a single controller with hierarchy inherent in the rule base.

Chapter 10

References

[1] Farago, E., Chinchalkar, S., Lizotte, D.J. and Trejos, A.L., 2019. Development of an EMG-Based Muscle Health Model for Elbow Trauma Patients. Sensors, 19(15), p.3309.

[2] Ziegler-Graham, K., MacKenzie, E.J., Ephraim, P.L., Travison, T.G. and Brookmeyer, R., 2008. Estimating the prevalence of limb loss in the United States: 2005 to 2050. Archives of Physical Medicine and Rehabilitation, 89(3), pp.422-429.

[3] Billock, J.N., 1986. Upper limb prosthetic terminal devices: Hands versus hooks. Clin Prosthet Orthot, 10(2), pp.57-65.

[4] Fourie, R. and Stopforth, R., 2017, July. The mechanical design of a biologically inspired prosthetic hand, the touch hand 3. In 2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech) (pp. 38-43). IEEE.

[5] Kakoty, N.M., Hazarika, S.M. and Gan, J.Q., 2016. EMG feature set selection through linear relationship for grasp recognition. Journal of Medical and Biological Engineering, 36(6), pp.883-890.

[6] Côté-Allard, U., Fall, C.L., Drouin, A., Campeau-Lecours, A., Gosselin, C., Glette, K., Laviolette, F. and Gosselin, B., 2019. Deep learning for electromyographic hand gesture signal classification using transfer learning. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 27(4), pp.760-771.

[7] AlOmari, F. and Liu, G., 2014. Analysis of extracted forearm sEMG signal using LDA, QDA, K-NN classification algorithms. The Open Automation and Control Systems Journal, 6(1).

[8] Gandolla, M., Ferrante, S., Ferrigno, G., Baldassini, D., Molteni, F., Guanziroli, E., Cotti Cottini, M., Seneci, C. and Pedrocchi, A., 2017. Artificial neural network EMG classifier for functional hand grasp movements prediction. Journal of International Medical Research, 45(6), pp.1831-1847.

[9] Akhmadeev, K., Rampone, E., Yu, T., Aoustin, Y. and Le Carpentier, E., 2017. A testing system for a real-time gesture classification using surface EMG. IFAC-PapersOnLine, 50(1), pp.11498-11503.

[10] Atzori, M., Cognolato, M. and Müller, H., 2016. Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands. Frontiers in Neurorobotics, Article 9, pp 1-10.

[11] Cai, S., Chen, Y., Huang, S., Wu, Y., Zheng, H., Li, X. and Xie, L., 2019. SVM-based classification of sEMG signals for upper-limb self-rehabilitation training. Frontiers in Neurorobotics, Article 31, pp 1-10.

[12] Negi, S., Kumar, Y. and Mishra, V.M., 2016, September. Feature extraction and classification for EMG signals using linear discriminant analysis. In 2016 2nd International Conference on Advances in Computing, Communication, & Automation (ICACCA)(Fall) (pp. 1-6). IEEE.

[13] Khezri, M. and Jahed, M., 2010. A neuro–fuzzy inference system for sEMG-based identification of hand motion commands. IEEE Transactions on Industrial Electronics, 58(5), pp.1952-1960.

[14] Caesarendra, W., Tjahjowidodo, T., Nico, Y., Wahyudati, S. and Nurhasanah, L., 2018, April. EMG finger movement classification based on ANFIS. In IOP Conf. Ser. J. Phys. Conf. Ser (Vol. 1007, p. 012005).

[15] Merletti, R. and Parker, P.J. eds., 2004. Electromyography: physiology, engineering, and non-invasive applications (Vol. 11). John Wiley & Sons.

[16] Park, C.H., Ham, S.Y. and Son, Y.S., 2016, April. Relationship between input power and power density of SMA spring. In Active and Passive Smart Structures and Integrated Systems 2016 (Vol. 9799, p. 97990R). International Society for Optics and Photonics.

[17] Akbari, A.A. and Talasaz, M., 2014. Prediction of above-elbow motions in amputees, based on electromyographic (EMG) signals, using nonlinear autoregressive exogenous (NARX) model. Iranian Journal of Medical Physics, 11(2, 3), pp.233-241.

[18] Wills, A., Schön, T.B., Ljung, L. and Ninness, B., 2013. Identification of Hammerstein–Wiener Models. Automatica, 49(1), pp.70-81.

[19] Chengwei, G. and Qian, J., 2011, April. Optimized proportional-integral-derivative control strategies and simulation for lower limb functional electrical stimulation. In 2011 Fourth International Conference on Information and Computing (pp. 420-423). IEEE.

[20] Zadeh, L.A., 1996. Fuzzy logic, neural networks, and soft computing. In Fuzzy Sets, Fuzzy Logic, And Fuzzy Systems: Selected Papers by Lotfi A Zadeh (pp. 775-782).

[21] Ghori, K.M., Imran, M., Nawaz, A., Abbasi, R.A., Ullah, A. and Szathmary, L., 2020. Performance analysis of machine learning classifiers for non-technical loss detection. Journal of Ambient Intelligence and Humanized Computing, pp.1-16.

[22] Keenan, K.G. and Valero-Cuevas, F.J., 2007. Experimentally valid predictions of muscle force and EMG in models of motor-unit function are most sensitive to neural properties. Journal of Neurophysiology, 98(3), pp.1581-1590.

[23] Pan, Y., Guo, Z., Li, X. and Yu, H., 2017. Output-feedback adaptive neural control of a compliant differential SMA actuator. IEEE Transactions on Control Systems Technology, 25(6), pp.2202-2210.

[24] Parshotam, D., S., 2020. Linear Shape Memory Alloy (Nitinol) Actuators in a 3D Printed Hand Prosthesis. Master's Dissertation in Engineering Sciences.

[25] Rashidifar, M.A., Rashidifar, A.A. and Ahmadi, D., 2013. Modeling and Control of 5DOF Robot Arm Using Fuzzy Logic Supervisory Control. IAES International Journal of Robotics and Automation, 2(2), pp. 56-68.

[26] Vaishnav, S.R. and Khan, Z.J., 2007, October. Design and performance of PID and fuzzy logic controller with smaller rule set for higher order system. In Proceedings of the World Congress on Engineering and Computer Science (pp. 24-26).

[27] Mohindru, P. and Sharma, G., 2015. Simulation performance of PID and fuzzy logic controller for higher order system. Simulation, 1(7).

[28] Englehart, K., Hudgins, B., Parker, P.A. and Stevenson, M., 1999. Classification of the myoelectric signal using time-frequency based representations. Medical Engineering & Physics, 21(6-7), pp.431-438.

[29] Ives, J.C. and Wigglesworth, J.K., 2003. Sampling rate effects on surface EMG timing and amplitude measures. Clinical Biomechanics, 18(6), pp.543-552.

[30] Chen, H., Zhang, Y., Zhang, Z., Fang, Y., Liu, H. and Yao, C., 2017, October. Exploring the relation between EMG sampling frequency and hand motion recognition accuracy. In 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 1139-1144). IEEE.

[31] Pires, M.B., Junior, J.J.A.M. and Stevan Jr, S.L., 2018. Development of an 8 channel sEMG wireless device based on ADS1299 with Virtual Instrumentation. arXiv preprint arXiv:1808.03711.

[32] Pizzolato, S., Tagliapietra, L., Cognolato, M., Reggiani, M., Müller, H. and Atzori, M., 2017. Comparison of six electromyography acquisition setups on hand movement classification tasks. PloS One, 12(10), p.e0186132.

[33] Nawab, S.H., Wotiz, R.P. and De Luca, C.J., 2008. Decomposition of indwelling EMG signals. Journal of Applied Physiology, 105(2), pp.700-710.

[34] Phinyomark, A., Limsakul, C. and Phukpattaranont, P., 2011. Application of wavelet analysis in EMG feature extraction for pattern classification. Measurement Science Review, 11(2), pp.45-52.

[35] Prats, M., Sanz, P.J. and Del Pobil, A.P., 2009, May. Vision-tactile-force integration and robot physical interaction. In 2009 IEEE International Conference on Robotics and Automation (pp. 3975-3980). IEEE.

[36] Sun, W., Liu, H., Tang, R., Lang, Y., He, J. and Huang, Q., 2019. sEMG-Based Hand-Gesture Classification Using a Generative Flow Model. Sensors, 19(8), p.1952.

[37] Siu, H.C., Shah, J.A. and Stirling, L.A., 2016. Classification of anticipatory signals for grasp and release from surface electromyography. Sensors, 16(11), p.1782.

[38] Improving the recognition of grips and movements of the hand using myoelectric signals.

[39] Schwarz, R.J. and Taylor, C.L., 1955. The anatomy and mechanics of the human hand. Artificial Limbs, 2(2), pp.22-35.

[40] De Luca, C.J., Gilmore, L.D., Kuznetsov, M. and Roy, S.H., 2010. Filtering the surface EMG signal: Movement artifact and baseline noise contamination. Journal of Biomechanics, 43(8), pp.1573-1579.

[41] Anand, D., Bhateja, V., Srivastava, A. and Tiwari, D.K., 2018. An approach for the preprocessing of EMG signals using canonical correlation analysis. In Smart Computing and Informatics (pp. 201-208). Springer, Singapore.

[42] Myers, L.J., Lowery, M., O'malley, M., Vaughan, C.L., Heneghan, C., Gibson, A.S.C., Harley, Y.X.R. and Sreenivasan, R., 2003. Rectification and non-linear pre-processing of EMG signals for cortico-muscular analysis. Journal of Neuroscience Methods, 124(2), pp.157-165.

[43] Kale, S.N. and Dudul, S.V., 2009. Intelligent noise removal from EMG signal using focused time-lagged recurrent neural network. Applied Computational Intelligence and Soft Computing, 2009.

[44] Phinyomark, A., Phukpattaranont, P. and Limsakul, C., 2011. Wavelet-based denoising algorithm for robust EMG pattern recognition. Fluctuation and Noise Letters, 10(02), pp.157-167.

[45] Andrade, A.O., Nasuto, S., Kyberd, P., Sweeney-Reed, C.M. and Van Kanijn, F.R., 2006. EMG signal filtering based on empirical mode decomposition. Biomedical Signal Processing and Control, 1(1), pp.44-55.

[46] Gradolewski, D., Tojza, P.M., Jaworski, J., Ambroziak, D., Redlarski, G. and Krawczuk, M., 2015. Arm EMG wavelet-based denoising system. In Mechatronics-Ideas for Industrial Application (pp. 289-296). Springer, Cham

[47] Zhang, D., Zhao, X., Han, J. and Zhao, Y., 2014, May. A comparative study on PCA and LDA based EMG pattern recognition for anthropomorphic robotic hand. In 2014 IEEE International Conference on Robotics and Automation (ICRA) (pp. 4850-4855). IEEE.

[48] Hudgins, B., Parker, P. and Scott, R.N., 1993. A new strategy for multifunction myoelectric control. IEEE Transactions on Biomedical Engineering, 40(1), pp.82-94.

[49] Abbaspour, S., Lindén, M., Gholamhosseini, H., Naber, A. and Ortiz-Catalan, M., 2020. Evaluation of surface EMG-based recognition algorithms for decoding hand movements. Medical & Biological Engineering & Computing, 58(1), pp.83-100.

[50] Too, J., Abdullah, A.R., Mohd Saad, N. and Tee, W., 2019. EMG feature selection and classification using a Pbest-guide binary particle swarm optimization. Computation, 7(1), p.12.

[51] Naik, G.R., Selvan, S.E., Gobbo, M., Acharyya, A. and Nguyen, H.T., 2016. Principal component analysis applied to surface electromyography: a comprehensive review. IEEE Access, 4, pp.4025-4037.

[52] Arozi, M., Caesarendra, W., Ariyanto, M., Munadi, M., Setiawan, J.D. and Glowacz, A., 2020. Pattern Recognition of Single-Channel sEMG Signal Using PCA and ANN Method to Classify Nine Hand Movements. Symmetry, 12(4), p.541.

[53] Too, J., Abdullah, A.R., Mohd Saad, N., Mohd Ali, N. and Tee, W., 2018. A new competitive binary Grey Wolf Optimizer to solve the feature selection problem in EMG signals classification. Computers, 7(4), p.58.

[54] Zhai, X., Jelfs, B., Chan, R.H. and Tin, C., 2017. Self-recalibrating surface EMG pattern recognition for neuroprosthesis control based on convolutional neural network. Frontiers in Neuroscience, 11, Article No. 379.

[55] Hartwell, A., Kadirkamanathan, V. and Anderson, S.R., 2018, August. Compact deep neural networks for computationally efficient gesture classification from electromyography signals. In 2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob) (pp. 891-896). IEEE.

[56] Asogbon, M.G., Samuel, O.W., Jiang, Y., Wang, L., Geng, Y., Sangaiah, A.K., Chen, S., Fang, P. and Li, G., 2020. Appropriate Feature Set and Window Parameters Selection for Efficient Motion Intent Characterization towards Intelligently Smart EMG-PR System. Symmetry, 12(10), p.1710.

[57] Kamavuako, E.N., Scheme, E.J. and Englehart, K.B., 2016. Determination of optimum threshold values for EMG time domain features; a multi-dataset investigation. Journal of Neural Engineering, 13(4), p.046011.

[58] Phinyomark, A., Hirunviriya, S., Limsakul, C. and Phukpattaranont, P., 2010, May. Evaluation of EMG feature extraction for hand movement recognition based on Euclidean distance and standard deviation. In ECTI-CON2010: The 2010 ECTI International Confernce on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (pp. 856-860). IEEE.

[59] Amado Laezza, R., 2018. Deep neural networks for myoelectric pattern recognition-An implementation for multifunctional control (Master's thesis).

[60] Yamanoi, Y., Ogiri, Y. and Kato, R., 2020. EMG-based posture classification using a convolutional neural network for a myoelectric hand. Biomedical Signal Processing and Control, 55, p.101574.

[61] Ahsan, M.R., Ibrahimy, M.I. and Khalifa, O.O., 2012, February. EMG motion pattern classification through design and optimization of neural network. In 2012 International Conference on Biomedical Engineering (ICoBE) (pp. 175-179). IEEE.

[62] Nazmi, N., Abdul Rahman, M.A., Yamamoto, S.I., Ahmad, S.A., Zamzuri, H. and Mazlan, S.A., 2016. A review of classification techniques of EMG signals during isotonic and isometric contractions. Sensors, 16(8), p.1304.

[63] Van Der Malsburg, C., 1986. Frank Rosenblatt: principles of neurodynamics: perceptrons and the theory of brain mechanisms. In Brain theory (pp. 245-248). Springer, Berlin, Heidelberg.

[64] Møller, M.F., 1993. A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks, 6(4), pp.525-533.

[65] Heath, R.A., Leong, P. and Jabri, M., 1993. Proceedings of the fourth Australian conference on neural networks.

[66] Simon, A.M., Hargrove, L.J., Lock, B.A. and Kuiken, T.A., 2011. A decision-based velocity ramp for minimizing the effect of misclassifications during real-time pattern recognition control. IEEE Transactions on Biomedical Engineering, 58(8), pp.2360-2368.

[67] Amsüss, S., Goebel, P.M., Jiang, N., Graimann, B., Paredes, L. and Farina, D., 2013. Self-correcting pattern recognition system of surface EMG signals for upper limb prosthesis control. IEEE Transactions on Biomedical Engineering, 61(4), pp.1167-1176.

[68] Wu, J., Li, X., Liu, W. and Wang, Z.J., 2019, June. sEMG Signal Processing Methods: A Review. In Journal of Physics: Conference Series (Vol. 1237, No. 3, p. 032008). IOP Publishing.

[69] Chen, H., Zhang, Y., Zhang, Z., Fang, Y., Liu, H. and Yao, C., 2017, October. Exploring the relation between EMG sampling frequency and hand motion recognition accuracy. In 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 1139-1144). IEEE.

[70] Li, G., Li, Y., Yu, L. and Geng, Y., 2011. Conditioning and sampling issues of EMG signals in motion recognition of multifunctional myoelectric prostheses. Annals of Biomedical Engineering, 39(6), pp.1779-1787.

[71] Pizzolato, S., Tagliapietra, L., Cognolato, M., Reggiani, M., Müller, H. and Atzori, M., 2017. Comparison of six electromyography acquisition setups on hand movement classification tasks. PloS one, 12(10), p.e0186132.

[72] Choi, N.C. and Lee, S.H., 2015. Discomfort evaluation of truck ingress/egress motions based on biomechanical analysis. Sensors, 15(6), pp.13568-13590.

[73] Phinyomark, A. and Scheme, E., 2018. EMG pattern recognition in the era of big data and deep learning. Big Data and Cognitive Computing, 2(3), p.21.

[74] Mahapatra, S., Mohanta, D., Mohanty, P.K. and Nayak, S.K., 2017. Classification of EMG Signals Using ANFIS for the Detection of Neuromuscular Disorders. In Recent Developments in Intelligent Computing, Communication and Devices (pp. 53-60). Springer, Singapore.

[75] Chowdhury, R.H., Reaz, M.B., Ali, M.A.B.M., Bakar, A.A., Chellappan, K. and Chang, T.G., 2013. Surface electromyography signal processing and classification techniques. Sensors, 13(9), pp.12431-12466.

[76] Zhang, K. and Wang, Z.Z., 2005. The application of BP neural network improved with LM algorithm in surface EMG signal classification. Zhongguo yi liao qi xie za zhi= Chinese Journal of Medical Instrumentation, 29(6), p.399.

[77] Staudenmann, D., Kingma, I., Daffertshofer, A., Stegeman, D.F. and van Dieën, J.H., 2006. Improving EMG-based muscle force estimation by using a high-density EMG grid and principal component analysis. IEEE Transactions on Biomedical Engineering, 53(4), pp.712-719.

[78] Kwon, D.Y., Park, B.K., Kim, J.W., Eom, G.M., Hong, J., Koh, S.B. and Park, K.W., 2014. Quantitative electromyographic analysis of reaction time to external auditory stimuli in drug-naïve Parkinson's disease. Parkinson's Disease, 2014.

[79] Kim, K. and Colgate, J.E., 2012. Haptic feedback enhances grip force control of sEMG-controlled prosthetic hands in targeted reinnervation amputees. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 20(6), pp.798-805.

[80] Armstrong, T.J. and Chaffin, D.B., 1978. An investigation of the relationship between displacements of the finger and wrist joints and the extrinsic finger flexor tendons. Journal of Biomechanics, 11(3), pp.119-128.

[81] Cadova, M., Vilimek, M. and Daniel, M., 2014. A comparative study of muscle force estimates using Huxley's and Hill's muscle model. Computer Methods in Biomechanics and Biomedical Engineering, 17(4), pp.311-317.

[82] Lecompte, S., Abou-Samra, M., Boursereau, R., Noel, L. and Brichard, S.M., 2017. Skeletal muscle secretome in Duchenne muscular dystrophy: a pivotal anti-inflammatory role of adiponectin. Cellular and Molecular Life Sciences, 74(13), pp.2487-2501.

[83] Parent, A., 1996. Carpenter's human neuroanatomy. Williams & Wilkins.

[84] Nijssen, J., Comley, L.H. and Hedlund, E., 2017. Motor neuron vulnerability and resistance in amyotrophic lateral sclerosis. Acta Neuropathologica, 133(6), pp.863-885.

[85] Biga, L.M., Dawson, S., Harwell, A., Hopkins, R., Kaufmann, J., LeMaster, M., Matern, P., Morrison-Graham, K., Quick, D. and Runyeon, J., 2020. Anatomy & Physiology.

[86] Scott, W., Stevens, J. and Binder–Macleod, S.A., 2001. Human skeletal muscle fiber type classifications. Physical Therapy, 81(11), pp.1810-1816.

[87] Hwang, K., Huan, F. and Kim, D.J., 2013. Muscle fibre types of the lumbrical, interossei, flexor, and extensor muscles moving the index finger. Journal of Plastic Surgery and Hand Surgery, 47(4), pp.268-272.

[88] McIntosh, J.S., Ringqvist, M. and Schmidt, E.M., 1985. Fiber type composition of monkey forearm muscle. The Anatomical Record, 211(4), pp.403-409.

[89] Lieber, R.L., Ljung, B.O. and Fridén, J., 1997. Intraoperative sarcomere length measurements reveal differential design of human wrist extensor muscles. Journal of Experimental Biology, 200(1), pp.19-25.

[90] Potvin, J.R. and Fuglevand, A.J., 2017. A motor unit-based model of muscle fatigue. PLoS Computational Biology, 13(6), p.e1005581.

[91] Henneman, E., 1985. The size-principle: a deterministic output emerges from a set of probabilistic connections. Journal of Experimental Biology, 115(1), pp.105-112.

[92] Mendell, L.M., 2005. The size principle: a rule describing the recruitment of motoneurons. Journal of Neurophysiology, 93(6), pp.3024-3026.

[93] Colquhoun, R.J., Magrini, M.A., Haun, C.T., Muddle, T.W., Tomko, P.M., Luera, M.J., Mackey, C.S., Vann, C.G., Martin, J.S., Young, K.C. and DeFreitas, J.M., 2018. Muscle phenotype is related to motor unit behavior of the vastus lateralis during maximal isometric contractions. Physiological Reports, 6(5), p.e13636.

[94] De Luca, C.J. and Hostage, E.C., 2010. Relationship between firing rate and recruitment threshold of motoneurons in voluntary isometric contractions. Journal of Neurophysiology, 104(2), pp.1034-1046.

[95] Westgaard, R.H. and De Luca, C.J., 2001. Motor control of low-threshold motor units in the human trapezius muscle. Journal of Neurophysiology, 85(4), pp.1777-1781

[96] Lo Presti, D., Dall'Orso, S., Muceli, S., Arichi, T., Neumane, S., Lukens, A., Sabbadini, R., Massaroni, C., Caponero, M.A., Formica, D. and Burdet, E., 2020. An fMRI Compatible Smart Device for Measuring Palmar Grasping Actions in Newborns. Sensors, 20(21), p.6040.

[97] Schmoll, M., Unger, E., Bijak, M., Stoiber, M., Lanmüller, H. and Jarvis, J.C., 2017. A novel miniature in-line load-cell to measure in-situ tensile forces in the tibialis anterior tendon of rats. Plos One, 12(9), p.e0185209.

[98] Arjunan, S.P., Kumar, D.K. and Naik, G., 2014. Computation and evaluation of features of surface electromyogram to identify the force of muscle contraction and muscle fatigue. BioMed Research International, 2014.

[99] Yokoyama, M., Koyama, R. and Yanagisawa, M., 2017. An evaluation of hand-force prediction using artificial neural-network regression models of surface EMG signals for handwear devices. Journal of Sensors, 2017.

[100] Sadun, A.S., Jalani, J. and Sukor, J.A., 2016, July. Force Sensing Resistor (FSR): a brief overview and the low-cost sensor for active compliance control. In First International Workshop on Pattern Recognition (Vol. 10011, p. 1001112). International Society for Optics and Photonics.

[101] Xiao, Z.G. and Menon, C., 2020. Towards the investigation on the effect of the forearm rotation on the wrist FMG signal pattern using a high-density FMG sensing matrix. Cogent Engineering, 7(1), p.1795051.

[102] Xiao, Z.G. and Menon, C., 2019. A review of force myography research and development. Sensors, 19(20), p.4557.

[103] Leone, F., Gentile, C., Ciancio, A.L., Gruppioni, E., Davalli, A., Sacchetti, R., Guglielmelli, E. and Zollo, L., 2019. Simultaneous sEMG classification of wrist/hand gestures and forces. Frontiers in Neurorobotics, 13, p.42.

[104] Barnea, A., Oprisan, C. and Olaru, D., 2012. Force sensitive resistors calibration for use in gripping devices. Mechanical Testing and Diagnosis, 2(3), pp.18-27.

[105] Hashemi, J., 2013. Novel methods in SEMG-Force estimation (PhD Thesis).

[106] Ye, Y., Liu, C., Zemiti, N. and Yang, C., 2019, October. Optimal Feature Selection for EMG-Based Finger Force Estimation Using LightGBM Model. In 2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN) (pp. 1-7). IEEE.

[107] Martinez, I.J.R., Mannini, A., Clemente, F., Sabatini, A.M. and Cipriani, C., 2020. Grasp force estimation from the transient EMG using high-density surface recordings. Journal of Neural Engineering, 17(1), p.016052.

[108] Kim, S., Kim, J., Kim, M., Kim, S. and Park, J., 2019. Grasping force estimation by sEMG signals and arm posture: tensor decomposition approach. Journal of Bionic Engineering, 16(3), pp.455-467.

[109] Luo, J., Liu, C. and Yang, C., 2019. Estimation of EMG-based force using a neural-network-based approach. IEEE Access, 7, pp.64856-64865.

[110] Ma, R., Zhang, L., Li, G., Jiang, D., Xu, S. and Chen, D., 2020. Grasping force prediction based on sEMG signals. Alexandria Engineering Journal.

[111] Wang, K., Zhang, X., Ota, J. and Huang, Y., 2018. Estimation of handgrip force from SEMG based on wavelet scale selection. Sensors, 18(2), p.663.

[112] Wu, C., Zeng, H., Song, A. and Xu, B., 2017. Grip force and 3D push-pull force estimation based on sEMG and GRNN. Frontiers in Neuroscience, 11, p.343.

[113] Yamanoi, Y., Morishita, S., Kato, R. and Yokoi, H., 2015, August. Selective Linear-Regression Model for hand posture discrimination and grip force estimation using surface electromyogram signals. In 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 4812-4815). IEEE.

[114] Kamavuako, E.N., Farina, D., Yoshida, K. and Jensen, W., 2009. Relationship between grasping force and features of single-channel intramuscular EMG signals. Journal of Neuroscience Methods, 185(1), pp.143-150.

[115] Barone, R., Ciancio, A.L., Romeo, R.A., Davalli, A., Sacchetti, R., Guglielmelli, E. and Zollo, L., 2016. Multilevel control of an anthropomorphic prosthetic hand for grasp and slip prevention. Advances in Mechanical Engineering, 8(9), p.1687814016665082.

[116] Adewuyi, A.A., Hargrove, L.J. and Kuiken, T.A., 2015. An analysis of intrinsic and extrinsic hand muscle EMG for improved pattern recognition control. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 24(4), pp.485-494.

[117] Seedat, N., Mohamed, I. and Mohamed, A.K., 2018, August. Custom force sensor and sensory feedback system to enable grip control of a robotic prosthetic hand. In 2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob) (pp. 248-253). IEEE.

[118] Reaz, M.B.I., Hussain, M.S. and Mohd-Yasin, F., 2006. Techniques of EMG signal analysis: detection, processing, classification and applications. Biological Procedures Online, 8(1), pp.11-35.

[119] Chowdhury, R.H., Reaz, M.B., Ali, M.A.B.M., Bakar, A.A., Chellappan, K. and Chang, T.G., 2013. Surface electromyography signal processing and classification techniques. Sensors, 13(9), pp.12431-12466.

[120] Amorim, C.F. and Marson, R.A., 2012. Application of surface electromyography in the dynamics of human movement. Computational intelligence in electromyography analysis—a perspective on current applications and future challenges. 1st ed. Rijeka: InTech, pp.391-408.

[121] Lieber, R.L., Jacobson, M.D., Fazeli, B.M., Abrams, R.A. and Botte, M.J., 1992. Architecture of selected muscles of the arm and forearm: anatomy and implications for tendon transfer. The Journal of Hand Surgery, 17(5), pp.787-798.

[122] Rutkove, S.B., 2007. Introduction to volume conduction. In The clinical neurophysiology primer (pp. 43-53). Humana Press.

[123] Squire, J.M., 2016. Muscle contraction: Sliding filament history, sarcomere dynamics and the two Huxleys. Global Cardiology Science & Practice, 2016(2).

[124] Kneiser, M., Boon, A.J., Brown, A.D., Hobson-Webb, L.A., Smith, B.E. and Litchy, W.J., 2015. AANEM, glossary of terms in neuromuscular and electrodiagnostic medicine. Muscle Nerve, p.62.

[125] FENGJUN, B., 2013. Muscle Force Estimation and Fatigue Detection Based on sEMG Signals (Doctoral dissertation).

[126] Sahebjada, S., 2016. Effect of age and gender on sEMG signals and force steadiness.

[127] Ferrari, E., Cooper, G., Reeves, N.D. and Hodson-Tole, E.F., 2018. Surface electromyography can quantify temporal and spatial patterns of activation of intrinsic human foot muscles. Journal of Electromyography and Kinesiology, 39, pp.149-155.

[128] Xu, L., Chen, X., Cao, S., Zhang, X. and Chen, X., 2018. Feasibility study of advanced neural networks applied to sEMG-based force estimation. Sensors, 18(10), p.3226.

[129] Shaheen, F., Verma, B. and Asafuddoula, M., 2016, November. Impact of automatic feature extraction in deep learning architecture. In 2016 International conference on digital image computing: techniques and applications (DICTA) (pp. 1-8). IEEE.

[130] McManus, L., Hu, X., Rymer, W.Z., Lowery, M.M. and Suresh, N.L., 2015. Changes in motor unit behavior following isometric fatigue of the first dorsal interosseous muscle. Journal of Neurophysiology, 113(9), pp.3186-3196.

[131] Fernando, J.B., Yoshioka, M. and Ozawa, J., 2016, August. Estimation of muscle fatigue by ratio of mean frequency to average rectified value from surface electromyography. In 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 5303-5306). IEEE.

[132] Phinyomark, A., Thongpanja, S., Hu, H., Phukpattaranont, P. and Limsakul, C., 2012. The usefulness of mean and median frequencies in electromyography analysis. Computational intelligence in electromyography analysis-A perspective on current applications and future challenges, pp.195-220.

[133] Farina, D., Fosci, M. and Merletti, R., 2002. Motor unit recruitment strategies investigated by surface EMG variables. Journal of Applied Physiology, 92(1), pp.235-247.

[134] Kamavuako, E.N., Farina, D., Yoshida, K. and Jensen, W., 2012. Estimation of grasping force from features of intramuscular EMG signals with mirrored bilateral training. Annals of Biomedical Engineering, 40(3), pp.648-656.

[135] Mathiesen, J.R., Bøg, M.F., Erkocevic, E., Niemeier, M.J., Smidstrup, A. and Kamavuako, E.N., 2010. Prediction of grasping force based on features of surface and intramuscular EMG. In 7th semester conference paper (pp. 1-9).

[136] Li, C., Ren, J., Huang, H., Wang, B., Zhu, Y. and Hu, H., 2018. PCA and deep learning based myoelectric grasping control of a prosthetic hand. Biomedical Engineering Online, 17(1), p.107.

[137] Gailey, A., Artemiadis, P. and Santello, M., 2017. Proof of concept of an online EMG-based decoding of hand postures and individual digit forces for prosthetic hand control. Frontiers in Neurology, 8, p.7.

[138] Prasad, A., Gupta, S.S. and Tyagi, R.K. eds., 2019. Advances in Engineering Design: Select Proceedings of FLAME 2018. Springer.

[139] Beretta-Piccoli, M., Boccia, G., Ponti, T., Clijsen, R., Barbero, M. and Cescon, C., 2018. Relationship between isometric muscle force and fractal dimension of surface electromyogram. BioMed Research International.

[140] Xu, L., Zhu, G. and Duan, X., 2016, November. Grasping force estimation for prosthetic hands via feature extraction of surface EMG. In 2016 International Conference on Advanced Electronic Science and Technology (AEST 2016). Atlantis Press.

[141] Baldacchino, T., Jacobs, W.R., Anderson, S.R., Worden, K. and Rowson, J., 2018. Simultaneous force regression and movement classification of fingers via surface EMG within a unified Bayesian framework. Frontiers in Bioengineering and Biotechnology, 6, p.13.

[142] Costa, M.V., Pereira, L.A., Oliveira, R.S., Pedro, R.E., Camata, T.V., Abrão, T., Brunetto, M.A. and Altimari, L.R., 2010, August. Fourier and wavelet spectral analysis of EMG signals in maximal constant load dynamic exercise. In 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology (pp. 4622-4625). IEEE.

[143] Chen, Y., Dai, C. and Chen, W., 2020. Cross-Comparison of EMG-to-Force Methods for Multi-DoF Finger Force Prediction Using One-DoF Training. IEEE Access, 8, pp.13958-13968.

[144] Winters, J.M. and Stark, L., 1987. Muscle models: what is gained and what is lost by varying model complexity. Biological Cybernetics, 55(6), pp.403-420.

[145] Winters, J.M., 1990. Hill-based muscle models: a systems engineering perspective. In Multiple muscle systems (pp. 69-93). Springer, New York, NY.

[146] Kleinbach, C., Martynenko, O., Promies, J., Haeufle, D.F., Fehr, J. and Schmitt, S., 2017. Implementation and validation of the extended Hill-type muscle model with robust routing capabilities in LS-DYNA for active human body models. Biomedical Engineering Online, 16(1), p.109.

[147] Buchanan, T.S., Lloyd, D.G., Manal, K. and Besier, T.F., 2004. Neuromusculoskeletal modeling: estimation of muscle forces and joint moments and movements from measurements of neural command. Journal of Applied Biomechanics, 20(4), pp.367-395.

[148] M. Schipplein, O.D. and Andriacchi, T.P., 1991. Interaction between active and passive knee stabilizers during level walking. Journal of Orthopaedic Research, 9(1), pp.113-119.

[149] Manal, K. and Buchanan, T.S., 2003. A one-parameter neural activation to muscle activation model: estimating isometric joint moments from electromyograms. Journal of Biomechanics, 36(8), pp.1197-1202

[150] Besier, T.F., Lloyd, D.G. and Ackland, T.R., 2003. Muscle activation strategies at the knee during running and cutting maneuvers. Medicine & Science in Sports & Exercise, 35(1), pp.119-127.

[151] Tomalka, A., Rode, C., Schumacher, J. and Siebert, T., 2017. The active force–length relationship is invisible during extensive eccentric contractions in skinned skeletal muscle fibres. Proceedings of the Royal Society B: Biological Sciences, 284(1854), p.20162497.

[152] A Heidlauf, T. and Röhrle, O., 2014. A multiscale chemo-electro-mechanical skeletal muscle model to analyze muscle contraction and force generation for different muscle fiber arrangements. Frontiers in Physiology, 5, p.498.

[153] Moo, E.K., Leonard, T.R. and Herzog, W., 2020. The sarcomere force–length relationship in an intact muscle–tendon unit. Journal of Experimental Biology, 223(6).

[154] Cifrek, M., Medved, V., Tonković, S. and Ostojić, S., 2009. Surface EMG based muscle fatigue evaluation in biomechanics. Clinical Biomechanics, 24(4), pp.327-340.

[155] Kuriki, H.U., Mello, E.M., De Azevedo, F.M., Takahashi, L.S.O., Alves, N. and de Faria Negrão Filho, R., 2012. The relationship between electromyography and muscle force. INTECH Open Access Publisher.

[156] Ahamed, N.U., Sundaraj, K., Alqahtani, M., Altwijri, O., Ali, M. and Islam, M., 2014. EMG-force relationship during static contraction: Effects on sensor placement locations on biceps brachii muscle. Technology and Health Care, 22(4), pp.505-513.

[157] Jones, L.A. and Hunter, I.W., 1982. The relation of muscle force and EMG to perceived force in human finger flexors. European Journal of Applied Physiology and Occupational Physiology, 50(1), pp.125-131.

[158] Padmanabhan, P. and Puthusserypady, S., 2004, September. Nonlinear analysis of EMG signals-a chaotic approach. In The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (Vol. 1, pp. 608-611). IEEE.

[159] Bigland, B. and Lippold, O.C.J., 1954. The relation between force, velocity and integrated electrical activity in human muscles. The Journal of physiology, 123(1), p.214.

[160] Kamavuako, E.N., Rosenvang, J.C., Bøg, M.F., Smidstrup, A., Erkocevic, E., Niemeier, M.J., Jensen, W. and Farina, D., 2013. Influence of the feature space on the estimation of hand grasping force from intramuscular EMG. Biomedical Signal Processing and Control, 8(1), pp.1-5.

[161] Coggshall, J.C. and Bekey, G.A., 1970. EMG-force dynamics in human skeletal muscle. Medical and biological engineering, 8(3), pp.265-270.

[162] Lawrence, J.H. and De Luca, C.J., 1983. Myoelectric signal versus force relationship in different human muscles. Journal of Applied Physiology, 54(6), pp.1653-1659.

[163] Vredenbregt, J. and Rau, G., 1973. Surface electromyography in relation to force, muscle length and endurance. In New Concepts of the Motor Unit, Neuromuscular Disorders, Electromyographic Kinesiology (Vol. 1, pp. 607-622). Karger Publishers.

[164] Al Harrach, M., Carriou, V., Boudaoud, S., Laforet, J. and Marin, F., 2017. Analysis of the sEMG/force relationship using HD-sEMG technique and data fusion: A simulation study. Computers in biology and medicine, 83, pp.34-47.

[165] Klute, G.K., Czerniecki, J.M. and Hannaford, B., 2002. Artificial muscles: Actuators for biorobotic systems. The International Journal of Robotics Research, 21(4), pp.295-309.

[166] Wen, R., Yuan, K., Wang, Q., Heng, S. and Li, Z., 2020. Force-Guided High-Precision Grasping Control of Fragile and Deformable Objects Using sEMG-Based Force Prediction. IEEE Robotics and Automation Letters, 5(2), pp.2762-2769.

[167] Choi, C., Kwon, S., Park, W., Lee, H.D. and Kim, J., 2010. Real-time pinch force estimation by surface electromyography using an artificial neural network. Medical engineering & physics, 32(5), pp.429-436.

[168] Zhang, S., Guo, S., Gao, B., Huang, Q., Pang, M., Hirata, H. and Ishihara, H., 2016. Muscle strength assessment system using sEMG-based force prediction method for wrist joint. Journal of Medical and Biological Engineering, 36(1), pp.121-131.

[169] Oweis, R.J., Rihani, R. and Alkhawaja, A., 2014. ANN-based EMG classification for myoelectric control. International Journal of Medical Engineering and Informatics, 6(4), pp.365-380.

[170] Arozi, M., Caesarendra, W., Ariyanto, M., Munadi, M., Setiawan, J.D. and Glowacz, A., 2020. Pattern Recognition of Single-Channel sEMG Signal Using PCA and ANN Method to Classify Nine Hand Movements. Symmetry, 12(4), p.541.

[171] Toledo-Pérez, D.C., Rodríguez-Reséndiz, J., Gómez-Loenzo, R.A. and Jauregui-Correa, J.C., 2019. Support vector machine-based EMG signal classification techniques: A review. Applied Sciences, 9(20), p.4402.

[172] Multi-Segmentation Parallel CNN Model for Estimating Assembly Torque Using Surface Electromyography Signals

[173] Estimating Biomechanical Time-Series with 3 Wearable Sensors: A Systematic Review of Machine 4 Learning Techniques

[174] Eklund, J.M., Mobasser, F. and Hashtrudi-Zaad, K., 2004, September. Hand force estimation using fast orthogonal search. In The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (Vol. 1, pp. 695-698). IEEE.

[175] Gavin, H.P., 2019. The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems. Department of Civil and Environmental Engineering, Duke University http://people. duke. edu/ hpgavin/ce281/lm. pdf, pp.1-19.

[176] Hagan, M.T. and Menhaj, M.B., 1994. Training feedforward networks with the Marquardt algorithm. IEEE transactions on Neural Networks, 5(6), pp.989-993.

[177] Lin, S.W., Chen, S.C., Wu, W.J. and Chen, C.H., 2009. Parameter determination and feature selection for back-propagation network by particle swarm optimization. Knowledge and Information Systems, 21(2), pp.249-266.

[178] Farid, N., ELBagoury, B.M., Roushdy, M. and Salem, A.B.M., Utilization of Levenberg-Marquardt based Neural Network Classifier in EMG signal Classification.

[179] Svensson, P., Antfolk, C., Malešević, N. and Sebelius, F., 2020. Characterization of pneumatic touch sensors for a prosthetic hand. IEEE Sensors Journal, 20(16), pp.9518-9527.

[180] Chalmers, G.R., 2008. Can fast-twitch muscle fibres be selectively recruited during lengthening contractions? Review and applications to sport movements. Sports Biomechanics, 7(1), pp.137-157.

[181] Yepes, J.C., Saldarriaga, A.J., Montoya-Leal, V., Orozco-Duque, A., Perez, V.Z. and BETANCUR, M.J., 2018, May. Classification Of Muscular Strength During Palmar Grasp Exercises Using Surface EMG Signals. In 2018 IX International Seminar of Biomedical Engineering (SIB) (pp. 1-7). IEEE.

[182] Li, Q.L., Song, Y. and Hou, Z.G., 2015. Estimation of lower limb periodic motions from sEMG using least squares support vector regression. Neural Processing Letters, 41(3), pp.371-388.

[183] Raikova, R., Aladjov, H., Celichowski, J. and Krutki, P., 2013. An approach for simulation of the muscle force modeling it by summation of motor unit contraction forces. Computational and Mathematical Methods in Medicine, 2013.

[184] Yousif, H.A., Zakaria, A., Rahim, N.A., Salleh, A.F.B., Mahmood, M., Alfarhan, K.A., Kamarudin, L.M., Mamduh, S.M., Hasan, A.M. and Hussain, M.K., 2019, November. Assessment of Muscles Fatigue Based on Surface EMG Signals Using Machine Learning and Statistical Approaches: A Review. In IOP Conference Series: Materials Science and Engineering (Vol. 705, No. 1, p. 012010). IOP Publishing.

[185] Roldán Jiménez, C., Bennett, P., Ortiz García, A. and Cuesta Vargas, A.I., 2019. Fatigue detection during sit-to-stand test based on surface electromyography and acceleration: a case study. Sensors, 19(19), p.4202.

[186] Gerdle, B., Henriksson-Larsen, K., Lorentzon, R. and Wretling, M.L., 1991. Dependence of the mean power frequency of the electromyogram on muscle force and fibre type. Acta Physiologica Scandinavica, 142(4), pp.457-465.

[187] Cao, H., Sun, S. and Zhang, K., 2017. Modified EMG-based handgrip force prediction using extreme learning machine. Soft computing, 21(2), pp.491-500.

[188] Buehler, W.J., Gilfrich, J.V. and Wiley, R.C., 1963. Effect of low-temperature phase changes on the mechanical properties of alloys near composition TiNi. Journal of applied physics, 34(5), pp.1475-1477.

[189] Kheirikhah, M.M., Rabiee, S. and Edalat, M.E., 2010, June. A review of shape memory alloy actuators in robotics. In Robot Soccer World Cup (pp. 206-217). Springer, Berlin, Heidelberg.

[190] Marfia, S. and Rizzoni, R., 2013. One-dimensional constitutive SMA model with two martensite variants: Analytical and numerical solutions. European Journal of Mechanics-A/Solids, 40, pp.166-185.

[191] Laurentis, K.J.D. and Mavroidis, C., 2002. Mechanical design of a shape memory alloy actuated prosthetic hand. Technology and Health Care, 10(2), pp.91-106.

[192] Iasnii, V. (2018), 'A damper device based on shape memory alloy wires' [PowerPoint presentation]. 8th International Engineering Week. Available at: $https://iew-digital.de/fileadmin/user_upload/SMA_based_damper_device.pdf$ (Accessed: 05 July 2020).

[193] Frenzel, J., Wieczorek, A., Opahle, I., Maaß, B., Drautz, R. and Eggeler, G., 2015. On the effect of alloy composition on martensite start temperatures and latent heats in Ni–Ti-based shape memory alloys. Acta Materialia, 90, pp.213-231.

[194] González-Sanz, G., Galé-Lamuela, D., Escolano-Margarit, D. and Benavent-Climent, A., 2019. Hysteretic Behavior and Ultimate Energy Dissipation Capacity of Large Diameter Bars Made of Shape Memory Alloys under Seismic Loadings. Metals, 9(10), p.1099.

[195] Lagoudas, D.C. ed., 2008. Shape memory alloys: modeling and engineering applications. Springer Science & Business Media.

[196] Šittner, P., Heller, L., Pilch, J., Curfs, C., Alonso, T. and Favier, D., 2014. Young's modulus of austenite and martensite phases in superelastic NiTi wires. Journal of materials engineering and performance, 23(7), pp.2303-2314.

[197] Rao, A., Srinivasa, A.R. and Reddy, J.N., 2015. Design of shape memory alloy (SMA) actuators (Vol. 3). Berlin: Springer International Publishing.

[198] Lagoudas, D.C. and Ding, Z., 1995. Modeling of thermoelectric heat transfer in shape memory alloy actuators: Transient and multiple cycle solutions. International Journal of Engineering Science, 33(15), pp.2345-2364.

[199] Diaconescu, E., 2008. The use of NARX neural networks to predict chaotic time series. Wseas Transactions on computer research, 3(3), pp.182-191.

[200] Simulink, M., 2005. Simulation and model-based design.

[201] Liang, C. and Rogers, C.A., 1997. One-dimensional thermomechanical constitutive relations for shape memory materials. Journal of intelligent material systems and structures, 8(4), pp.285-302.

[202] Dutta, S.M. and Ghorbel, F.H., 2005. Differential hysteresis modeling of a shape memory alloy wire actuator. IEEE/ASME Transactions on Mechatronics, 10(2), pp.189-197.

[203] Terriault, P. and Brailovski, V., 2011. Modeling of shape memory alloy actuators using Likhachev's formulation. Journal of Intelligent Material Systems and Structures, 22(4), pp.353-368.

[204] Filip, P. and Mazanec, K., 1994. Influence of work hardening on the reactive stress in a TiNi shape memory alloy. Materials Science and Engineering: A, 174(2), pp.L41-L43.

[205] Gallardo Fuentes, J.M., Gümpel, P. and Strittmatter, J., 2002. Phase change behavior of nitinol shape memory alloys. Advanced engineering materials, 4(7), pp.437-452.

[206] Yu, H. and Young, M.L., 2017. Three-dimensional modeling for deformation of austenitic NiTi shape memory alloys under high strain rate. Smart Materials and Structures, 27(1), p.015031.

[207] Alazzawi, S. and Filip, P., 2019. Modeling the transient behavior of the NiTi shape memory actuator using finite element analysis: Parametric study of the rate effects. Results in Materials, 1, p.100015.A

[208] Song, H., Kubica, E. and Gorbet, R., 2011, November. Resistance modelling of SMA wire actuators. In International workshop on smart materials, structures & NDT in aerospace conference.

[209] Tanaka, K., 1986. A thermomechanical sketch of shape memory effect: one-dimensional tensile behavior.

[210] Tanaka, K., Kobayashi, S. and Sato, Y., 1986. Thermomechanics of transformation pseudoelasticity and shape memory effect in alloys. International Journal of Plasticity, 2(1), pp.59-72.

[211] Boyd, J.G. and Lagoudas, D.C., 1994. Thermomechanical response of shape memory composites. Journal of intelligent material systems and structures, 5(3), pp.333-346.

[212] Machado, L.G., 2007. Shape memory alloys for vibration isolation and damping. Texas A&M University.

[213] Simo, J.C. and Hughes, T.J., 2006. Computational inelasticity (Vol. 7). Springer Science & Business Media.

[214] Alsayed, M., Bassiuny, A.M. and Bakr, E.M., 2010, December. Identification of Shape Memory Alloy Smart Actuator Using Nonlinear Autoregressive (NARX) Model. In Fifth Assiut University International Conference On Mechanical Engineering Advanced Technology For Industrial Production, Assiut, Egypt.

[215] Nicholson, D.E., 2011. Thermomechanical behavior of high-temperature shape memory alloy Ni-Ti-Pd-Pt actuators.

[216] Gustus, A., Stillfried, G., Visser, J., Jörntell, H. and van der Smagt, P., 2012. Human hand modelling: kinematics, dynamics, applications. Biological cybernetics, 106(11-12), pp.741-755.

[217] Arimoto, S., 2008. Control Theory of Multi-fingered Hands: A Modelling and Analytical–Mechanics Approach for Dexterity and Intelligence. Springer Science & Business Media.

[218] Flores, P. and Lankarani, H.M., 2016. Contact force models for multibody dynamics (Vol. 226, pp. 28-51). Cham: Springer.

[219] Ashmore, K., Cialdella, S., Giuffrida, A., Kon, E., Marcacci, M. and Di Matteo, B., 2019. ArtiFacts: Gottfried "Götz" von Berlichingen—The "Iron Hand" of the Renaissance. Clinical Orthopaedics and Related Research®, 477(9), pp.2002-2004.

[220] Kochan, A., 2005. Shadow delivers first hand. Industrial robot: an international journal.

[221] Kargov, A., Pylatiuk, C., Martin, J., Schulz, S. and Döderlein, L., 2004. A comparison of the grip force distribution in natural hands and in prosthetic hands. Disability and Rehabilitation, 26(12), pp.705-711.

[222] Piazza, C., Grioli, G., Catalano, M.G. and Bicchi, A., 2019. A century of robotic hands. Annual Review of Control, Robotics, and Autonomous Systems, 2, pp.1-32.

[223] Zollo, L., Roccella, S., Guglielmelli, E., Carrozza, M.C. and Dario, P., 2007. Biomechatronic design and control of an anthropomorphic artificial hand for prosthetic and robotic applications. IEEE/ASME Transactions On Mechatronics, 12(4), pp.418-429.

[224] Cho, K.J., Rosmarin, J. and Asada, H., 2007, April. SBC hand: a lightweight robotic hand with an SMA actuator array implementing C-segmentation. In Proceedings 2007 IEEE International Conference on Robotics and Automation (pp. 921-926). IEEE.

[225] Xu, Z. and Todorov, E., 2016, May. Design of a highly biomimetic anthropomorphic robotic hand towards artificial limb regeneration. In 2016 IEEE International Conference on Robotics and Automation (ICRA) (pp. 3485-3492). IEEE.

[226] ROBOTICS, Your Guide To The World Of Robotics, Available at: $https://robots.ieee.org/robots/shadow/$. (Accessed: 03 January 2020).

[227] ROBOTICS & AUTOMATION, Available at: $https://roboticsandautomationnews.com/2019/09/04/researchers-develop-robot-fingertips-for-grasping-and-releasing-objects-in-dry-and-wet-conditions/25354/$. (Accessed: 03 January 2020).

[228] Atasoy, A., Kaya, E., Toptas, E., Kuchimov, S., Kaplanoglu, E. and Ozkan, M., 2016, August. 24 DOF EMG controlled hybrid actuated prosthetic hand. In 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 5059-5062). IEEE.

[229] Baril, M., Laliberte, T., Gosselin, C. and Routhier, F., 2013. On the design of a mechanically programmable underactuated anthropomorphic prosthetic gripper. Journal of Mechanical Design, 135(12).

[230] Ten Kate, J., Smit, G. and Breedveld, P., 2017. 3D-printed upper limb prostheses: a review. Disability and Rehabilitation: Assistive Technology, 12(3), pp.300-314.

[231] Andrianesis, K. and Tzes, A., 2015. Development and control of a multifunctional prosthetic hand with shape memory alloy actuators. Journal of Intelligent & Robotic Systems, 78(2), pp.257-289.

[232] Price, A.D., Jnifene, A. and Naguib, H.E., 2007. Design and control of a shape memory alloy based dexterous robot hand. Smart Materials and Structures, 16(4), p.1401.

[233] Xu, Z., Todorov, E., Dellon, B. and Matsuoka, Y., 2011, May. Design and analysis of an artificial finger joint for anthropomorphic robotic hands. In 2011 IEEE International Conference on Robotics and Automation (pp. 5096-5102). IEEE.

[234] Parshotam, D., S., 2020. Linear Shape Memory Alloy (Nitinol) Actuators in a 3D Printed Hand Prosthesis. Master's Theses in Engineering Sciences.

[235] App Store, 2020. Anatomy Learning - 3D Anatomy, Available at: $< https://apps.apple.com/us/app/anatomy-learning-3d-anatomy/id1528713417 >$ (Accessed: 11 January 2020).

[236] Warnecke, D., Meßemer, M., de Roy, L., Stein, S., Gentilini, C., Walker, R., Skaer, N., Ignatius, A. and Dürselen, L., 2019. Articular cartilage and meniscus reveal higher friction in swing phase than in stance phase under dynamic gait conditions. Scientific reports, 9(1), pp.1-9.

[237] Rapoport, S., Mizrahi, J., Kimmel, E., Verbitsky, O. and Isakov, E., 2003. Constant and variable stiffness and damping of the leg joints in human hopping. J. Biomech. Eng., 125(4), pp.507-514.

[238] Oatis, C.A., 1993. The use of a mechanical model to describe the stiffness and damping characteristics of the knee joint in healthy adults. Physical Therapy, 73(11), pp.740-749.

[239] Knez, L., Slavič, J. and Boltežar, M., 2017. A sequential approach to the biodynamic modeling of a human finger. Shock and Vibration, 2017.

[240] Wright, V. and Johns, R.J., 1961. Quantitative and qualitative analysis of joint stiffness in normal subjects and in patients with connective tissue diseases. Annals of the rheumatic diseases, 20(1), p.36.

[241] Milner, T.E. and Franklin, D.W., 1998. Characterization of multijoint finger stiffness: dependence on finger posture and force direction. IEEE Transactions on Biomedical engineering, 45(11), pp.1363-1375.

[242] Chen Chen, F., Appendino, S., Battezzato, A., Favetto, A., Mousavi, M. and Pescarmona, F., 2013. Constraint study for a hand exoskeleton: human hand kinematics and dynamics. Journal of Robotics, 2013.

[243] Sancho-Bru, J.L., Perez-Gonzalez, A., Vergara-Monedero, M. and Giurintano, D., 2001. A 3-D dynamic model of human finger for studying free movements. Journal of biomechanics, 34(11), pp.1491-1500.

[244] Paredes-Madrid, L., Palacio, C.A., Matute, A. and Parra Vargas, C.A., 2017. Underlying physics of conductive polymer composites and force sensing resistors (FSRs) under static loading conditions. Sensors, 17(9), p.2108.

[245] Yang, D.P., Zhao, J.D., Gu, Y.K., Wang, X.Q., Li, N., Jiang, L., Liu, H., Huang, H. and Zhao, D.W., 2009. An anthropomorphic robot hand developed based on underactuated mechanism and controlled by EMG signals. Journal of Bionic Engineering, 6(3), pp.255-263.

[246] Ko, J., 2011. Fuzzy PWM-PID control and shape memory alloy actuator for cocontracting antagonistic muscle pairs in an artificial finger (Doctoral dissertation).

[247] Gilardi, G., Haslam, E., Bundhoo, V. and Park, E.J., 2010. A shape memory alloy based tendon-driven actuation system for biomimetic artificial fingers, part II: modelling and control. Robotica, 28(5), pp.675-687.

[248] Bandara, D.S.V., Gopura, R.A.R.C., Kajanthan, G., Brunthavan, M. and Abeynayake, H.I.M.M., 2014, June. An under-actuated mechanism for a robotic finger. In The 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent (pp. 407-412). IEEE.

[249] Armstrong, T.J. and Chaffin, D.B., 1978. An investigation of the relationship between displacements of the finger and wrist joints and the extrinsic finger flexor tendons. Journal of Biomechanics, 11(3), pp.119-128.

[250] Kamper, D.G., Cruz, E.G. and Siegel, M.P., 2003. Stereotypical fingertip trajectories during grasp. Journal of neurophysiology, 90(6), pp.3702-3710.

[251] Cordella, F., Zollo, L., Guglielmelli, E. and Siciliano, B., 2012. A bio-inspired grasp optimization algorithm for an anthropomorphic robotic hand. International Journal on Interactive Design and Manufacturing (IJIDeM), 6(2), pp.113-122.

[252] Chandra, A., Chandna, P. and Deswal, S., 2011. Analysis of hand anthropometric dimensions of male industrial workers of Haryana state. International Journal of Engineering (IJE), 5(3), pp.242-256.

[253] Hume, M.C., Gellman, H., McKellop, H. and Brumfield, R.H., 1990. Functional range of motion of the joints of the hand. Journal of hand surgery, 15(2), pp.240-243.

[254] Howard, R.D., 1990. Joint and actuator design for enhanced stability in robotic force control (Doctoral dissertation, Massachusetts Institute of Technology).

[255] Saadeh, M.Y., Carambat, T.D. and Arrieta, A.M., 2017, September. Evaluating and modeling force sensing resistors for low force applications. In Smart Materials, Adaptive Structures and Intelligent Systems (Vol. 58264, p. V002T03A001). American Society of Mechanical Engineers.

[256] Fu, Q. and Santello, M., 2018. Improving fine control of grasping force during hand–object interactions for a soft synergy-inspired myoelectric prosthetic hand. Frontiers in Neurorobotics, 11, p.71.

[257] Anderson, D.H., 1987. Mathematical modelling in biomedicine: optimal control of biomedical systems.: Y. Cherruault, D. Reidel, Dordrecht, Holland, 1986, xviii+ 258 pp., $49.50, Dfl 130.00,£ 36.25.

[258] Morari, M. and Gentilini, A., 2001. Challenges and opportunities in process control: Biomedical processes. American Institute of Chemical Engineers. AIChE Journal, 47(10), p.2140.

[259] Banks, H.T., 2013. Modeling and control in the biomedical sciences (Vol. 6). Springer Science & Business Media.

[260] Swan, G.W., 1981. Optimal control applications in biomedical engineering—a survey. Optimal Control Applications and Methods, 2(4), pp.311-334.

[261] Hajizadeh, I., Rashid, M., Sevil, M., Brandt, R., Samadi, S., Hobbs, N. and Cinar, A., 2018. Adaptive model predictive control for nonlinearity in biomedical applications. IFAC-PapersOnLine, 51(20), pp.368-373

[262] Houk, J.C., 1988. Control strategies in physiological systems. The FASEB journal, 2(2), pp.97-107.

[263] Lassoued, A. and Boubaker, O., 2020. Modeling and control in physiology. In Control Theory in Biomedical Engineering (pp. 3-42). Academic Press.

[264] Del Giudice, M., 2015. Self-regulation in an evolutionary perspective. In Handbook of biobehavioral approaches to self-regulation (pp. 25-41). Springer, New York, NY.

[265] Houk, J.C. and Rymer, W.Z., 2011. Neural control of muscle length and tension. Comprehensive Physiology, pp.257-323.

[266] Chappell, P., 2016. Mechatronic Hands Prosthetic and Robotic Design (Vol. 105). Institution of Engineering and Technology.

[267] Jazar, R.N., 2010. Theory of applied robotics: kinematics, dynamics, and control. Springer Science & Business Media.

[268] Jamshidi, M. ed., 2017. Systems of systems engineering: principles and applications. CRC press.

[269] Jamshidi, M. and Malek-Zavarei, M., 1986. Linear control systems; a computer-aided approach (No. U10 J31). Pergamon

[270] Åström, K.J., Hägglund, T. and Astrom, K.J., 2006. Advanced PID control (Vol. 461). Research Triangle Park: ISA-The Instrumentation, Systems, and Automation Society.

[271] Huang, A.C. and Chien, M.C., 2010. Adaptive control of robot manipulators: a unified regressor-free approach. World scientific.

[272] Zollo, L., Roccella, S., Guglielmelli, E., Carrozza, M.C. and Dario, P., 2007. Biomechatronic design and control of an anthropomorphic artificial hand for prosthetic and robotic applications. IEEE/ASME Transactions On Mechatronics, 12(4), pp.418-429.

[273] Rodriguez-Cheu, L.E. and Casals, A., 2006, February. Sensing and control of a prosthetic hand with myoelectric feedback. In The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. (pp. 607-612). IEEE.

[274] Zhao, J., Xie, Z., Jiang, L., Cai, H., Liu, H. and Hirzinger, G., 2006, February. A five-fingered underactuated prosthetic hand control scheme. In The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. (pp. 995-1000). IEEE.

[275] Kwek, L.C., Wong, E.K., Loo, C.K. and Rao, M.V.C., 2003. Application of active force control and iterative learning in a 5-link biped robot. Journal of Intelligent and Robotic Systems, 37(2), pp.143-162.

[276] Tzafestas, S., Raibert, M. and Tzafestas, C., 1996. Robust sliding-mode control applied to a 5-link biped robot. Journal of Intelligent and Robotic Systems, 15(1), pp.67-133.

[277] Hewit, J.R. and Burdess, J.S., 1981. Fast dynamic decoupled control for robotics, using active force control. Mechanism and Machine Theory, 16(5), pp.535-542.

[278] Arimoto, S., 2008. Control Theory of Multi-fingered Hands: A Modelling and Analytical–Mechanics Approach for Dexterity and Intelligence. Springer Science & Business Media.

[279] Yuden, M.A.M., Ghazaly, M.M., Amran, A.C., Jamaludin, I.W., Yee, K.H., Yaacob, M.R., Abdullah, Z. and Kiat, Y.C., 2017. POSITIONING CONTROL PERFORMANCES OF A ROBOTIC HAND SYSTEM. Jurnal Teknologi, 79(1).

[280] Shauri, R.L.A., Salleh, N.M. and Hadi, A.K.A., 2014, November. PID position control of 7-DOF three-fingered robotic hand for grasping task. In 2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014) (pp. 70-74). IEEE.

[281] Cirstea, M., Dinu, A., McCormick, M. and Khor, J.G., 2002. Neural and fuzzy logic control of drives and power systems. Elsevier.

[282] Sets, F. and Zadeh, L.A., 1965. Inform. Control, 8, pp.338-353.

[283] Oviedo, J.J.E., Vandewalle, J.P. and Wertz, V., 2006. Fuzzy logic, identification and predictive control. Springer Science & Business Media.

[284] Zadeh, L.A., 1973. Outline of a new approach to the analysis of complex systems and decision processes. IEEE Transactions on systems, Man, and Cybernetics, (1), pp.28-44.

[285] Boubaker, O., 2020. Medical robotics. Control Theory in Biomedical Engineering, pp.153-204.

[286] Lam, H.K., 2016. Polynomial fuzzy model-based control systems. Stability Analysis and Control Synthesis Using Membership Function-Dependent Techniques. Cham: Springer-Verlag, 307.

[287] Ajiboye, A.B. and Weir, R.F., 2005. A heuristic fuzzy logic approach to EMG pattern recognition for multifunctional prosthesis control. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 13(3), pp.280-291.

[288] Driankov, D., Hellendoorn, H. and Reinfrank, M., 2013. An introduction to fuzzy control. Springer Science & Business Media.

[289] McNeill, F.M. and Thro, E., 2014. Fuzzy logic: a practical approach. Academic Press.

[290] Englehart, K., Hudgin, B. and Parker, P.A., 2001. A wavelet-based continuous classification scheme for multifunction myoelectric control. IEEE Transactions on Biomedical Engineering, 48(3), pp.302-311.

[291] Englehart, K. and Hudgins, B., 2003. A robust, real-time control scheme for multifunction myoelectric control. IEEE transactions on biomedical engineering, 50(7), pp.848-854.

[292] Alassar, A.Z., Abuhadrous, I.M. and Elaydi, H.A., 2010, March. Comparison between FLC and PID Controller for 5DOF robot arm. In 2010 2nd International Conference on Advanced Computer Control (Vol. 5, pp. 277-281). IEEE.

[293] Bowers, D.L. and Lumia, R., 2003. Manipulation of unmodeled objects using intelligent grasping schemes. IEEE Transactions on Fuzzy Systems, 11(3), pp.320-330.

[294] Bose, B.K., 1994. Expert system, fuzzy logic, and neural network applications in power electronics and motion control. Proceedings of the IEEE, 82(8), pp.1303-1323.

[295] Mahfouf, M., Abbod, M.F. and Linkens, D.A., 2001. A survey of fuzzy logic monitoring and control utilisation in medicine. Artificial intelligence in medicine, 21(1-3), pp.27-42.

[296] Ekemezie, P.N. and Osuagwu, C.C., 2000. Effect of Varying Controller Parameters on the Performance of a Fuzzy Logic Control System. Nigerian Journal of Technology, 19(1), pp.40-51.

[297] Lee, K.W. and Choi, H., 2010. Analysis of FLC with changing fuzzy variables in frequency domain. International Journal of Control, Automation and Systems, 8(3), pp.695-701.

[298] Feigenbaum, E.A., 1977. The art of artificial intelligence. 1. Themes and case studies of knowledge engineering. Stanford Univ CA Dept of Computer Science.

[299] Michie, D., 1991. Expert systems: The end of the beginning. AI & society, 5(2), pp.142-147.

[300] Jiang, Y., Yang, C. and Ma, H., 2016. A review of fuzzy logic and neural network based intelligent control design for discrete-time systems. Discrete Dynamics in Nature and Society, 2016.

[301] Mozhanova, M., 2012. Design of a High–Resolution Surface Electromyogram (EMG) Conditioning Circuit. Undergraduate. Worchester Polytechnic Institute.

Appendices

A.1 Appendix A

A.1.1 Ethical clearance

The formality of acquiring ethical clearance is to ensure responsibility and accountability for the conduced research. It also minimizes the risk of infringing on the rights, dignity and safety of the participants. With the ultimate purpose of the research leading to favourable outcomes, ethical clearance strengthens the legitimacy of the research results.

A.1.1.1 Purpose of ethical clearance

This dissertation involves the use of sEMG electrodes placed on the belly of 6 target forearm muscles, 3 flexors and 3 extensors, recording muscle activity during various grasps. Additionally a pressure sensitive glove is used to gather information on the grip force produced by the hand. Owing to the fact that the participants are students over the age of 18, approval from the registrar is required additional to ethical clearance. Ethical clearance ensures a correct and ethical methodology and protocol, guaranteeing anonymity and the presentation of results to the participants. In conjunction with the Physiology Faculty and Movement Lab, research was conducted on five participants, with the subsequent data collected being used for the purpose of training classifier and regressor models. Another reason imperative for ethical clearance is located in the potential to publish the results, which may only be realised with ethical clearance evidence.

A.1.1.2 Scope and ethical clearance procedure

The following points outline the scope, procedure and research boundaries granted by the ethical clearance. No risk is anticipated with this study and ethical clearance is granted unconditionally.

- This study will take place in the Movement Lab, situated in the Khanya block on Education Campus. This is due to the EMG equipment being situated there and thus the testing cannot be relocated.

- The expected duration of each procedure (full duration), is roughly 1 hour excluding questionnaire and evaluation. This will cover the attachment of the EMG electrodes as well as the force sensitive glove and data collection.

- For proper collection of EMG data, it is required that the participant's dominant forearm be cleanly shaved the day before testing. This guarantees no inter participant bias as a result of body hair.

- When placing the EMG sensors on the participant's dominant forearm, it is important to locate the target muscles as each person's physiology is different. To do this a combination of palpating (feeling) and measuring will take place on the participant's forearm.

- Once the locations of the muscles are known, the EMG electrodes will be attached and secured using elasto-plast tape, making sure the electrode firmly contacts the participant's skin.

- The force sensitive glove is given to the participant to put on and thereafter the placement of the force sensitive resistors is verified to ensure activation during gripping. The setup of the EMG electrodes and force glove is estimated to take 15 minutes.

- After setup, a 10 second calibration of the force glove will take place at which time the participant's hand will remain in a rest position.

- Once calibration is completed, one of 6 grasps will be demonstrated to the participant by the principle investigator on the 3D printed grasp apparatus provided. Once instructed to start the movement, the participant needs to wait 2 seconds before grasping, 6 seconds while grasping and 2 seconds after grasping .

- The demonstration of the grasp, by the principle investigator together with the data collection amounting to 10 seconds will be repeated for a total of 6 grasps and rest. The process of collecting the grasp data will then be repeated for the following 45 minutes.

- The target muscles investigated are: flexor digitorum superficialis, flexor carpi ulnaris, flexor pollicis longus, extensor digitorum, extensor carpi radialis longus and extensor pollicis longus.

- The target grasps investigated are: 3 Jaw chuck pinch, cylindrical grasp, hook grasp, lateral pinch, spherical grasp, tip pinch and rest.

- Participating in this study is completely voluntary and will not affect the participant in any way if they refuse to partake. No remuneration will be given and participation will bear no financial cost to the participant.

- At any point during the study if the participant wishes to withdraw, they are entitled to do so with no consequences.

- The principle investigator and supervisor may ask the participant to stop at any time during the procedure if they feel the participant is at risk or fails to follow the procedure outlined.

- The participant's data will only be made available to the principle investigator and supervisor,and may be used anonymously in publications.

- The study is open to students, male and female, over the age of 18, with no serious hand impairments. There is no preference of the participant's dominant hand.

A.1.1.3 Approved ethical clearance

Shown in Figure A.1.1 is the approved ethical clearance certificate (M190766). Following the acquisition of the ethical clearance certificate, data was collected for a period of one week from five participants.

UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG

R14/49 Mr J-P Hynek

HUMAN RESEARCH ETHICS COMMITTEE (MEDICAL)
CLEARANCE CERTIFICATE NO. M190766

NAME: Mr J-P Hynek
(Principal Investigator)
DEPARTMENT: School of Mechanical, Industrial and Aeronautical Engineering University

PROJECT TITLE: Real-time fuzzy logic control of a smart material actuated, robotic prosthesis hand with grip compensation

DATE CONSIDERED: Ad hoc

DECISION: Approved unconditionally

CONDITIONS:

SUPERVISOR: Professor J Pedro

APPROVED BY: Dr CB Penny, Chairperson, HREC (Medical)

DATE OF APPROVAL: 2019/08/27

This clearance certificate is valid for 5 years from date of approval. Extension may be applied for.

DECLARATION OF INVESTIGATORS

To be completed in duplicate and **ONE COPY** returned to the Research Office Secretary on the 3rd Floor, Phillip Tobias Building, Parktown, University of the Witwatersrand, Johannesburg.
I/we fully understand the conditions under which I am/we are authorized to carry out the above-mentioned research and I/we undertake to ensure compliance with these conditions. Should any departure be contemplated, from the research protocol as approved, I/we undertake to submit details to the Committee. **I agree to submit a yearly progress report**. When a funder requires annual re-certification, the application date will be one year after the date when the study was initially reviewed. In this case, the study was initially reviewed in **July** and will therefore reports and re-certification will be due early in the month of **July** each year. Unreported changes to the application may invalidate the clearance given by the HREC (Medical).

Principal Investigator Signature

2019/08/27
Date

PLEASE QUOTE THE CLEARANCE CERTIFICATE NUMBER IN ALL ENQUIRIES

Figure A.1.1: Ethics Approval

Using students as participants meant that registrar approval was required, as shown in Figure A.1.2. Prior to data collection, the simulation framework was setup, research conducted, relevant apparatus constructed, initial training and development of the EMG and EMG-force control system accomplished using dummy variables and tasks not directly requiring ethical clearance. This simplified data processing.

UNIVERSITY OF THE
WITWATERSRAND,
JOHANNESBURG

OFFICE OF THE DEPUTY REGISTRAR

30 August 2019

Juan-Paul Hynek
Student number 754214
MSc Mechanical Engineering
School of MIA Engineering

TO WHOM IT MAY CONCERN

"Real-time fuzzy logic control of a smart material actuated, robotic prosthesis hand with grip compensation"

This letter serves to confirm that the above project has received permission to be conducted on University premises, and/or involving staff and/or students of the University as research participants. In undertaking this research, you agree to abide by all University regulations for conducting research on campus and to respect participants' rights to withdraw from participation at any time.

If you are conducting research on certain student cohorts, year groups or courses within specific Schools and within the teaching term, permission must be sought from Heads of School or individual academics.

Ethical clearance has been obtained. (Protocol number: M190766)

Carol Crosley
University Registrar

Private Bag 3, Wits, 2050, South Africa | T +27 11 717 1204/8 | F +27 86 553 2271 | www.wits.ac.za

Figure A.1.2: Registrar Approval

B.2 Appendix B

B.2.1 Single-channel electromyographic sensor development

Undoubtedly an integral part of a RPH is the sEMG sensor. This component fundamentally sets the tone for the sensitivity of a RPH by either providing good or bad quality data upon which, specific to this dissertation, classification and grip compensation is built. Also, it is established in literature as the component with the largest contributing cost to the development of a RPH [301]. This, in turn, renders current RPH solutions costly and infeasible, particularly in developing countries. It must be stated, however, that the low-cost sEMG sensor presented in this section was not developed as a solution to the aforementioned factors, but rather as an alternative in-case an sEMG device could not be procured. Having said this, the prototype sEMG sensor offers a lot of potential for further consideration in low-cost RPH solutions.

As a point of clarification, ethical clearance was not necessary for the development of the sEMG sensor prototype presented here since no tests were conducted on test subjects, but rather analysed by using a oscilloscope. The basic working principles and structure of a sEMG sensor touched on in Section 3.1.1 will now be further elaborated.

Starting with the input, circular Covident Ag/AgCl electrodes with press studs were used and subsequently reduced in diameter to ensure an inter spacing of 20mm. This inter spacing, previously discussed in relation to the Delsys Trigno sEMG sensors, has to do with both electrodes seeing similar muscle fibre densities. Furthermore, a bipolar electrode configuration was used as it removes noise signals common to both electrodes thereby improving the SNR of the sEMG circuit. Conversely, another sEMG electrode configuration rarely adopted is a monopolar configuration. As seen in Figure **??**, the sEMG sensor, the third electrode labelled *COM* accounts for the bipolar functionality. This electrode is typically placed outside the vicinity of the other two electrodes, away from motor points and on a bony landmark. Promising results were also shown for the case when the electrode was grounded with the sEMG circuit. In Figure **??**, the electrodes labelled L and R follow no particular order since the measurements conducted between them are differential, however, the same directional sentiment shared for the Delsys Trigno sensors remains consistent. The cables were braided to further minimize electro-magnetic noise created when in close proximity with one another. Having established the input, the next step focuses on the pre-amplification of the sEMG signal using an instrument amplifier. These types of amplifiers are laser trimmed to maintain precision and incorporate dual rails with good swing characteristics. A drawback of a instrument amplifier is that it requires a negative rail. The gain of the operational amplifier was set using a variable resistor such that a gain of 50 times the original signal was observed. Taking advice from publications such as [301], this gain was found to work adequately, and did not promote much growth of noise. To rid the signal from low-frequency artefacts it was filtered using a first order lowpass filter. The use of a variable resistor in this filter meant that the cutoff frequency could be tuned to the system. In this case, a lowpass cutoff frequency of 10Hz was acceptable.

A non-inverting operational amplifier did much of the amplification of the signal hereafter. A gain amounting to 100 was set using variable resistors and was found to produce an output voltage within the bounds of the operational amplifiers rails, ensuring no distortion took place. The total gain of the sEMG circuit was 5000 and it should be said that an inverting operational amplifier may be used to replace the nn-inverting operational amplifier. The amplification of high-frequency noise is a by-product of signal gain and was therefore filtered using a first order highpass filter with variable cutoff frequency. The highpass cutoff frequency was set to 400Hz in the same manner the lowpass cutoff frequency was tuned. Using filters presents a slight drop in the magnitude of the signal, thus as a corrective measure a unity gain operational amplifier follows.

Furthermore, the last step involved considers the input requirements of the ADC, which in the case of an Arduino Uno is a 10-bit ADC with voltage bounds of 0V to 5V. To accomplish this, it is first assumed that the sEMG signals acquired fall within the -2.5V to 2.5V range, and a tension adder is used. The tension adder, which is a operational amplifier configuration, is set to add 2.5V, which scales the output of the sEMG circuit to the 0V to 5V range accepted by the Arduino Uno.

Further research was conducted using a ADS1115 ADC with a higher resolution, but it was concluded that the Delta-Sigma framework capped the sampling speed at 860SPS. This sampling speed in reality is not achievable, resulting in the ADC not being suitable to capture sEMG signals.

From Figure B.2.1, it should be noted that more than one quad operational amplifier integrated circuit was incorporated in the sEMG circuit due to the change in supply voltages. Also, shown in Figure B.2.1, it the top view of the through-hole sEMG sensor. This was the first attempt at developing the sEMG sensor and followed the recommendations given by [301]. It was found to perform satisfactorily as it was able to amplify the signals appropriately. The methods of production for the developed sEMG sensor, included trace layout of the circuit board and soldering prototype board respectively.

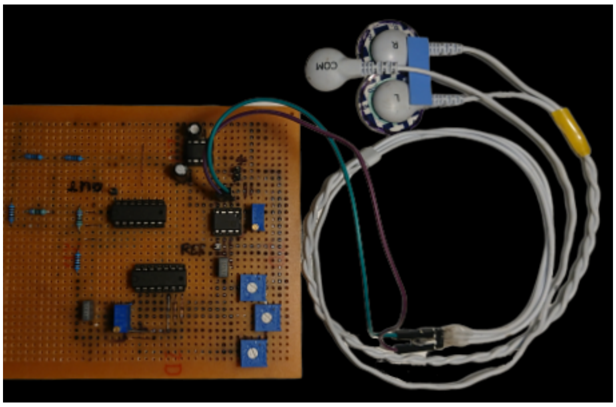


Figure B.2.1: Prototype single-channel sEMG sensor

In continuing with the presentation of the sEMG sensor shown in Figure **??**, the relevant components comprising the circuit and the hurdles experienced are now discussed. The sEMG sensor circuit shown in Figure B.2.2 may be broken down into sub-systems namely, power-supply, amplification, filtering and level-shifting. Taking a look at amplification first, the sEMG signals captured through male header-pins are relayed to the AD620ANZ instrument amplifier. Furthermore, a 10KΩ multi-turn variable resistor, set to 1KΩ, i.e., R4, was used to tune the gain of the amplifier to 49.91. An Arduino Uno supplied the necessary voltage required for the positive rail of the amplifier and was also used as a stepping stone in the creation of the negative voltage. Using a ICL7660 integrated circuit with decoupling capacitors, 5V was inverted to -5V. This output was subsequently used to supply the negative rail of the amplifier.

The lowpass filter consisted of a single-turn variable resistor in series with a polyester non-polar capacitor. From the equation governing cutoff frequency, it was calculated that a resistance of 72.34KΩ, i.e., R1, was required to yield a cutoff frequency of 10Hz. Also, it was calculated that the filter would introduce a phase shift of -45deg.

The second phase of amplification was done using a TLC274 quad operational amplifier. The resistances of single-turn variable resistors R2 and R3 were 50KΩ and 500Ω, with a ratio of 100, is equivalent to the gain of the amplifier. It should be noted that the quad amplifier, shown in Figure B.2.2, is depicted as being segmented, however, is one integrated circuit. Like the lowpass filter, the highpass filters was assigned a resistor value of 3.98KΩ, resulting in a cutoff frequency of 400Hz. The unity gain stage is trivial and not further explained.

On the other hand, the level shifting stage uses a potential divider and tension adder to centralize sEMG signals around a 2.5V axis. The fixed resistor values were carefully selected to obey configuration ratios and were of the 1% variety. There is a further crucial point to be considered. As seen in Figure B.2.2, the negative rail is supplied with ground, since the expected output must lie within the range 0V to 5V. This meant that a separate TLC274 amplifier had to be used. Depicted in Figure B.2.2 the header pin labelled *A* represents the connection to a Arduino Uno analog input pin.

A hurdle in creating the sEMG circuit was sourcing a quad operational amplifier with a bandwidth range large enough to register the frequency of the sEMG signals. It was found that the LM324 quad operational amplifier stated in [301] was inadequate in this department. Moreover, the data-sheet of the amplifier suggested a maximum bandwidth of 1000Hz, which is not achieved in reality due to very poor swing characteristics. After discarding this amplifier, a socket was soldered in its place and various quad operational amplifiers were tested, including TL074CN, TLC274 and TL084CN. It was concluded through viewing the output, that the TLC274 gave superior results with a much fast response time.

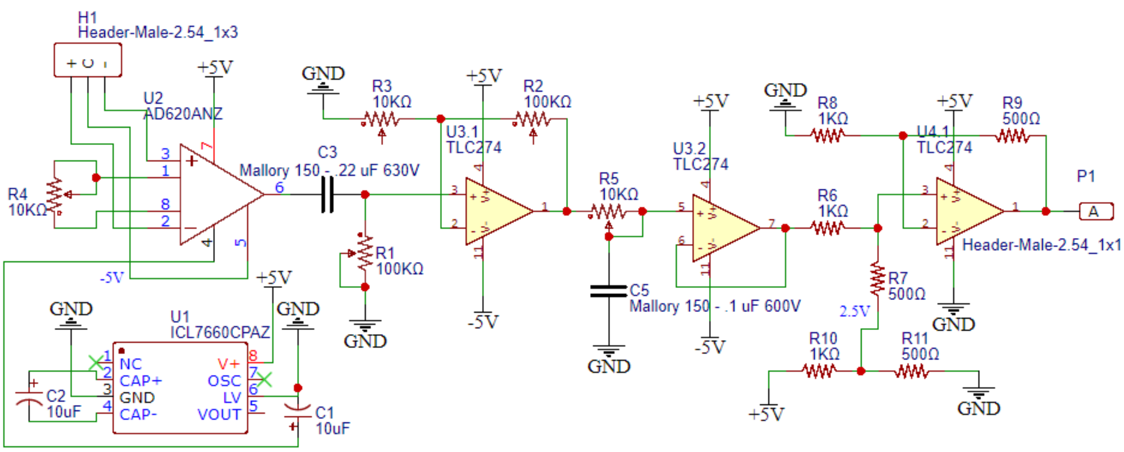


Figure B.2.2: Prototype single-channel sEMG sensor circuit

Performing a *analogRead* command with a baud rate of 115200 in the Arduino integrated development environment, resulted in the snapshot shown in Figure B.2.3. The results presented are for a random electrical disturbance, in the frequency range 0HZ to 500Hz, applied to the electrodes of the sEMG sensor using a signal generator. The 10-bit resolution of the Arduino Uno results in 1024 divisions in the input range, meaning that the minimum voltage is roughly 5mV.

Transitioning to the source sEMG signal is accomplished by dividing this minimum by the overall sEMG circuit gain, which results in a minimum sEMG signal increment of 1uV. Considering the raw sEMG signal shown in Figure **??**, the 1uV increment is sufficient for adequate resolution.

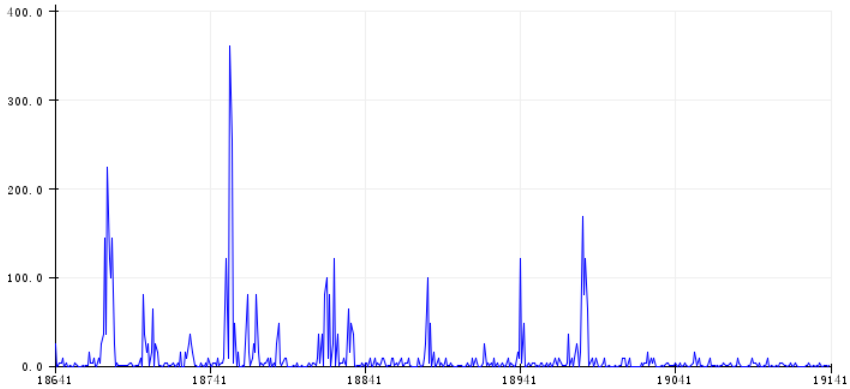


Figure B.2.3: Prototype sEMG sensor output

Table 11.1 details the bill of components and connection present in the sEMG circuit shown in Figure B.2.2. By replicating the circuit in software such as EasyEDA it is possible to generate mask or gerber files for the manufacturing of a smaller printed circuit board.

Table 11.1: Bill of components for prototype sEMG sensor

Label	Description	Value
R1	Variable single-turn 100KΩ resistor	72.34KΩ
R2	Variable single-turn 100KΩ resistor	72.34KΩ
R3	Variable single-turn 10KΩ resistor	72.34KΩ
R4	Variable multi-turn 10KΩ resistor	72.34KΩ
R5	Variable multi-turn 10KΩ resistor	72.34KΩ
R6	1% Resistor	1KΩ
R7	1% Resistor	500Ω
R8	1% Resistor	1KΩ
R9	1% Resistor	500Ω
R10	1% Resistor	1KΩ
R11	1% Resistor	500Ω
C1	Electrolytic capacitor	10uF
C2	Electrolytic capacitor	10uF
C3	Polyester capacitor	.22uF
C4	Polyester capacitor	.10uF
U1	ICL7660 CMOS switched-capacitor voltage converter	-
U2	AD620ANZ high accuracy instrumentation amplifier	-
U3	TLC274 quad single supply operational amplifier	-
U4	TLC274 quad single supply operational amplifier	-
+	Electrode 1	-
-	Elctrode 2	-
C	Common electrode	-
A	Arduino Uno analog pin	-

C.3 Appendix C

C.3.1 SMA Test Rig Setup

Figure C.3.1 shows the test rig developed for the data acquisition of input power and output displacement of the SMA actuator.

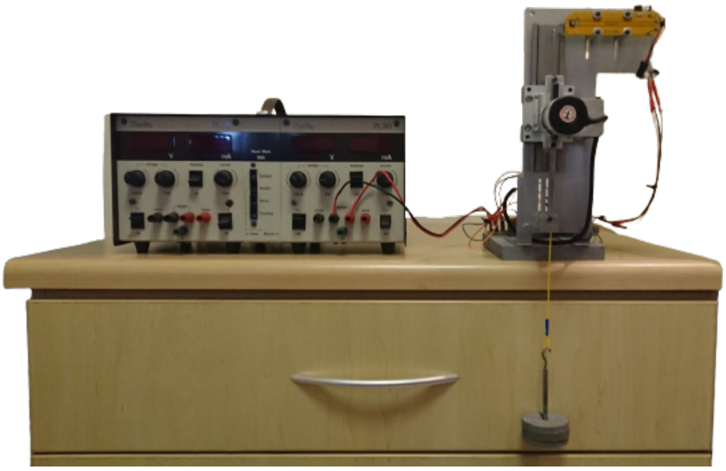


Figure C.3.1: SMA actuator test rig with power source, Nitinol actuator and optical encoder