An Intelligent Prosthetic Hand using Hybrid Actuation and Myoelectric Control

by

Beng Guey Lau

Submitted in accordance with the requirements for the degree of Doctor of Philosophy

The University of Leeds
School of Mechanical Engineering

October 2009

The candidate confirms that the work submitted is his/her own and that appropriate credit has been given where reference has been made to the work of others.

Abstract

This thesis details the design and development of an intelligent prosthetic hand based on hybrid DC and Shape Memory Alloy (SMA) actuation and controlled by only two myoelectric sensors.

A prosthesis as a tool makes no pretence of trying to replace the lost limb physiologically but it works as an aid to help provide some of the lost functions and is an interchangeable device worn and used as needed.

Much research has been carried out to develop artificial prosthetic hands with capabilities similar to the human hand. The human hand is a very complex grasping tool, that can handle objects of different size, weight and shape; however, they are far from providing its manipulation capabilities. This is for many different reasons, such as active bending is limited to two or three joints and user-unfriendliness. These limitations are present in commercial prosthetic hands, together with others always complained about by patients and amputees, such as inability to provide enough grasping functionality and heavy weight. Several robotic and anthropomorphic hands may have sufficient active degrees of freedom to allow dexterity comparable to that of the human hand. Unfortunately, they cannot be used as prostheses due to their physical characteristic that poses several serious limitations on human-hand interaction.

Hence, the motivation for this research is to investigate the use of a hybrid actuation mechanism in the design and development of an intelligent prosthetic hand. This work highlights user-friendliness and involves a proper mechanical design with more active degrees of freedom and incorporating an intelligent control system.

A system with a finger prototype is considered. Testing through simulation and physical models reveals a number of limitations. A hybrid actuation system, to increase the finger active degrees of freedom is therefore developed, with a mechanism consisting of DC and SMA actuators. Besides, only two myoelectrodes channels (enhancing the user-friendliness of the device) are used for the system control input signal.

Two novel features are developed in the new prosthetic hand. Firstly, its hybrid actuation mechanism has the advantage of increasing the active degrees of freedom; secondly, using only two myoelectric sensors has potential for controlling more than three patterns of fingers movements.

By using artificial neural network patterns classification technique, three and five patterns of wrist joint movement corresponding to finger movement can be recognised as more than 85% correct and furthermore, seven as 70% correct.

Acknowledgements

I'd like to thank a number of people each one of whom has helped me with his own way through my PhD. I couldn't have done it without you.

First of all I'd like to thank Mum, Dad, and the rest of my family for their love and support throughout my four years of studies away from them. This one is for you!

A special mention to my supervisors Dr. Abbas Dehghani and Dr. Robert. C. Richardson for their support and concern and for giving me the opportunity to present my work in a international conferences. It has been an honour studying and working by your side all those years. Thank you Abbas and Robert.

A special thank you to Dr. David Gow, who is from Smart Centre, NHS Lothian, for the encouragement, patience and kind support. Thank you David.

Thanks to the staff in the School of Mechanical Engineering that have helped me through my time there. In particular Mr. David Readman and Mr. Steve Caddick have been a great source of technical support and hoarded parts; thanks for all the advice, help and teasing jokes! You have made my time in the labs fun and worthwhile!

Contents

List of Abbreviations

ANFIS	Adaptive Neuro-Fuzzy Inference System
ANN	Artificial Neural Network
AR	Autoregressive Coefficients
ARX	Autoregressive with Exogenous Terms
ARMAX	Autoregressive Moving Average with Exogenous Terms
CWT	Continuous Wavelet Transform
DIP	Distal Interphalangeal
DOF	Degree of Freedom
DP	Distal Phalange
DWT	Discrete Wavelet Transform
EAPs	Electroactive Polymers
EMG	Electromyograph
FFTs	Fast Fourier Transforms
FL	Fuzzy Logic
IMES	Implantable Myoelectric Sensor
logsig	Log-sigmoid
MAV	Mean Absolute Value
MCP	Metacarpophalangeal
MP	Middle Phalange
MSE	Mean square error
MVC	Maximum Voluntary Contraction
Ni-Ti	Nickel-Titanium
PID	Proportional Integral Derivative
PIP	Proximal Interphalangeal
PP	Proximal Phalange
pureline	linear transfer function
RMS	Root Mean Square
sEMG	Surface Electromyography
SMAs	Shape Memory Alloys
SOM	Self-organizing map
STFT	Shoft Time Fourier Transform

tansig	tan-sigmoid
VAR	Variance
WAMP	Wilson Amplitude
WL	Waveform Length
WT	Wavelet Transform

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Background

The prosthesis as a tool makes no pretence of trying to replace the lost limb’s physiological appearance. As a matter of fact, it works as an aid to help provide some of the functions that were lost due to accident, war or congenital condition. Moreover, the prosthesis is an interchangeable device that can be used only when needed. Much effort in the field of upper-extremity prosthesis research is directed towards the development of prostheses as true limb replacements.

Prostheses were developed for function, cosmetic appearance, and a psycho-spiritual sense of being entire, but not necessarily in that order. These needs have existed from the past until today. Early prosthetic principles that were developed exist to this day and are amazingly efficient in function [1]. The earliest example of a prosthesis for which we have visual evidence is the cosmetic wooden hallux prosthesis found in Egypt about 1000 BCE. A transtibial (above foot but below knee) prosthesis with a wooden socket reinforced with bronze sheets was built by Romans in around 300 BCE. On the other hand, the earliest example of an upper limb prosthesis was designed to allow knights to firmly hold or lock onto a sword or shield in battle in the 15th century. However, this device was usually heavy, cumbersome, and could only function in battle [1].

Circa 1790, a conceptual progress in upper limb prosthetic design continued with Gavin Wilson’s artificial hand, capable of holding a knife, fork, or pen. Peter Baliff, a Berlin dentist developed the first body-powered prosthetic hand activated by elbow and shoulder motion in circa 1816 [1]. The concept of using the remaining muscles of a limb to operate terminal device remained central to all development in upper limb prosthetics until the practical introduction of myoelectrically controlled prosthetics in the beginning of 1958 [1].

The electrical activity naturally generated by contracting muscle in a residual limb is amplified, processed and used to control the flow of electricity from a battery to a motor, which operates an artificial limb [2]. However the design of a clinically useful myoelectric prosthesis is extremely difficult. The prosthesis must be comfortable, work reliably and have a natural appearance both at rest and during functional activities. Otto Bock in Germany began the development of a myoelectric prosthesis and introduced its first system for clinical use in 1967. Myoelectric control was incorporated into the system using two-channel agonist-antagonist myoelectric signals to provide amplitude-modulated control [2].

In fact, most patients or amputees feel uncomfortable when using the current commercial prosthetic hands, because they are usually heavy and unable to provide enough grasping functionality and lack of degree of freedom [3, 4]. Commercially available devices, such as the Otto Bock Sensor Hand [1, 4, 5] and Touch Bionic's i-Limb [11] are far from providing the manipulation capabilities of the human hand. For example, their active bending is restricted to two or three joints, which are actuated by a single heavy motor drive acting simultaneously on the metacarpophalangeal (MCP) joint of the thumb, index and middle finger, while other joints can bend only passively or can not bend at all. The most commonly used myoelectric hand has one degree of freedom, opening and closing. It uses only two myoelectrodes sensors in order to increase training efficiency and practical reliability of the device. If there is need to increase the functionality of prosthetic devices, more myo-electrode sensors can be added, however at the expense of training efficiency.

Several robotic and anthropomorphic hands may have sufficient active dof, such as the Shadow Hand with up to twenty DOF as in [4], MIT Hand, NASA's Robonaut Hand, etc., allowing them to have dexterity comparable to that of the human hand. Unfortunately though, these types of hands cannot be used as prostheses due to their physical characteristic (e.g. pneumatic actuator) that pose several serious limitations on human-hand interaction [6].

Due to the limitations of electric motor actuators being used in commercial prosthetic devices, some researchers have found some potential in using a new generation of small, smart material-based, high powered, biomimetic actuators to replace electric motors [7] in order to improve the functionality and reduce the weight of these prosthetic devices. These artificial muscle actuators include Shape Memory Alloys (SMAs), Electroactive polymers (EAPs) [7, 8] and McKibben pneumatic artificial muscles. EAP materials have higher response speeds and lower densities when compared to SMAs; however their generated actuation force is extremely lower than SMAs [9]. The big advantages of SMAs are their incredible small size, volume and weight, their high force-to-weight ratio, their low cost and human muscle-like behaviour. However, they also have some limitations, such as the large length of wire required to create significant motion, limited life cycle and non-linear effects such as hysteretic phenomena [3].

In the light of the historical background and those problems encountered in current devices, further research and development is warranted.

1.2 Aim and objectives

The aim of this work is to investigate the use of a hybrid actuation mechanism in the design and development of an intelligent prosthetic hand. This work highlights user-friendliness in usage and involves a proper mechanical design in more active degrees of freedom incorporating an intelligent control system.

The aim of this work will be associated with the objectives stated below.

1.2.1 Objectives

A number of objectives specified in this work allow the realisation of the main aim. These are as follows:

1. Prosthetic Finger Design

   To propose and vigorously study a prosthetic finger. In order to avoid complexity in hand design and control, a prosthetic finger design is proposed and vigorously studied. It is postulated that the hand design and control would be similar to a single prosthetic finger. This finger design is used as a prototype to allow the hybrid actuation mechanism to be implemented in order to provide some functionality of a real hand, such as coarse and fine grasping.

2. Hybrid Actuation System Design

The actuation mechanism used is a hybrid system consisting of a DC actuator and a Shape Memory Alloy (SMA) actuator. This hybrid actuation system will allow to increase the degree of freedom of the prosthetic finger.

3. Hybrid Actuation Control System Design

A control system is a device to manage, command, direct or regulate the behaviour of other devices or systems. The system is designed to control the hybrid actuation mechanism in order to 'bring' the finger, especially fingertip, in to a correct position.

4. Control Input Signal Consideration

The electrical activity generated by contracting muscles in forearm will be used as a control input signal for the prosthetic fingers. The electrical activity of the muscle is called a myoelectric signal and is monitored using surface Electromyography (sEMG) sensors (myoelectrodes). Only two myoelectrodes will be used in order to facilitate the amputees' or users' training and use.

5. Control Input Signal Processing

The raw surface EMG (sEMG) signal contains valuable information in a particularly useless form. This information is useful only if it can be quantified by a signal processing method to achieve an accurate and actual sEMG signal that we would like to use.

6. Control Input Signal Pattern Recognition

    The pattern recognition will be considered using pattern classification technique to train and test the varying patterns of the human finger's movement (thumb is not considered).

1.3 Thesis outline

The remainder of this thesis is divided into eight chapters and the content of each is summarised below.

Chapter 2: Literature Review

Prior to presenting the design, analysis and experimental aspects of this research, it is useful to have a historical perspective or a literature review. This chapter provides the background history on the two major components of this research: actuation and control. The actuation components include the mechanical hand/finger design and the control includes sEMG signal processing techniques.

Chapter 3: Design of a Prosthetic Finger

This chapter begins by discussing the preliminary finger design and development, which allow the hybrid actuation mechanism to be implemented in order to provide some functionality of a real hand, such as coarse and fine grasping. In order to avoid complexity in hand design and control, a prosthetic finger design is proposed and vigorously studied. It is postulated that the hand design and control would be similar to a single prosthetic finger.

Chapter 4: Hybrid Actuation Mechanism

Development of a hybrid actuation system applied to the mechanism finger design of Chapter 3 will be discussed. A selection of appropriate gears or other additional mechanism parts of the system will also be presented. The main task of this chapter is to show that our design of the prosthetic finger closely resembles the functionality of the human finger.

Chapter 5: Control System Design of Actuation Mechanism

This chapter is presented to describe mathematical tools and algorithms that build dynamical models from input-output data. The input data of the DC and SMA actuators are the value of input voltage and the output data are the pulse of the encoder in the DC actuator and the stroke change in the SMA actuator.

Chapter 6: Electromyography Signal and its Processing Technique

This chapter will discuss the electromyography signal and its signal processing technique. Electromyography (EMG) is an analytical tool used to record the electrical activity of muscles. It can be used for clinical/biomedical applications. The raw data from surface EMG (sEMG) signals contain valuable information in a particularly useless form. This information is useful only if it can be quantified by a signal processing method to achieve accurate and actual sEMG signal.

Chapter 7: Pattern Recognition

This chapter will discuss the development of a control system for finger pattern classification by using an artificial neural network (ANN). This will involve the fundamental of the neural network and the result of how many finger movement patterns can be trained.

Chapter 8: Summary and Conclusions

In this final chapter, the work of the thesis is summarised. It discusses the current work, summarises the results and proposes ways to overcome the limitations and issues to be encountered in future development of the prosthetic hand configuration. Recommendations are also made for future work.

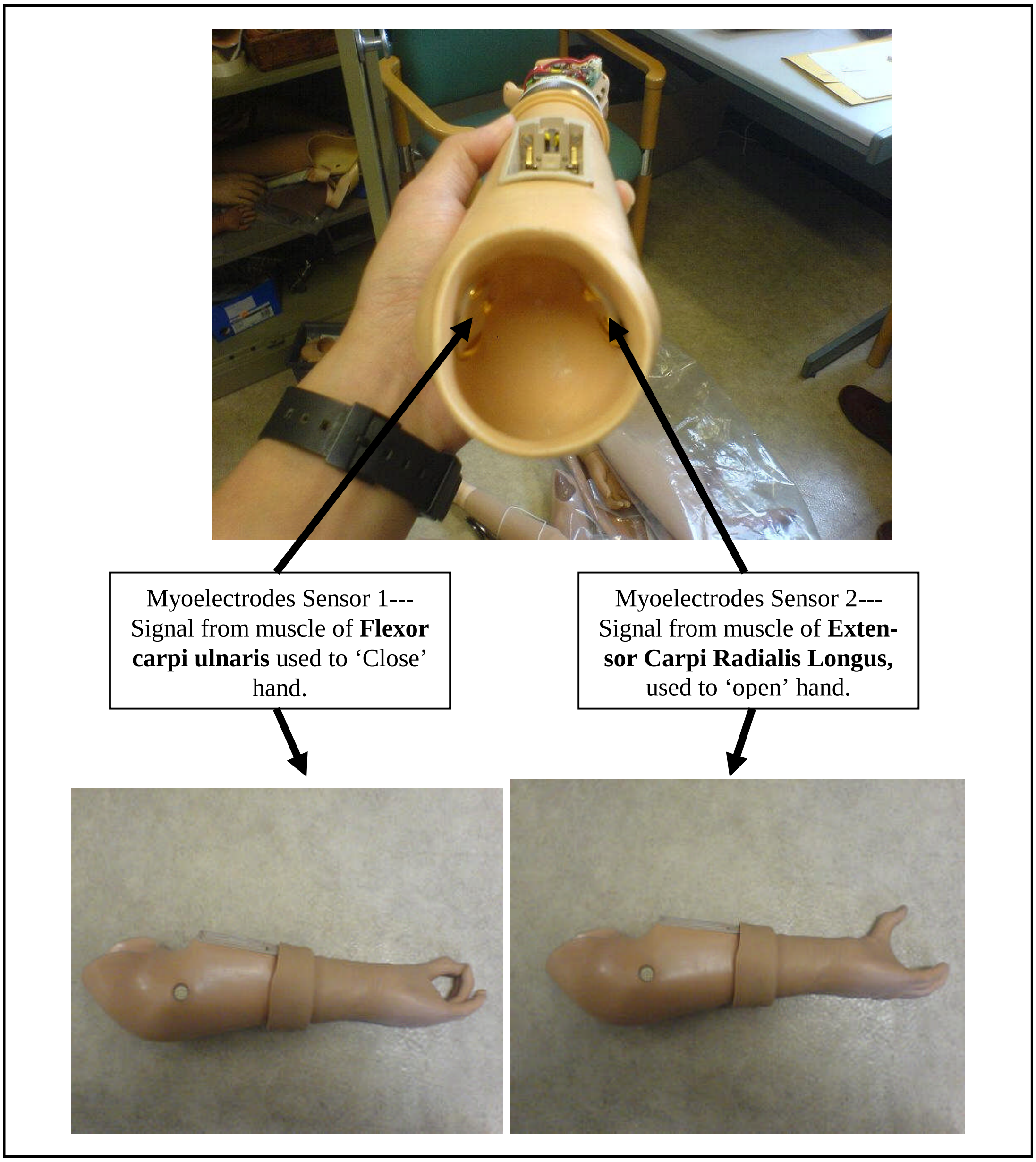
CHAPTER 2

Literature Review

2.1 Introduction

The loss of limb can severely affect the quality of life of an amputee and thus render the most common every day tasks difficult if not impossible. Over the last few decades, major progress has been made in the development of intelligent prostheses, which can at least partially fulfil the requirements of the missing limb. The hand is viewed as one of the most important parts of the human body as it allows for adaptation, exploration, prehension, perception and manipulation.

The concept of using the remaining muscles of a limb to operate a terminal device remained central to all development in upper limb prosthetics until the practical introduction of myoelectrically controlled prosthetics in the beginning of 1958 [1]. The myoelectric prosthesis is controlled by the action potential of the muscle that develops as a secondary to the excitation of the central nervous system. The articulated hand of the myoelectric prosthesis is usually activated through at least two sets of electrodes over an opposing muscle group, such as extensors and flexors. The electrical potential of the muscle is detected on the skin by the electrodes incorporated into the socket of the prosthesis. This potential, picked up by the electrodes, then controls the opening and closing action of the prosthesis, powered by a battery usually incorporated within the prosthesis [10]. Figure 2.1 shows the current commercial myoelectric prosthetic hand for children manufactured by Otto Bock.



Myoelectrodes Sensor 1---Signal from muscle of **Flexor carpi ulnaris** used to 'Close' hand.

Myoelectrodes Sensor 2---Signal from muscle of **Extensor Carpi Radialis Longus,** used to 'open' hand.

Figure 2.1: Myoelectric prosthetic hand for children.

Prior to presenting the design, analysis, and experimental aspects of the work of myoelectrical prosthetic hand, it is useful to have an historical perspective or a literature review. This chapter provides the background history on the two major components of this research: actuation and sEMG control input signal. The actuation components are pneumatic, hydraulic, SMA, etc., which is discussed in Section 2.2. The sEMG control input signal includes the digital signal processing and sEMG pattern recognition discussed in Section 2.3.

2.2 Actuation System Design

An actuator is a mechanical device for moving or controlling a mechanism or system. It is typically a mechanical device that takes energy, usually created by air, electricity or liquid, which it converts into some kind of motion. In engineering, actuators are frequently used as mechanisms to introduce motion or to clamp an object so as to prevent motion. In electronic engineering, actuators are a subdivision of transducers and are devices which transform an input signal that is mainly an electrical signal into motion. Specific examples include electrical motor, pneumatic actuators, electroactive polymers (EAP), shape memory alloy (SMA), etc..

Electrical motors, pneumatic actuators, hydraulic actuators and SMA actuators are typically used in prosthetic limb design [1, 3, 6, 7, 11-15]. Current commercial myoelectric prosthetic limbs mostly use electrical motors, such as Otto Bock [1] and Touch Bionics' i-limb [11].

2.2.1 Review of Actuators

Traditionally, artificial hand designs have tended to be bulky, heavy and noisy due to use of electro-mechanical actuators [16]. Besides, the electro-mechanical actuators' heavy weight, size, and shape have been restrictive and uncomfortable to the users [3, 12, 16]. As a result of the problems inherent with this type of actuation, designers have been adapting various other actuation techniques for use in their place.

Typically, electric motors are not small enough and can only be placed remotely, like ahead of the wrist joint of the hand, such as in the Otto Bock prosthetic hand (Figure 2.1). This positioning increases the complexity of the mechanical designs, since all the tendons driving the fingers have to pass through the wrist joint. Due to these constraints, the design and development of most of the current commercial prosthetic hands have failed to provide enough grasping functionality and have few active DOF restricting their effectiveness [15].

Several robotic and anthropomorphic hands may have sufficient active DOF, such as the Shadow Hand with up to 20 DOF [17], allowing them to have dexterity comparable to that of the human hand, though unfortunately these types of hands cannot be used as prostheses due to their physical characteristics.

Constrained by these limitations of the electric motor or electro-mechanical actuator, some researchers have used pneumatic, hydraulic or SMA actuation to replace the motor.

First, among the pneumatic actuators, the McKibben Artificial Muscle [14] is one of the most excellent examples. Its properties are found to be similar to the real muscle in the sense of its physical flexibility and light weight. We can refer to the Anthroform Biorobotic Arm [18] for an idea of the application. The force-to-weight ratio of this device is one of the best attributes, reaching 400:1, which is absolutely critical for mobile robot applications. This technology was commercialised for robotic applications in the 1990s by the Shadow Robot Group in the UK. A rubber tube covered in tough plastic netting which shortens in length constituted in their air muscle, which looks like a human muscle when inflated with compressed air at low pressure and contracts by up to 40% of its original length when actuated. The air muscle has smooth and neural movement because it has no "stiction" and it has an immediate response. According to the Shadow Robot Group Company's, their Air Muscle diameter is just 6mm, with the strength, speed and fine stroke of a finger muscle in a human hand, as shown in Figure 2.2. It is also able to lift more than 70 kg at a pressure of four bars (60 psi) when its Air Muscle diameter is 30mm.

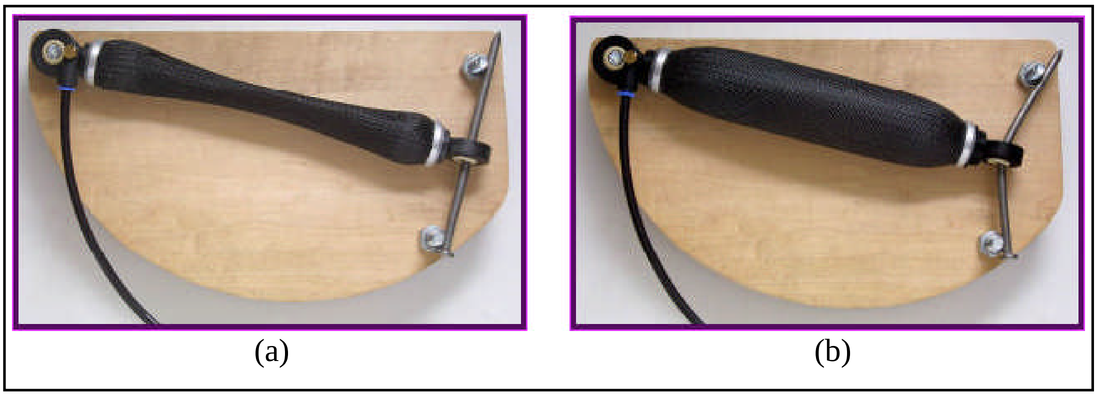


Figure 2.2: Pneumatic muscle from Shadow Robot Group Company: (a) Muscle extending, (b) Muscle contraction.

Secondly, the hydraulic actuator [13] presents a powerful miniaturised hydraulic system of compact design used for the actuation of artificial hands. These hydraulic prostheses have advantages in construction, design and performance, including adaptivity during grasping and holding of an object. This hydraulic system consists of one hydraulic pump, one fluid reservoir, five electric valves, an electronics unit, eight flexible fluidic actuators, plus one battery for power supply and two myoelectric electrodes placed in the socket. Figure 2.3 shows this miniaturised hydraulic actuator for artificial hands, the maximum grasping force of which is 110N.

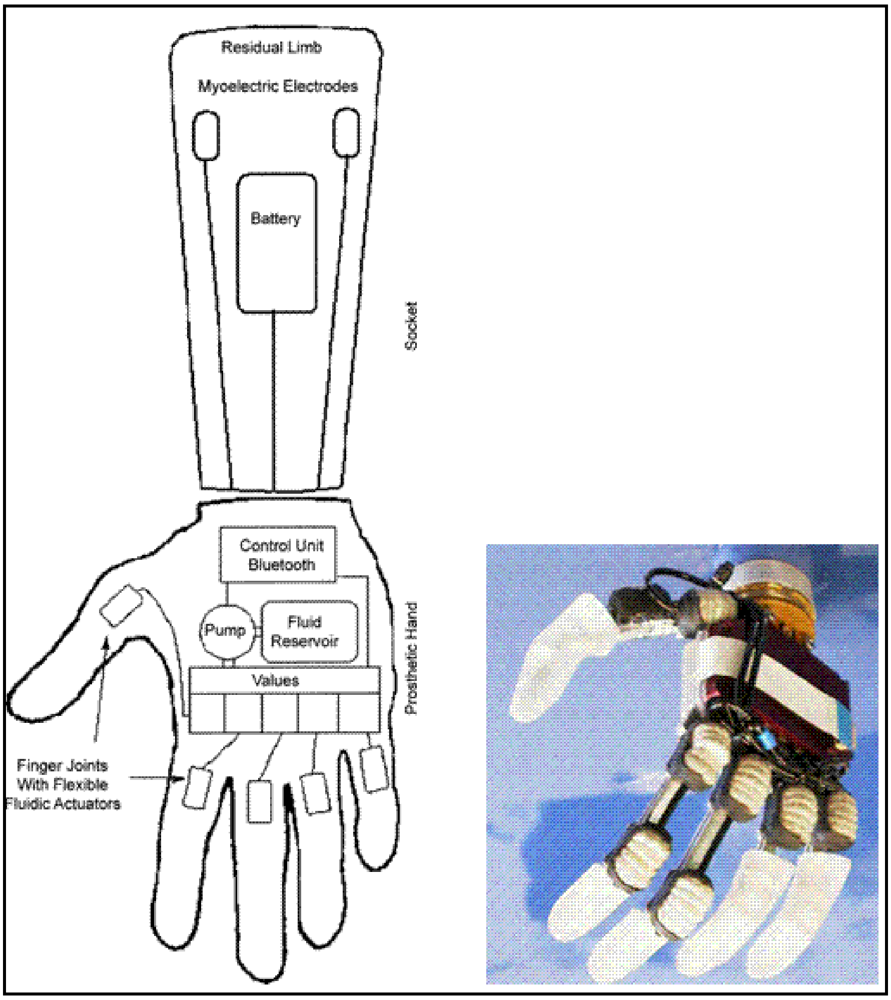


Figure 2.3: Miniaturised hydraulic actuator for artificial hands [13].

This prosthetic hand prototype may accomplish up to five grasping patterns, which are power, precision, tripod and hook grasp, and stretching of an index finger. The grasping patterns are pre-assigned in the hand’s control unit and are accomplished as follows. Firstly, the electronics control unit receives the control signals from the user’s upper limb muscle via the two myoelectrodes. The control unit is responsible for the analysis of control signals, selection of pre-programmed grasping patterns and operation of the hydraulic pump and valves. Afterwards, the hydraulic pump generates the fluid pressure to the actuators and electric valves control the actuator action. Depending on the grip type selected, the corresponding actuators will be filled and the joint will move. However, it has a total of just five active DOF (one at each finger) due to use of five electric valves and three passive DOF on the PIP joint of thumb, index and middle fingers.

Third, SMA allows crystalline phase and shape changes when heated or cooled [7]. Heating is enabled by passing an electric current through the wire. As the wire cools, it expands back to its original length. Although they are sometimes considered slow for demanding robotic applications, recovery rates can be increased significantly by using various heat sinks [7]. There are a lot of types of memory alloys, but compared with corrosion-resistant nickel-titanium alloy (Ni-Ti), Cu-Zn-Al and Cu-Al-Ni are the worst in their chemical, physical and mechanical characteristics. As for Ni-Ti, it creates direct linear motion by contraction up to the 10% of its length when heated.

A lot of researchers prefer using SMA attached to tendon wires as their prosthesis actuator due to its operational similarity to human muscles [8]. Besides, the big advantages of the SMA are their incredibly small size and volume, their high force-to-weight ratio and their low cost [3, 19]. However, the one drawback with SMA is that in prosthetic devices large lengths of SMA are needed [3].

A new biomimetic tendon-driven actuation system is presented in [6] for prosthetics based on the combination of compliant tendon cables and SMA wires that form a set of agonist-antagonist artificial muscle pairs for the required flexion/extension or abduction/adduction of the finger joint (Figure 2.4). The performance of the proposed actuation system is demonstrated using three active and one passive DOF of each finger. The three active dof are positioned on the MCP joint flexion/extension, MCP joint abduction/adduction and PIP joint flexion/extension. One passive DOF is located on the DIP joint, coupled with the PIP joint.

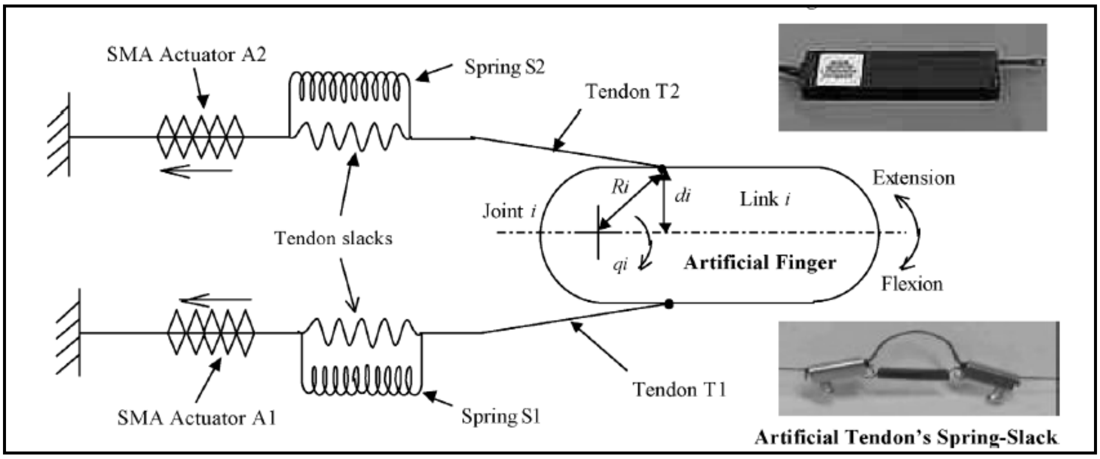


Figure 2.4: Differential spring-biased SMA joint actuation mechanism [6].

In Figure 2.4, one end of the tendon cable is attached to the artificial finger structure, mimicking the attachment of the natural tendon to the finger bones, while the other end of the cable is tied to the SMA actuator. The actuators are placed remotely to the finger joint, similar to the natural finger's extrinsic musculature. Joint rotation is produced by the contractile action of two SMA actuators, placed in opposition to each other in a double spring-biased fashion. As shown in the lower inset of Figure 2.4, passive compliance is introduced in the tendon cables of the artificial finger by connecting a spring in parallel to a slack portion of each tendon cable, such that, as the SMA actuator contracts, the spring in the corresponding tendon elongates until the slack is absorbed and the tendon is tight. At this point, the tendon cable can be considered to have 'infinite' stiffness and further SMA actuator contraction causes tension to be transferred to the finger for link rotation. This simple spring-slack artificial tendon effectively mimics the nonlinear stiffness of the natural tendon which tends to infinity as it approaches its natural limit of extension. The dual spring-biased configuration permits the two SM actuators to work as an agonist-antagonist pair, enabling both active extension and flexion of the joint.

As an SMA actuator contracts, the spring in the tendon cable to which the actuator is connected expands, absorbing the slack in the active tendon until the cable is fully stretched and tight. While the finger can rotate simultaneously during the absorbing of the slack, any further contraction of the SMA actuator at this point acts directly on the finger, rotating it about the joint axis. Simultaneously, as the finger flexes or extends, the spring in the opposing tendon expands and the slack in the passive tendon is also absorbed. Figure 2.5 shows the artificial finger with six tendon cables routed through the finger core and attached to the corresponding six remotely placed SMA actuators.

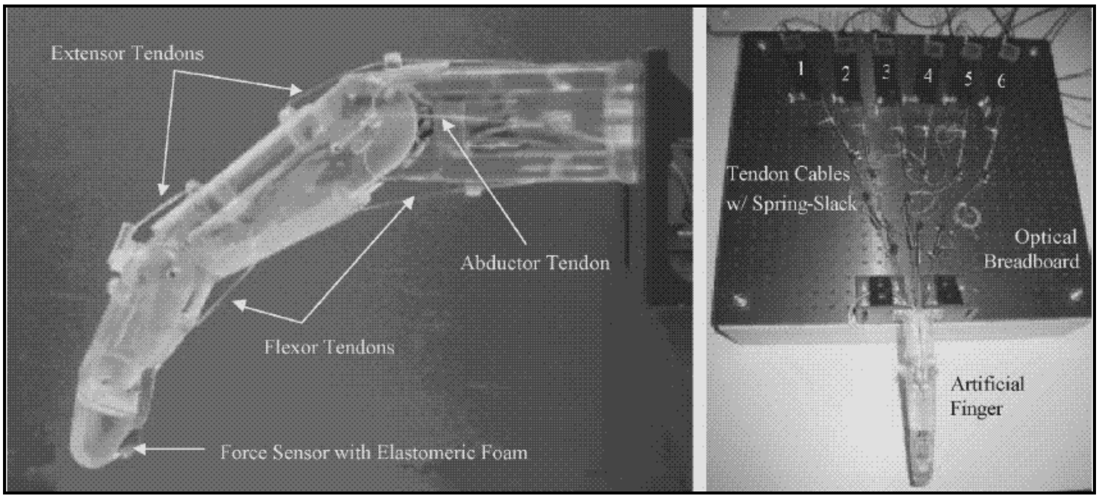


Figure 2.5: Artificial finger with six tendon cables routed through finger core and attached to six corresponding remotely placed SMA actuators [6].

Electroactive Polymers (EAPs) are another type of actuator to replace the electric motor or hydraulic/pneumatic one, being operationally similar to the SMA and human muscles. It is a lightweight and direct-driven actuator able to provide ways to increase the functionality of artificial hands without adding mechanical complexity [8]. However, EAPs have low actuation forces, mechanical energy density and lack of robustness compared to the SMA [9]. Hence, a lot of researchers refer more to use of SMA than EAP.

Even though the pneumatic, hydraulic and SMA actuators have achieved some success, sometimes they are not ideal in use. For example, pneumatic and hydraulic actuators need electric valves and a compressed air generator or pump too, which are neither light nor small. As for SMA, its thermal to mechanical energy conversion has proved difficult in practical applications and just a small amount of absolute force is obtained from one SMA wire.

Considering the above advantages and limitations of some actuators, we finally decided to combine DC and SMA actuators. The DC actuator is small and light enough and is placed in the palm to control the flexion/extension movement of the MCP joint. The SMA is placed in the palm but at a different level from the DC actuator and to control the flexion/extension movement of the PIP joint. This combination can not only increase the finger's active dof but also can create a light enough prosthetic hand. Besides, it will save spare in the prosthetic socket as we are trying to position these two actuator inside the palm.

2.3 sEMG Control Input Signal

An electromyograph (EMG), is also referred to as a myogram is an analytical tool used to record the electrical activity of muscles. Basically, the EMG produces an electrical current which is proportional to the level of muscle activity. As it can also be used to detect any abnormal electrical activity in the muscle, it plays an important role in the diagnosis of many diseases and conditions, for example muscular dystrophy, inflammation of muscle, pinched nerves or even peripheral nerve damage [20, 21].

There are two types of EMG: intramuscular or invasive EMG and surface or non-invasive EMG [22, 23]. The intramuscular EMG is actually a needle electrode inserted through the skin into the muscle. Its activity is displayed on an oscilloscope. Several needle electrodes may be placed at various points in the muscle to obtain an informative EMG. The information (size and shape of the waveform) produced on the oscilloscope enables the muscle to respond to nervous stimulation. The size of the muscle fibre affects the frequency and the amplitude of the action potential [22, 24]. Surface EMG (sEMG) is obtained by placing the electrodes just on and not into the skin to detect the electrical activity of the muscle. Therefore the sEMG does not involve piercing the skin and does not hurt [22]. The current state of sEMG provides many important and useful applications but it has many limitations, which must be understood, especially the factors which influence the EMG signal [21, 25].

From [25], it is known that there are some typical disturbances for myoelectric prostheses. These are motion artefacts caused by movement or detachment of myoelectrodes, low frequency electrical interference and radio frequency interference, e.g. by mobile phones or radio sets.

The myoelectrode should be placed on the thickest part of the muscle (position 1 of Figure 2.6), in order to obtain the strongest signal. The differential inputs should be in the muscle fibre direction. Positions 3 and 4 of Figure 2.6 are less suitable, as there are fewer motor units in the pickup area of the myoelectrodes. Position 2 should also be avoided, because interference of the motor unit action potentially reduces the signal strength in that area [21, 25].

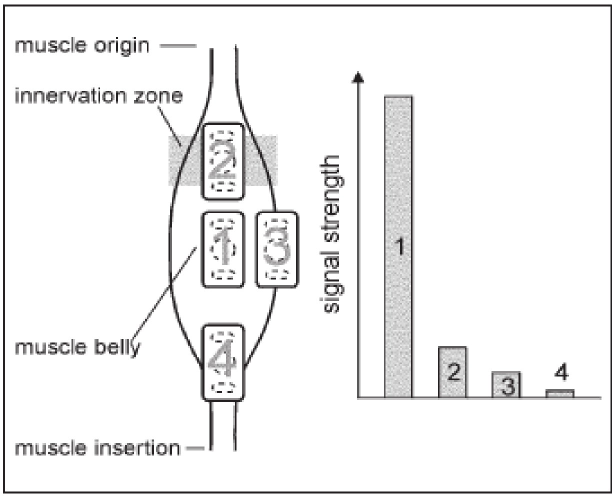


Figure 2.6: Influence of myoelectrode position on signal strength[21, 25].

The purpose of the implanted myoelectric sensors (Figure 2.7) is to provide multi-function prosthesis control [24]. The implantable sensor is small enough to be injected into muscles, around 2mm diameter by 15mm long and is individually addressable and hermetically encapsulated.

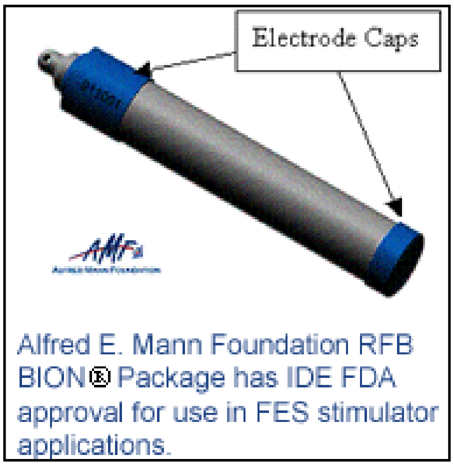

Electrode Caps
AMF
Alfred E. Mann Foundation RFB BION® Package has IDE FDA approval for use in FES stimulator applications.

Figure 2.7: Implanted myoelectric sensors (IMES) [24].

The implanted myoelectric sensors will receive their power, digital addressing and command signals by forward telemetry from an external transmitter coil laminated into the prosthetic socket. The implants will send their EMG data to the prosthesis controller by reverse telemetry (Figure 2.8) [24].

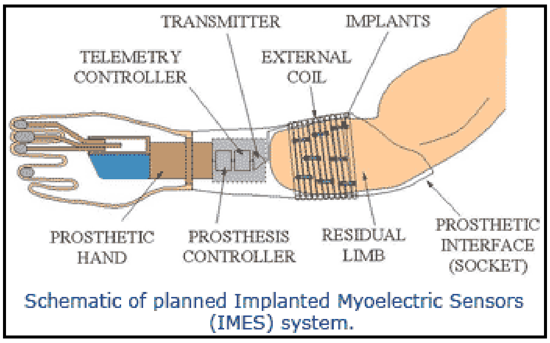


Figure 2.8: Implantable myoelectric sensor (IMES) concept. External coil laminated directly into prosthetic interface during socket fabrication and telemetry controller incorporated along with prosthesis controller in body of prosthesis [27].

2.3.1 Surface Myoelectrodes

The commercial prosthetics, such as Otto Bock Sensor Hand and Touch Bionic's i-Limb, use only two myoelectric electrodes [1, 4, 5]. However, some researchers such as [26] suggest that this, as far as one can understand, offers no fine control over single finger or over the required amount of force. Besides, the current commercial devices are far from providing the manipulation capabilities of the human hand. For example, their active bending is restricted to two or three joints, actuated by a single heavy motor drive acting simultaneously on the MCP joint of the thumb, index and middle fingers, while other joints can bend only passively or can not bend at all. The most commonly used myoelectric hand has one DOF, opening and closing [4, 26, 27].

In order to have fine control and for the patients to be able to enforce the correct grasping type, more than two myoelectric electrodes are needed [26, 28-31]. Some researchers [28] use 64 channel of myoelectric electrodes for decoding twelve finger movements, which are ten individuated finger movements (finger flexion and extension) and two middle, ring, and little fingers as group movements (flexion and extension). The electrodes are placed by dividing the posterior and anterior forearm into five levels respectively, where the levels one to five are characterised by 32, 28, 24, 19 and 12 electrodes, respectively. The accuracy is above 80% for pattern classification. In another work [30] eight myoelectrodes are used to decode six gestures patterns of hand movement and achieved accuracy up to 93.7%. Up to ten myoelectrodes has also been used [26] to decode five patterns: no action taken; grasp by opposing thumb and index finger, thumb and middle, thumb and ring, and thumb and all other fingers. Its accuracy is around 80% for pattern classification. Some researchers [31], using up to fifteen myoelectrodes attempted to decode five patterns, which are the MCP joint's movement of thumb, index, middle, ring and little fingers.

It is seen that these kinds of techniques can increase the number of hand/finger movements and the functionality of a prosthetic hand. However, increasing the number of myoelectrodes also increases the complexity of learning by amputees. Besides, the remaining limb of 'real' amputees has just limited space to allow the placing of so many of myoelectrodes.

2.3.2 Signal Processing Techniques

Generally, the processing of sEMG signals can be divided into three stages [32-34]. The first is data acquisition that includes amplification, analogue to digital conversion and signal conditioning. Second, a signal processing stage to extract desired features from the biosignal, and third, a feature selection stage by retaining information important for the later application such as classification of signals using an Artificial Neural Network (ANN). A schematic diagram of the stages is shown in Figure 2.9.

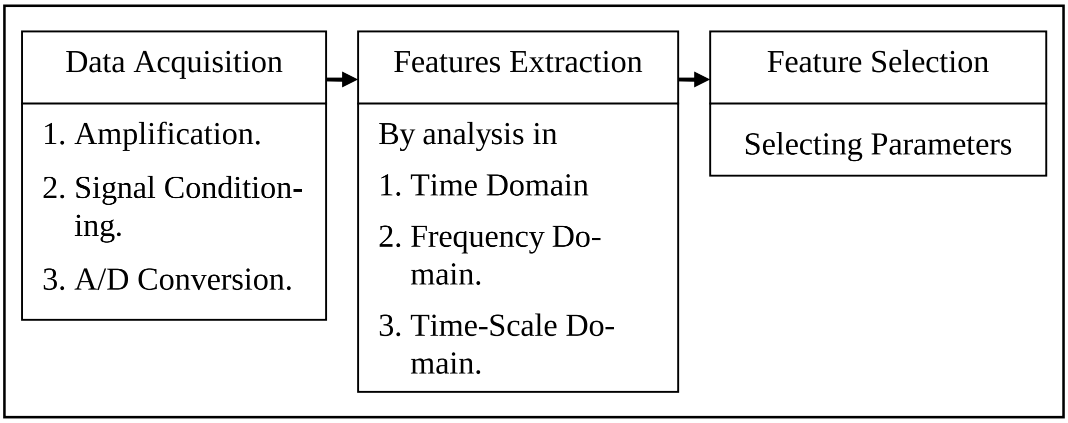


Figure 2.9: Stages of sEMG signal processing.

2.3.2.1 Data acquisition

A signal is first detected at the intended biological site by using surface electrodes as sensors. The electrodes also provide interface between an electrical recording device and the biological system. After detection by the electrodes, the signal is usually amplified, filtered and converted to a digital signal monitored on the screen. Amplification is needed as muscle signal is generally weak and only generates in the microvolt range [35]. Since the sEMG signals are small, their measurement is also susceptible to interference from electrical equipment, such as movement of cable carrying signals from the body to the measuring instrument. Besides, in order to eliminate the 'unwanted' interference signal, typically a bandpass filter is used as first stage at the electrode site which effectively cancels the ambient electrical noise.

2.3.2.2 Feature Extraction and Selection

Feeding a myoelectric signal presented as a time sequence directly to a classifier is impractical, due to the large number of inputs and randomness of the signal. Therefore, the sequence must be mapped into a smaller dimension vector, called a feature vector [36, 37]. Information extracted from EMG signals represented in a feature vector is chosen to minimise the control error [38, 39]. Features represent raw myoelectric signals for classification, so the success of any pattern recognition problem depends almost entirely on the selection and extraction of features. A wide spectrum of features has been introduced in the literature for myoelectric classification. Features fall into one of three domains: time, frequency and time-scale [20, 32, 36, 39-42].

2.3.2.2.1 Time Domain

Common procedures are used to detect muscle activation. These are basically described by the observable lobes appearing in the sEMG time series. There are several digital operations that can be performed to obtain the desired information.

1. *Full-wave Rectification*

   One of the first operations performed on the sEMG signal is full wave rectification. This can be done digitally by defining:

$$\mathrm{sEMG}_{\mathrm{rec}}(\mathrm{i}) = |\,\mathrm{sEMG}(\mathrm{i})\,| \quad \text{(Equation 2.1)}$$

   where sEMG(i) is the *i*th sample of the discrete sEMG signal.

2. *Root Mean Square*

This is used to calculate the amplitude of the sEMG and therefore also as a force indicator. It is defined by:

$$\mathrm{RMS} = \sqrt{\frac{\sum_{i=0}^{n} A_i^2}{N}} \quad \text{(Equation 2.2)}$$

where A_i is amplitude in *i*th sample and N is total number of samples.

3. *Mean Absolute Value*

This is an estimate of the mean absolute value (MAV) of the signal x_i in a segment *i* that is N samples in length.

$$\overline{X}_i = \frac{1}{N}\sum_{i=1}^{N} | x_i | \quad \text{(Equation 2.3)}$$

4. *Wilson Amplitude*

This is the number of counts for each change of the sEMG signal amplitude that exceeds a predefined threshold. It is given by:

$$\mathrm{WAMP} = \sum_{i=1}^{N} f(| x_i - x_{i-1} |) \quad \text{(Equation 2.4)}$$

where f(x) = 1 if x>threshold, 0 otherwise. This unit is an indicator of firing of motor unit action potentials (MUAP) and, therefore, an indication of muscle contraction level.

5. *Variance:*

The variance is a measure of the signal power and is calculated as:

$$\mathrm{VAR} = \frac{1}{N-1}\sum_{i=1}^{N} x_i^2 \quad \text{(Equation 2.5)}$$

where x_i is the *i*th sample and N is the total number of samples.

6. *Waveform Length:*

This is the cumulative length of the waveform over the time segment. It is defined as:

$$l_0 = \sum_{i=1}^{N} | \Delta x_i | \text{ (Equation 2.6)}$$

where $\Delta x_i = x_i - x_{i-1}$. This parameter gives a measure of waveform amplitude, frequency and duration all in one.

7. *Autoregressive Coefficients:*

Using an autoregressive (AR) model, new samples are represented as linear combination of earlier samples. The model can be represented as:

$$\hat{y}_i = \sum_{k=1}^{M} a(k) y(i-k) + w(i) \text{ (Equation 2.7)}$$

where y is the sEMG signal, a(k) is the coefficients of the model, w(i) is a random white noise and M is the order of the model. M=4 is suitable for EMG signals [41].

2.3.2.2.2 Frequency Domain

Spectral (frequency domain) analysis is mostly used to study muscle fatigue [36, 41]. It is commonly used in applications where oscillators or repetitive patterns are involved, for instance in the case of motor unit (MU) activation and pathological tremor [41].

Power spectral density (PSD) plays a major role in spectral analysis. In wide-sense stationary stochastic signals, PSD is defined as the Fourier transform of the autocorrelation function of a signal. Its two characteristic variables, the mean and median frequency, provide some basic information about signal spectrum and its change over time. Fourier analysis is a mathematical technique for transforming a signal from time domain to frequency domain by breaking down a signal into constituent sinusoids of different frequencies. Fourier transform is a generalisation of the Fourier series where function is represented by the sum of sines and cosines. Instead, Fourier transform uses exponentials and complex numbers.

The Fourier transform of input signal x(t) is defined as:

$$F(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} \text{ [43] (Equation 2.8)}$$

<u>where</u> ω is the angular frequency, $\omega = 2\pi f$, f is the input frequency, x(t) is the time domain signal and $F(\omega)$ is its Fourier transform represented in frequency domain.

Equation 2.8, expressing the Fourier transform, calculates the frequency, amplitude and phase of each sine wave needed to make up any given signal. It is a linear transform from time to frequency domain and can be used to analyse the spectral component of a signal [43].

1. *Mean Frequency*

Fatigue is related to the frequency of motor unit (MU) activation [41]. The evolution of the mean frequency is used as a fatigue index. It is defined as:

$$\overline{F} = \frac{\sum_{i=0}^{n} f_i A_i^2}{\sum_{i=0}^{n} A_i^2} \text{ (Equation 2.9)}$$

<u>where</u> $\overline{F}$ is mean frequency, f_i is frequency in *i*th sample, A_i is amplitude in *i*th sample and *n* is total number of samples.

2. *Median Frequency*

This is another parameter that can be used to assess muscle fatigue. The median frequency is given by the frequency that divides the power spectrum into two regions containing the same amount of power. The median frequency is the frequency having 50% or half of the frequency distribution on each side.

2.3.2.2.3 Time-Scale Domain

Fast Fourier Transforms (FFTs) are most commonly used to determine the frequency spectrum of the sEMG signal. However, its analysis has a serious drawback. In transforming to the frequency domain, time information is lost. When looking at a Fourier transform of a signal, it is impossible to tell when a particular event took place [36, 44, 45]. This is acceptable for stationary signals, as their properties do not change over time. However, myoelectric signals contain numerous non-stationary characteristics. Another Fourier Transform, Short Time Fourier Transform (STFT), is a form to solve the FFT drawback and maps a signal into time and frequency functions. Hence, it provides information telling ‘when’ and ‘at’ what frequencies a signal occurs [36, 44]. However, because it is using a technique called windowing to analyse a small section of signal, the information is therefore obtained with limited precision determined by the size of the window. Thus, narrow window has good time resolution, but poor frequency resolution; while wide window has poor time resolution, but good frequency resolution [32, 36, 44]. Besides, another drawback of STFT is assuming the signal is stationary within the window size, therefore it is not suitable to be used for non-stationary sEMG signal [46]. This resolution problem suggests that there is a need to use variable lengths in analysing windows with short ones for high frequencies and long ones for low frequencies. Wavelet transform is the method able to accommodate this with the use of related time-scale analysis, thus providing a flexible time-frequency resolution. The wavelet analysis does not use a time-frequency region, but rather a time-scale region [44].

A wavelet transform (WT) enables local analysis, i.e. to analyse a localised area of a large signal. Wavelet analysis reveals data aspects that other techniques miss, such as trends, breakdown points, discontinuities in higher derivative and self-similarity [44]. WT is an efficient mathematical tool for local analysis of non-stationary and fast transient signals [20]. It does not use a time-frequency region, but rather a time-scale region to analyse signals, as shown in Figure 2.10. Thus it provides a flexible time-frequency resolution, unlike FFT and STFT, which are analysis in frequency domain and Frequency-Time Domain [44].

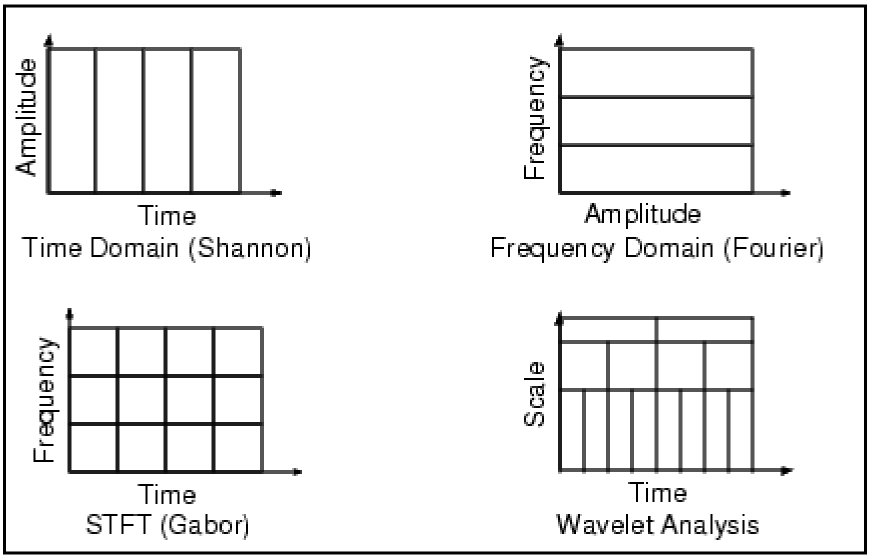


Figure 2.10: Different domains of signal analysis [44].

Wavelet analysis breaks up the signal into shifted and scaled versions of the original or ‘mother’ wavelet. Wavelet transforms use basis function or mother wavelet with time width adapted to each frequency band [44]. Wavelets are used to transform the signal under investigation into time-scale representation which presents the signal information in a more useful form. The idea of relative time-frequency resolution allows the time-scale component to be considered as related by a time translation and a time stretch [44].

There is a correspondence between scale and frequency in wavelet analysis: a low scale shows the rapidly changing details of a signal while a high frequency and a high scale illustrates slowly changing coarse features, with a low frequency [36, 44]. Figure 2.11 shows the relationship between scale and frequency.

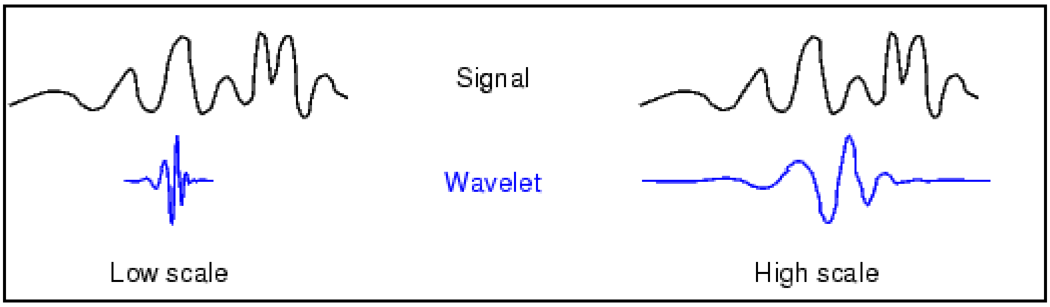


Figure 2.11: Low and high scale signal [44].

Two types of wavelet transform can be defined: the continuous wavelet transform (CWT) and the discrete wavelet transform (DWT).

Due to the computational condition, real-world data signals have been processed at discrete time to perform in discrete signals form. Hence, unlike the DWT, the 'continuous' character in CWT is marked by its ability to operate at a scale from that of the original signal up to some maximum scale determined by the application needed for detailed analysis with available computational power. The translation of CWT is also 'continuous' during computation as the analysing wavelet is shifted smoothly over the full domain of the analysed function [44]. Figure 2.12 shows the CWT in terms of 'continuous' shifting.

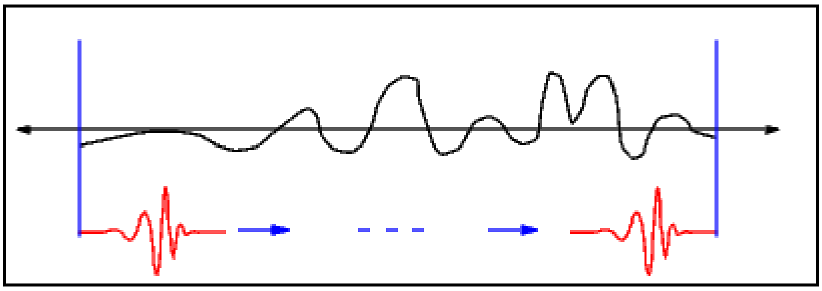


Figure 2.12: CWT 'continuous' in terms of shifting [44].

The CWT is defined as the sum over all time of the signal multiplied by scaled, shifted versions of the wavelet function, as shown in Equation 2.10 [44].

$$\mathrm{CWT}(a,\tau) = \int_{-\infty}^{\infty} x(t)\psi_{a,\tau}(t)dt \quad \text{(Equation 2.10)}$$

where $x(t)$ is input signal, position τ represents the translation diameter of time shifting, a represent scale and the basic function $\psi_{a,\tau}$ is obtained by scaling the mother wavelet $\psi(t)$ at position τ and scale a.

The mathematical expression of a wavelet family which consists of members or daughter wavelets $\psi_{a,\tau}$ is obtained by scaling and time-shifting of the mother wavelet $\Psi(t)$, defined in Equation 2.11 [44]:

$$\psi_{a,\tau}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-\tau}{a}\right) \quad \text{(Equation 2.11)}$$

When ‘a’ becomes large, the basic function $\psi_{a,\tau}$ becomes a stretched version of the prototype, which emphasises the low frequency components. A small ‘a’ contracts the basic function $\psi_{a,\tau}$ and stresses the high frequency components [44].

Substituting Equation 2.11 to Equation 2.10 gives the following [44]:

$$\mathrm{CWT}(a,\tau) = \left(\frac{1}{\sqrt{a}}\right)\int_{-\infty}^{\infty} x(t)\psi_{a,\tau}\left(\frac{t-\tau}{a}\right)dt \quad \text{(Equation 2.12)}$$

An example of a signal analysed by CWT is depicted in Figure 2.13. The darker and brighter regions indicate larger coefficients and stronger transforms for the corresponding scales and time of occurrence. The colour bar on the right side of the scalogram plot is the number of CWT coefficients. Large value of coefficient will stretch the signal (low frequency) and low value of coefficient will compress the signal (high frequency), as shown in Figure 2.14 [44].

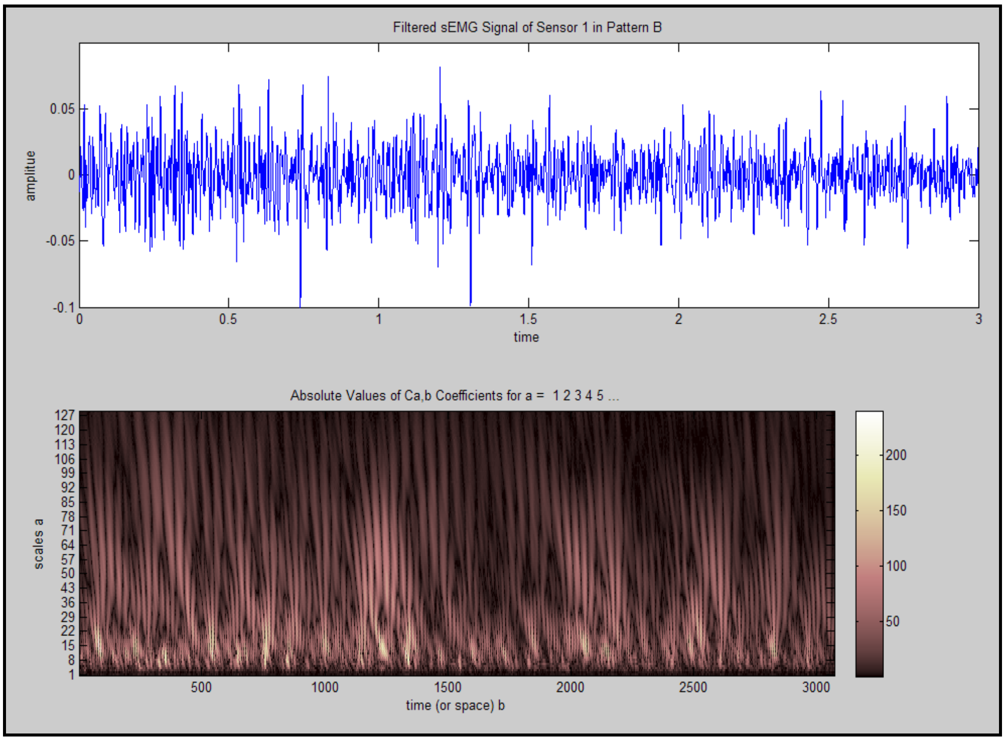


Figure 2.13: Example of CWT.

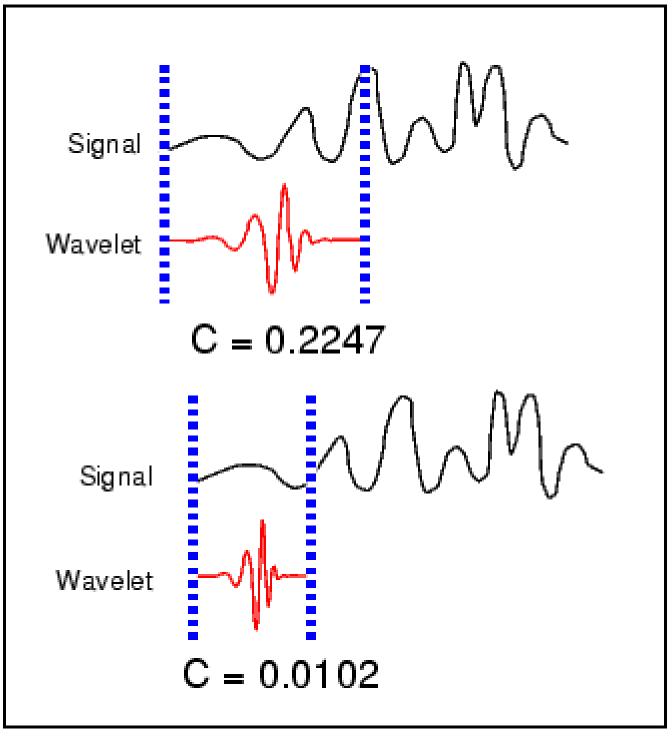


Figure 2.14: Coefficient of CWT [44].

DWT uses the multiresolution analysis based on multirate filter banks. The multiresolution theory was developed by Mallat in 1988 [44]. While the translation of CWT is carried out in a smooth continuous fashion, DWT is in discrete steps. In DWT, the 'original' signal S passes through two complementary filters and emerges as two signals. One signal contains 'approximations' A, in high scale, low frequency components of the signal, and another, 'details' D, in low scale, high frequency components. The filtering process, at its most basic level, looks like Figure 2.15.

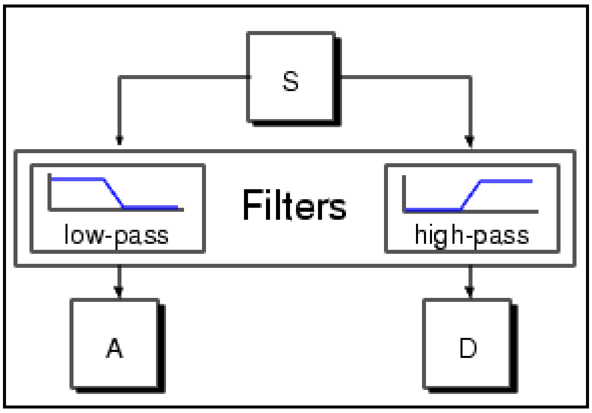


Figure 2.15: DWT basic filtering process [44].

If this operation is actually done with a real digital signal, in the end there is twice as much data as at the start. Suppose, for instance, that the original signal S consists of

1000 samples of data, then the resulting signals will each have 1000 samples, for a total of 2000, as shown in Figure 2.16 (a).

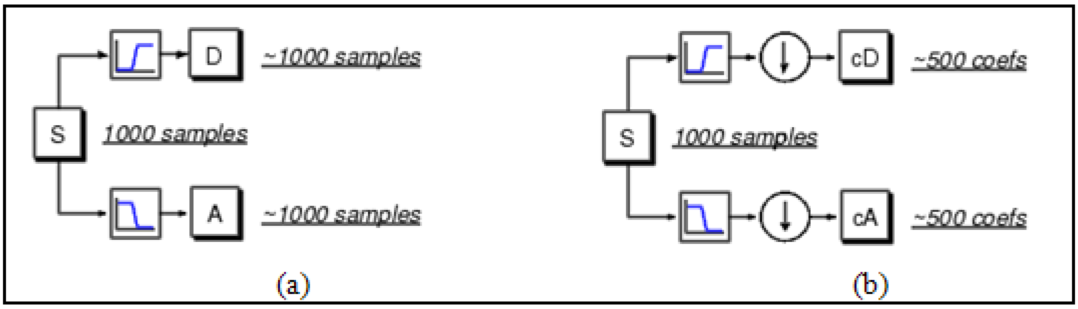


Figure 2.16: Analysis filter bank (a) without down-sampling, (b) with down-sampling by factor of two [44].

In order to avoid this, down-sampling the filtered sequences by a factor of two is done, as shown in Figure 2.16 (b).

The decomposition process can be iterated, with successive approximations being decomposed in turn, so that one signal is broken down into many lower resolution components, as shown in Figure 2.17.

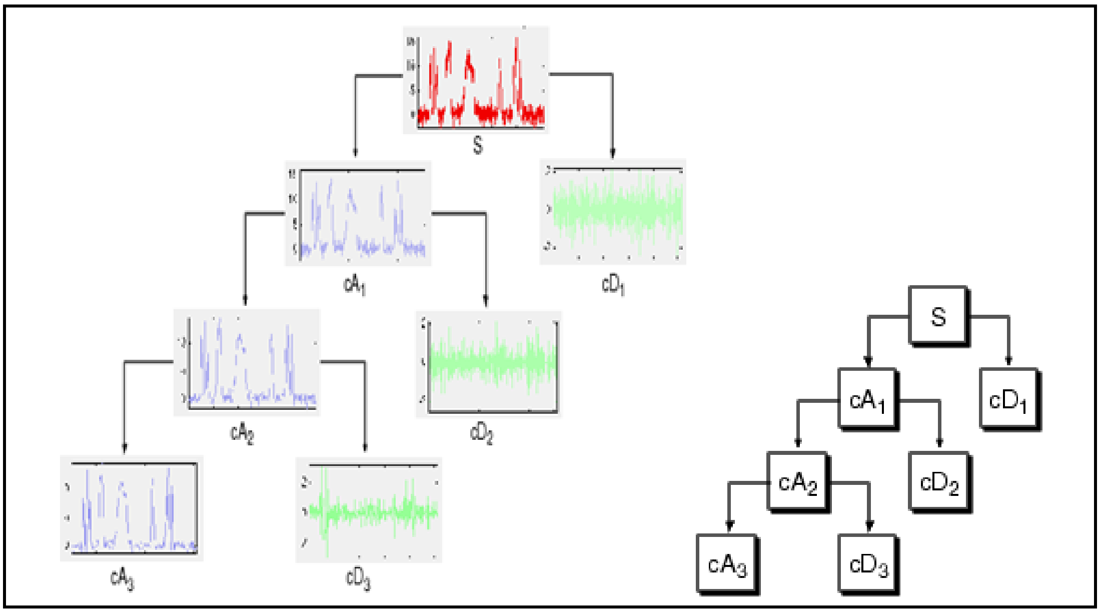


Figure 2.17: DWT multiple-level decomposition tree [44].

In DWT, for many signals, the low frequency content is the most important part as it gives the signal its identity. The high frequency content, on the other hand, adds the characteristics or details of the signal. Hence, in Multiresolution analysis, the low pass filters are used to extract the dominant component of low frequency content. Also in DWT, the term ‘approximations’ and ‘details’ are referred to as the components that can be used to reconstruct the signal. The approximations are the higher scale that is the low frequency components of the signal; the details are the low scale that is the high frequency components.

Mean absolute value (MAV) and root mean square (RMS) are two well-known time domain features [36]. Theoretically, when a signal is modelled as a Gaussian random process, RMS provides the maximum probability estimation of amplitude in a constant force and non-fatiguing contraction. RMS is the better fit at high level of contraction and MAV is well fit for low contractions and fatigued muscle [36].

Some researchers [28, 31] used MAV, VAR, WL and WAMP of time domain features to represent their ‘original’ signal and import to neural network to classify 12 output patterns and 5 output patterns, respectively. A control system for powered upper-limb prostheses has been devised [47] using time domain features, which are MAV, MAV slope and WL, and a simple multilayer perceptron artificial neural network as a classifier. This controller identified four types of muscular contraction using signals measured from the biceps and triceps.

STFT frequency domain and DWT time-scale domain techniques have also been used [48] to extract raw sEMG signal features and import to neural network to classify six output patterns. However, it is not reported which features were selected.

Another research work [42] shows that feature sets based upon the Short Time Fourier Transform (frequency domain) and the wavelet transform provide an effective representation for classification, provided that they are subject to an appropriate form of dimensionality reduction. It also mentions that, given that transient MES patterns have structure in both time and frequency, it is suggested that the signal energy which would discriminate amongst contraction types would be best concentrated in a dual representation.

2.3.3 Pattern Recognition

The field of pattern recognition is concerned with the automatic discovery of regularities in data through the use of computer algorithms and with the use of these regularities to take action such as classifying the data into different categories [49]. Pattern recognition aims to classify data or patterns based either on *a priori* knowledge or on statistical information extracted from the patterns. The patterns to be classified are usually groups of measurements or observations, defining points in an appropriate multi-dimensional space. This is in contrast to pattern matching, where the pattern is rigidly specified. It is the last stage experiment of sEMG control input signal, as shown in Figure 2.18. The selected features of the sEMG signal will be imported to one of the techniques of pattern recognition for training and classification.

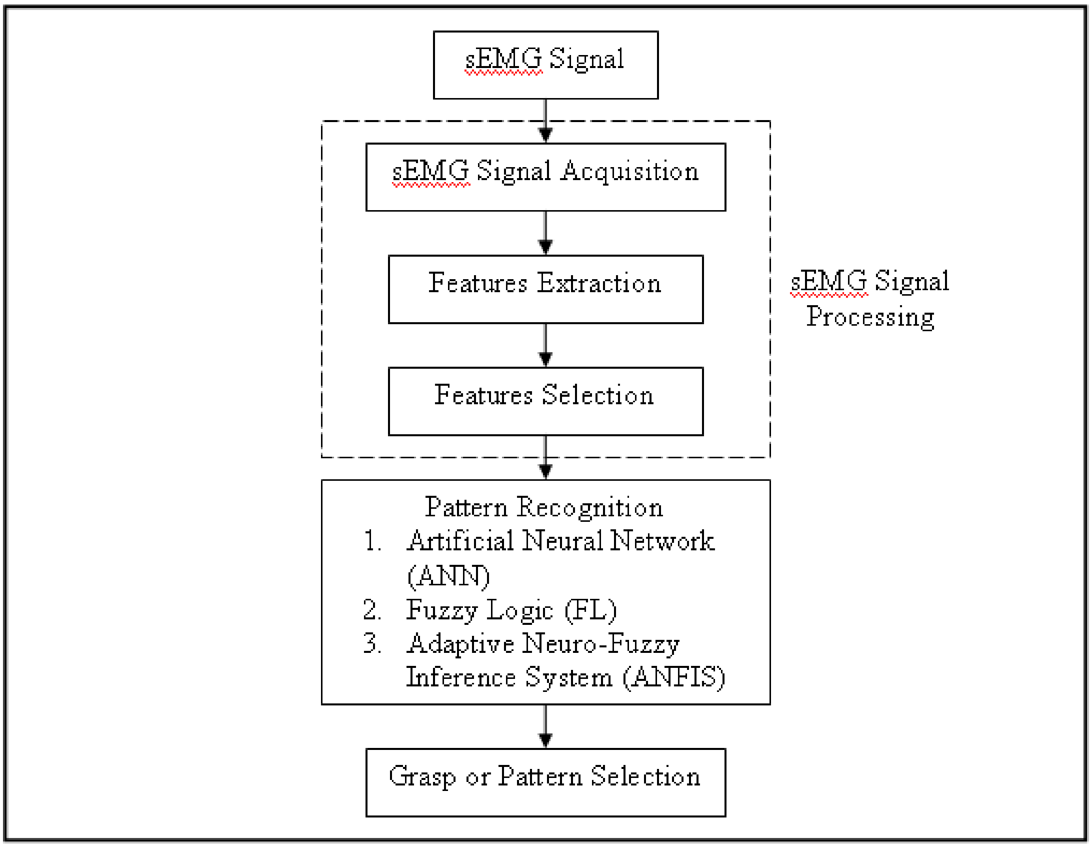


Figure 2.18: Grasp or pattern recognition of raw sEMG signal.

To control more complex hands requires a more advanced human-machine interface involving multiple sEMG electrodes and a suitable pattern recognition system to interpret the raw signal. sEMG pattern recognition has been applied in controlling prosthetic devices for amputees [50]. The main prosthesis control functions of interest were flexion and extension of the forearm and pronation and supination of the wrist, respectively. The hand actions, such as hand opening and grasping, and finger bending were also of some interest.

Many researchers have used different techniques to recognise sEMG, such as artificial neural networks (ANN), fuzzy logics (FL) and adaptive neuro-fuzzy inference system (ANFIS) [48, 51].

2.3.3.1 Fuzzy Logic

Fuzzy logic (FL) using if-then rules provides a mathematical framework that allows to model the uncertainties associated with approximate reasoning, especially for the control systems where mathematical models are difficult to derive, including human perceptual and information processing. The fuzzy system, initially fuzzifies inputs to values at interval [0,1] using a set of membership functions (MF) [38].

Some researchers [50, 52-54] have used FL in three pattern recognition of myoelectric signals for a hand prosthesis. The fuzzy logic controller consists of three modules: fuzzification / defuzzification module, fuzzy inference engine and fuzzy rule base that allows the process of uncertainty in the human thinking. Such information can be expressed with imprecise linguistic forms such as ‘low’, ‘medium’ and ‘high’. In a fuzzy system, the subjective judgments can be formulated through the fuzzification module that transforms them into appropriate fuzzy linguistic variables characterized by membership function in a specified universe of discourse.

FL pattern recognition techniques have also been used to train three output patterns, with a correctly classified rate above 85% [52]. Six myoelectrodes collected sEMG raw signal and the input features to the FL were 63Hz, 125Hz, and 250Hz bands of Fast Fourier Transform (FFT). These three output patterns were palmar 3-finger grasp, lateral grasp, and hook grasp, as shown in Figure 2.19 (a), (c) and (d).

In a research work [54] the sEMG similarities and differences have been quantified for six major grasping patterns of the human hand, which are cylindrical, hook, palmar, lateral, spherical, and tip, as shown in Figure 2.19 (a) to (f). They used twenty needle myo-electrode sensor inserts in ten extrinsic hand muscles to collect myoelectric raw signals of six patterns. Fuzzy c-means (FCM) were used in assessing the extent to which the grasping patterns overlapped electromyographically. The results show that no grasp is completely distinct from any other in transient or in steady-state and that the EMG patterns of the tip and palmar grasps, in particular, overlap significantly.

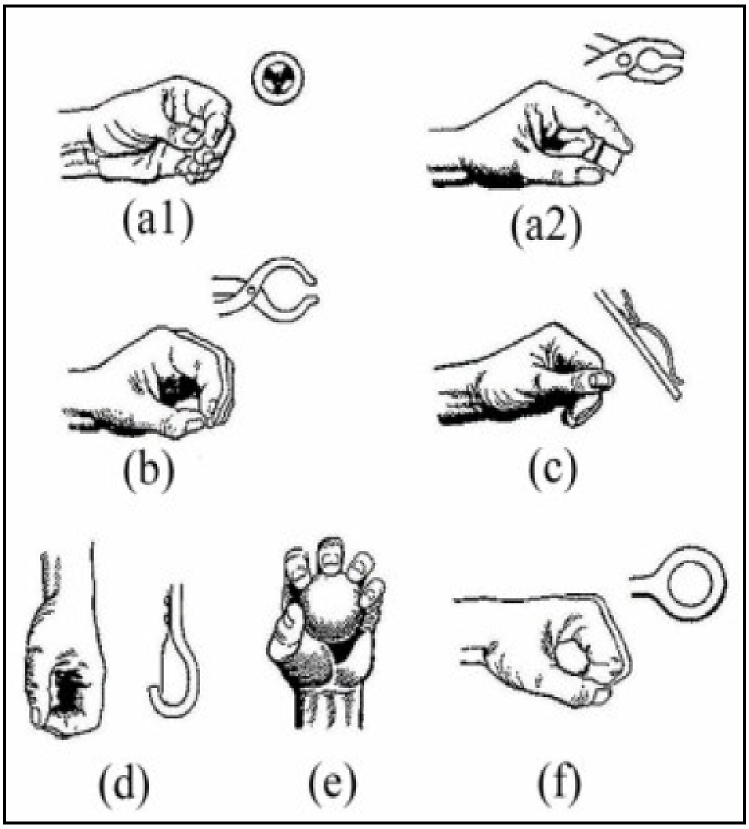


Figure 2.19: Seven major categories of prehensile grasps: a1) palmar 3-finger-thumb opposes index and middle fingers; a2) palmar 2-finger; b) tip-thumb opposes index only; c) lateral; d) hook; e) spherical; f) cylindrical [52, 54].

FL systems are advantageous in biomedical signal processing and classification [50]. Biomedical signals are not always strictly repeatable and may sometimes even be contradictory. Furthermore, using trainable fuzzy systems, it is possible to discover patterns in data which are not easily detected by other methods, as can also be done with neural networks. The significant advantage of FL over ANN is that in FL system it is possible to integrate this incomplete but valuable knowledge into the fuzzy logic system due to the system's reasoning style, which is similar to that of human decision-making and more closely than ANN. It is reported that the fuzzy logic algorithm can have recognition accuracy in the range of 80% to 97.5% [38, 50].

2.3.3.2 Artificial Neural Network

ANN systems are simplified mathematical models of the brain-like systems that function as parallel distributed computing networks which can be trained to learn new associations, functional dependencies and new patterns. ANNs are adaptive, that is they can automatically adjust to modify their behaviour in response to nonlinear dynamics of their environment. Over the last few years, ANNs have been successfully applied in medicine and biomedical studies [55].

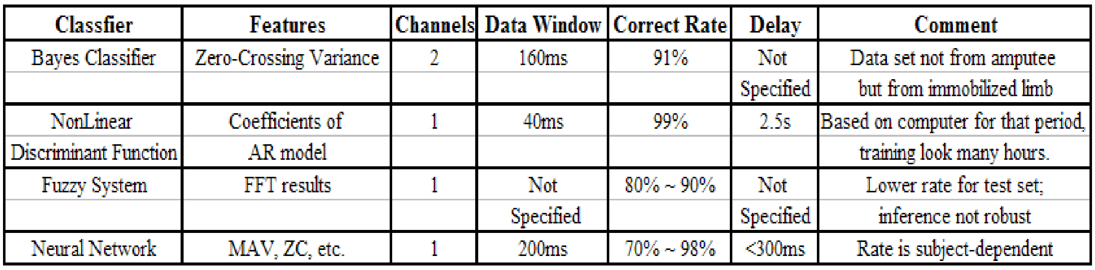
Feedforward multilayer perceptrons with a varying number of hidden layer neurons have been used [28] for classifying 12 output patterns. The authors used a continuous transfer function for the hidden layer neurons, tan-sigmoid, and a logistic function for the output layer, log-sigmoid. Their classifier's overall accuracy (averaged across the 12 movement types) in decoding the individuated and combined finger movements of the six subjects is around 80%.

Another researcher [31] described the features extracted that would serve as input to an ANN. A simple feedforward network was used due to its success in traditional classification and simple regression application. The neural network was constructed with a single layer of tan-sigmoid hidden layer neurons and a layer of tan-sigmoid output neurons. This network structure was chosen to provide an output vector consisting of the approximate position of the MCP joint for each finger. Their classifier's overall accuracy (averaged across the 5 movement types) is above 90%.

In another research work [48] a three (input, hidden and output) layers network was used with back propagation learning to train up to six output patterns. These six output patterns are large and small cylindrical, large and small spherical, pinch and key as illustrated in Figure 2.19 (f), (e), (c) and (a2). The mean values of the success rates are around 75-80%.

Table 2.1 shows the typical EMG classification systems. ANN has the second highest correct rate compared to nonlinear discriminant, however, it has the fastest control [50] as its delay is less than 300ms.

Table 2.1: Typical EMG classification systems [50].



Classfier	Features	Channels	Data Window	Correct Rate	Delay	Comment
Bayes Classifier	Zero-Crossing Variance	2	160ms	91%	Not Specified	Data set not from amputee but from immobilized limb
NonLinear Discriminant Function	Coefficients of AR model	1	40ms	99%	2.5s	Based on computer for that period, training look many hours.
Fuzzy System	FFT results	1	Not Specified	80% ~ 90%	Not Specified	Lower rate for test set; inference not robust
Neural Network	MAV, ZC, etc.	1	200ms	70% ~ 98%	<300ms	Rate is subject-dependent

2.3.3.3 Adaptive Neuro-Fuzzy Inference System

While fuzzy logic allows for an inference mechanism under uncertainty, ANN affords system learning, adaptation, parallelism of information processing and generalisation over time. A hybrid system, called adaptive neuro-fuzzy inference system, combines the concepts of fuzzy logic and ANN . It uses a given input/output data set to construct a fuzzy inference system (FIS) whose membership function parameters are tuned or adjusted, using either a backpropagation algorithm alone or in combination with a least squares type of method. This adjustment allows the fuzzy system to learn from the data modelled [56].

A research work [38] proposes a hybrid method for training a fuzzy system consisting of backpropagation (BP) and least mean square (LMS). The authors extract four features of the raw sEMG signal in order to train six patterns, which are hand opening and closing, wrist radial flexion and extension, pinch (Figure 2.19 (b)) and thumb flexion, based on two channels of differential surface electrodes. The average system accuracy for the combined approach is 96%. They described that the difference between the FL and ANFIS is that FL is adjusting membership function parameters in order to minimise the output of each fuzzy rule and estimating the number of rules. The ANFIS adapts the parameters of a Sugeno type inference system using neural network [56]. For training the fuzzy system, ANFIS employs backpropagation for the parameters associated with the input membership functions and LMS estimation for those associated with the output membership functions. Figure 2.20 shows the basic ANFIS structure.

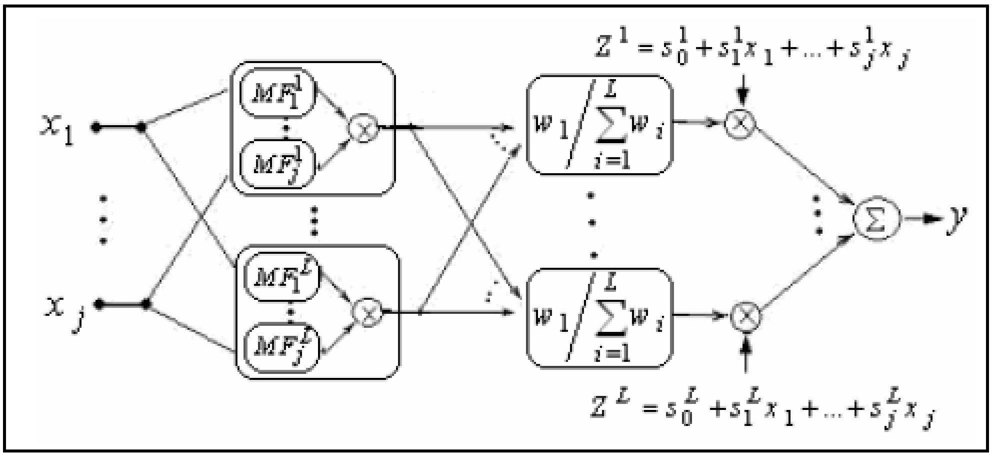

$Z^1 = s_0^1 + s_1^1 x_1 + ... + s_j^1 x_j$
x_1
x_j
MF_1^1
MF_j^1
MF_1^L
MF_j^L
$w_1 / \sum_{i=1}^{L} w_i$
$w_1 / \sum_{i=1}^{L} w_i$
Σ
y
$Z^L = s_0^L + s_1^L x_1 + ... + s_j^L x_j$

Figure 2.20: Basic adaptive neuro-fuzzy inference system structure [38, 56].

Although the ANFIS system can have a high rate of pattern recognition, it is reported [53] that fuzzy logic combined with more complex algorithms, such as ANN, has slow update rates and has not been assessed on individuals who have experienced limb-loss.

2.4 Summary and Contribution of this Project

The prosthesis as a tool makes no pretence of trying to replace the lost limb's physiological appearance. In fact, it works as an aid to help provide some of the functions that were lost due to accidents, war or congenital conditions. Moreover, the prosthesis is an interchangeable device that can be used only when needed. Much effort in the field of upper-extremity prosthesis research is directed towards the development of prostheses as true limb replacements.

In reality, most patients or amputees feel uncomfortable when using the current commercial prosthetic hands, because they are usually heavy, are unable to provide enough grasping functionality and lack degree of freedom [3, 4]. Commercially available devices, such as Otto Bock Sensor Hand [1, 4, 5] and Touch Bionic's i-Limb are far from providing the manipulation capabilities of the human hand. For example, their active bending is restricted to two or three joints, which are actuated by a single heavy motor drive acting simultaneously on the Metacarpophalangeal (MCP) joint of the thumb, index and middle finger, while other joints can bend only passively or can not bend at all. The most commonly used myoelectric hand has one degree of freedom, opening and closing. It uses only two myoelectrode sensors in order to increase training efficiency and practical reliability of the device. A number of researchers [26, 28-31] have suggested that if there is a need to increase the functionality of prosthetic devices, to have fine control, and that if the patients must be able to enforce the correct grasping type, more than two myoelectrodes sensors can be added, however at the expense of training efficiency.

Consequently, the motivation for this research is to investigate the use of a hybrid actuation mechanism in the design and development of an intelligent prosthetic hand. This work highlights user-friendliness in usage and involves a proper mechanical design in more active degree of freedom incorporating an intelligent control system. Besides, based on the Section 2.2.1, we are going to use both SMA and DC actuators to form the hybrid actuation mechanism. The reasons for using SMA are its operational similarity to human muscles [8], its incredible small size and volume, its high force to weight ratio and its low cost [3, 19], and the reasons for using the small size and light in weight DC actuator are its high torque and easiness to control.

The motivation for this research was turned into the aim of this project and will be associated with the objectives described in the Chapter 1.

The electrical activity generated by contracting muscle in forearm will be used as a control input signal for the prosthetic fingers. The electrical activity of the muscle, the myoelectric signal (Section 2.3), is monitored by using sEMG sensors (myoelectrodes). Only two myoelectrodes will be used in order to make the system more user-friendly for the amputees or users to easily train and use (Section 2.3.1). The sEMG signal contains valuable information in a particularly useless form. This information is useful only if it can be quantified by a signal processing (Section 2.3.2) method to achieve the accurate and actual sEMG signal that we would like to use.

Pattern recognition aims to classify data or patterns based either on *a priori* knowledge or on statistical information extracted from the patterns. The patterns to be classified are usually groups of measurements or observations, defining points in an appropriate multi-dimensional space (Section 2.3.3). This is in contrast to pattern matching, where the pattern is rigidly specified. Hence, the control system of the sEMG signal will be considered using pattern classification technique to train and test the varying patterns of human fingers’ movement (thumb not included).

The main contribution of this research work can be summarised as:

1. Work in the area of a hybrid actuation mechanism for powered prosthetic hands to provide more dexterity.

2. An extension of research in the area of signal processing and ANN when only two myoelectrodes are used to increase the number of extracted features to control more movement in a prosthetic hand.

CHAPTER 3
Design of a Prosthetic Finger

3.1 Introduction

This chapter turns to discussing the preliminary finger design and development to allow the hybrid actuation mechanism to increase the number of active DOF to provide the functionality of a real hand in order to mimic movements such as coarse and fine grasping. The chapter covers just the mechanical design of the finger, such as the size, shape, joint and degrees of freedom. Other characteristics, such as force and torque, follow in the next chapter.

To avoid complexity in hand design and control, a prosthetic finger design is proposed and rigorously studied. It is postulated that the hand design and control would be similar to that of a single prosthetic finger. In order to design a proper prosthetic finger to mimic the human finger, the human upper limb anatomy and movement are investigated in Section 3.2. This gives an insight into finger geometry and movement range of each finger segment. After human anatomy, section 3.3 presents the mechanical design of the three DOF prosthetic finger. Due to large variations in the sizes of human hands and fingers, the author has decided to use his own middle finger as the design model. Section 3.4 explains the structure strength in relation to the material used in rapid prototyping so that the material used in the prosthetic finger will not fracture in grasping an object. Section 3.5 covers the forward and inverse kinematics of the designed finger.

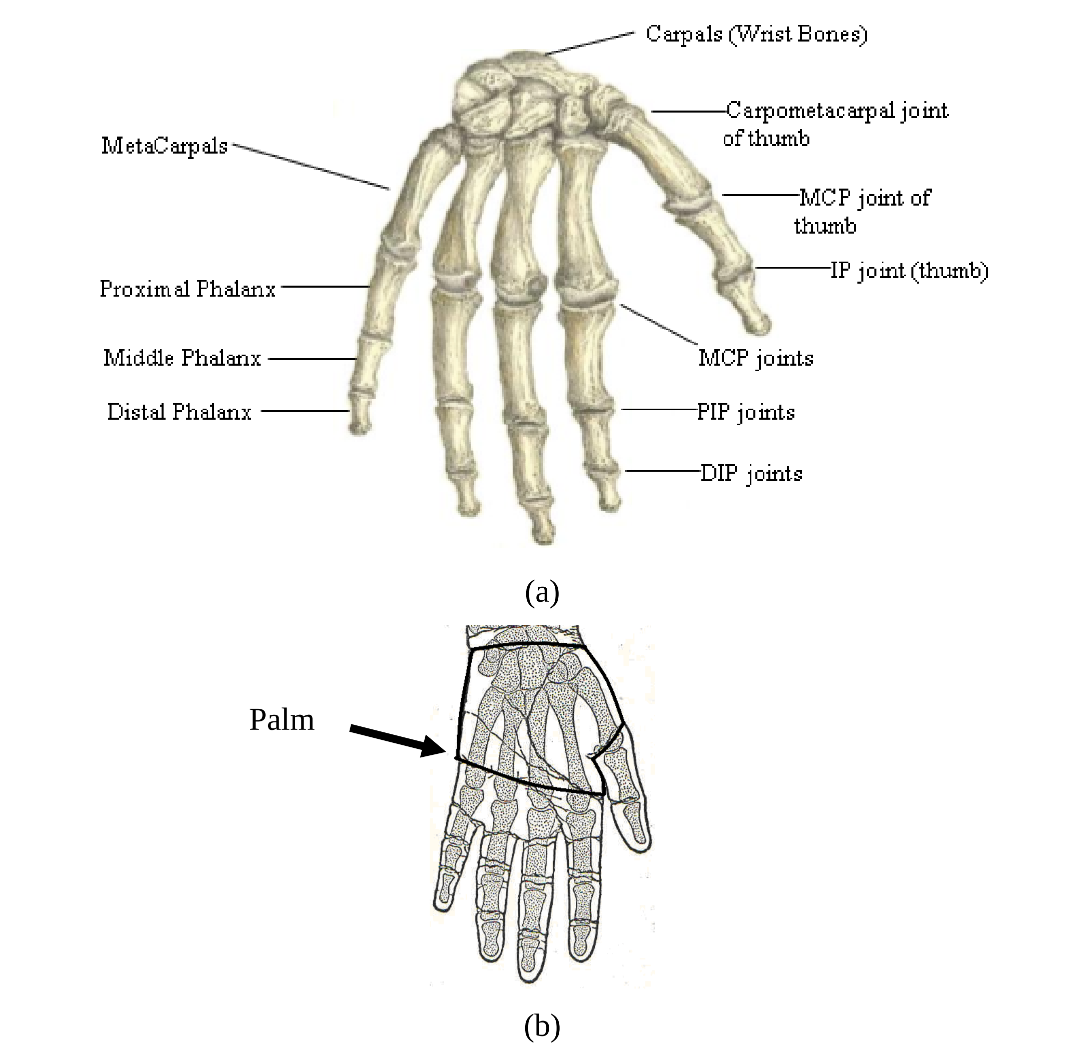
3.2 Hand and Finger Anatomy

This section presents a brief overview of human hand anatomy and a description of the joint articulation during movement in order to design a biomechanically correct prosthetic finger. The force and torque of the hand/finger are discussed in the following chapter.

Figure 3.1 shows an illustration of hand anatomy. The human hand consists of a broad palm (metacarpals and carpals) with five digits (thumb and four fingers: index, middle, ring, and little finger), attached to the forearm by a wrist joint. The thumb is located on the lateral side of the palm and is connected to the carpals.

Each finger consists of three bones or phalanges (proximal, PP; middle, MP; distal, DP phalanxes) and three joints (metacarpophalangeal MCP, proximal interphalangeal PIP and distal interphalangeal DIP)

The MCP joint has 2 DOF, abduction-adduction and flexion-extension. The PIP and DIP joints have 1 DOF, flexion-extension. The MCP joint abduction-adduction angle is around 20^0 and flexion-extension is 90^0. The flexion-extension movement of the PIP and DIP joints is 100^0 ~ 110^0 and 80^0 ~ 90^0, respectively [6, 8, 15, 57].



(a)

(b)

Figure 3.1: Anatomy of human left hand (a) Bones and joints [57], (b) Palm.

3.3 Finger Design

The general overall design concept for the prosthetic finger presented here is described in mechanical terms and related to human anatomy. The finger prototype is used to allow the hybrid actuation mechanism to provide some functionality of a real hand, such as coarse and fine grasping. To avoid complexity in hand design and control, a prosthetic finger design is proposed and rigorously studied. It is postulated that the hand design and control would be similar to that of a single prosthetic finger.

The ultimate goal is to develop a prosthetic finger for an adult. Hence, it is designed to replicate as closely as possible the size of an adult's finger. The proposed finger design is a three DOF hybrid actuation mechanism, which mimics the kinematics and functionality of the human finger. Its force/torque will be presented in the next chapter.

3.3.1 Dimensions

The dimensions of the design in Figure 3.2 and Table 3.1 are based on the size of the author's middle finger and it is custom-made as a result of large variations in the size of human hands and fingers. The abduction-adduction movement of the MCP joint is just 20^0 and does not contribute significantly to its functionality. For this reason, it is ignored, leaving the design of the finger with 3 DOF (1 DOF at each of the joints).

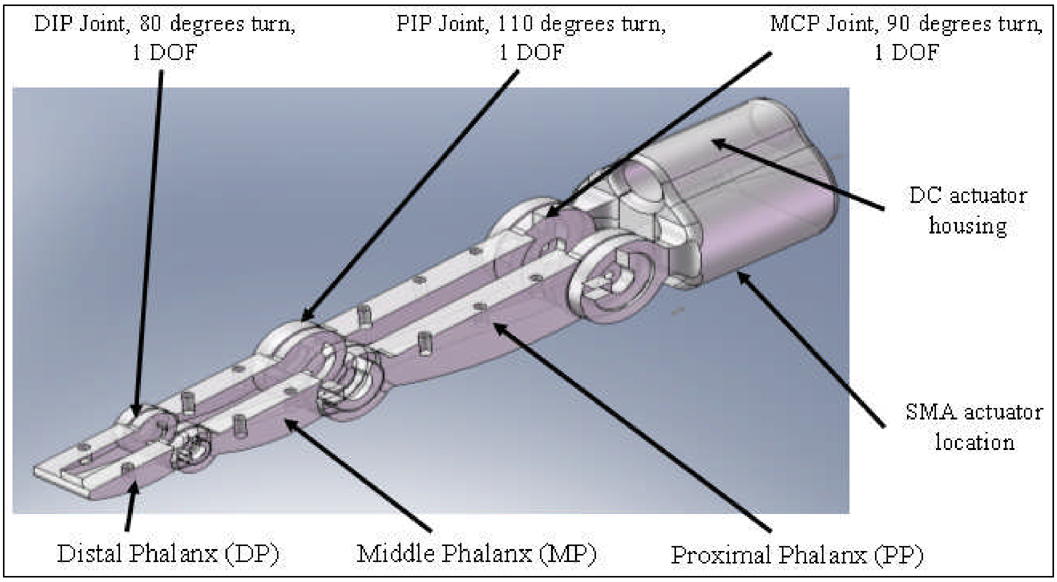
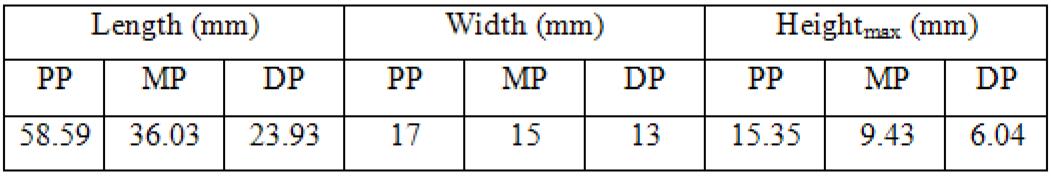

DIP Joint, 80 degrees turn, 1 DOF
PIP Joint, 110 degrees turn, 1 DOF
MCP Joint, 90 degrees turn, 1 DOF
DC actuator housing
SMA actuator location
Distal Phalanx (DP)
Middle Phalanx (MP)
Proximal Phalanx (PP)

Figure 3.2: Prosthetic middle finger design.

Table 3.1: Dimensions of middle finger phalanges.



Length (mm)			Width (mm)			$\text{Height}_{\text{max}}$ (mm)		
PP	MP	DP	PP	MP	DP	PP	MP	DP
58.59	36.03	23.93	17	15	13	15.35	9.43	6.04

Lengths of the PP, MP and DP are 58.59mm, 36.03mm and 23.93mm, respectively. Hence, the total length of the middle finger is 118.55mm. The width of each phalange is 17mm, 15mm and 13mm, respectively. The length of the finger's palm housing, in which is to place DC and SMA actuators, is 33mm, the height is 29.50 mm and the width 30mm (see Figure 3.3). The central point of MCP joint to the end of the finger's palm housing is 47.37mm. Consequently, the total length of the middle finger and the finger's palm housing is 165.92mm. It is slightly shorter than the author's middle finger (including palm). The total length of the author's middle finger (including palm) is 182mm. Therefore, 16.08mm length can still be add to the finger's palm housing. Further discussion of this 16.08mm will be in Chapter 4.

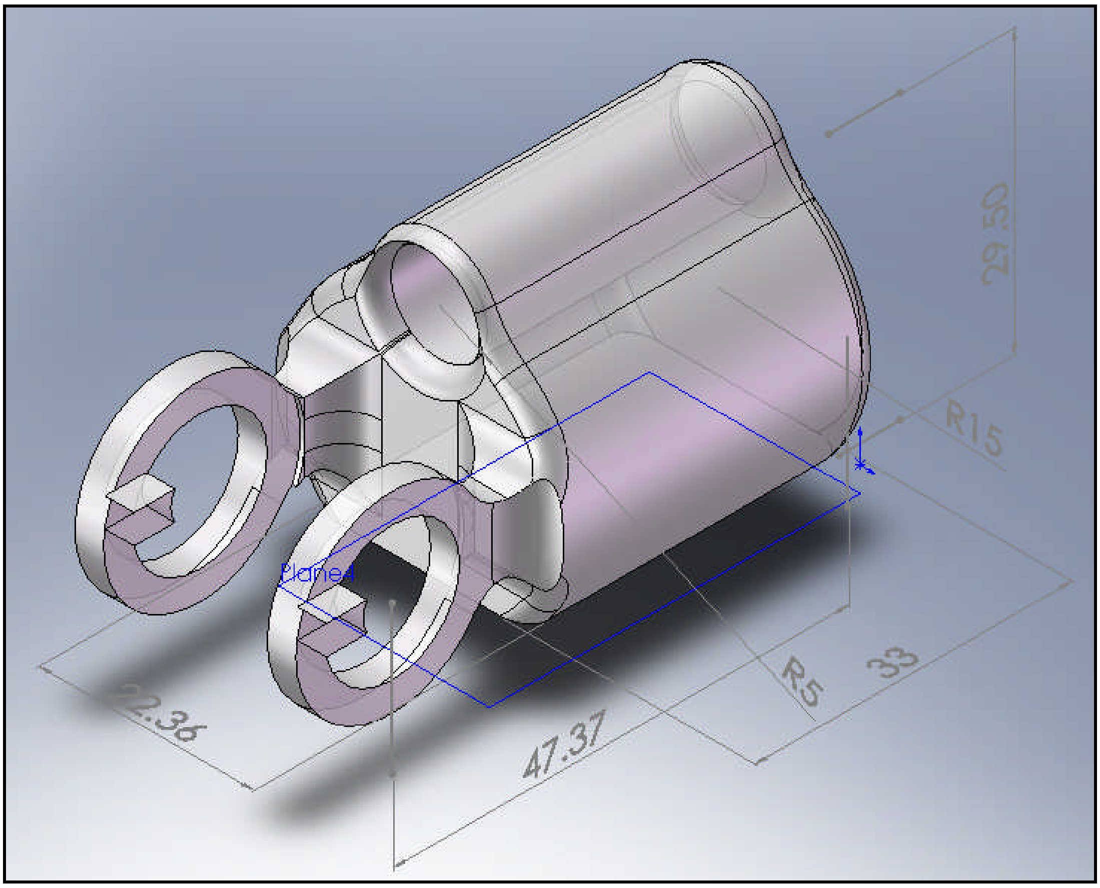


Figure 3.3: Finger's palm housing design (in mm)

3.3.2 Joints and Degrees of Freedom

In this design, a total of three revolute joints are used, connected in each joint or each DOF, as shown in Figure 3.4. Each phalanx consists of one head and one tail. The fixed axis (with a stopper) of rotation in each phalanx is called the head and each rotational axis is called the tail. The head of the finger's palm is connected to the tail of the proximal phalanx (PP), forming the MCP joint (see Figure 3.4), which provides 90^0 of movement, i.e. flexion-extension. The head of the PP is connected to the tail of the middle phalanx (MP), forming the PIP joint, which provides 110^0, and the head of the MP to the tail of the distal phalanx (DP), forming the DIP joint, which provides 80^0 of flexion-extension. The degree of angular movement for the PIP and DIP joints are selected based on the largest and smallest values of the range in normal human finger, as reported on page 38. These two values are also achieved from the author's middle finger movement measurements.

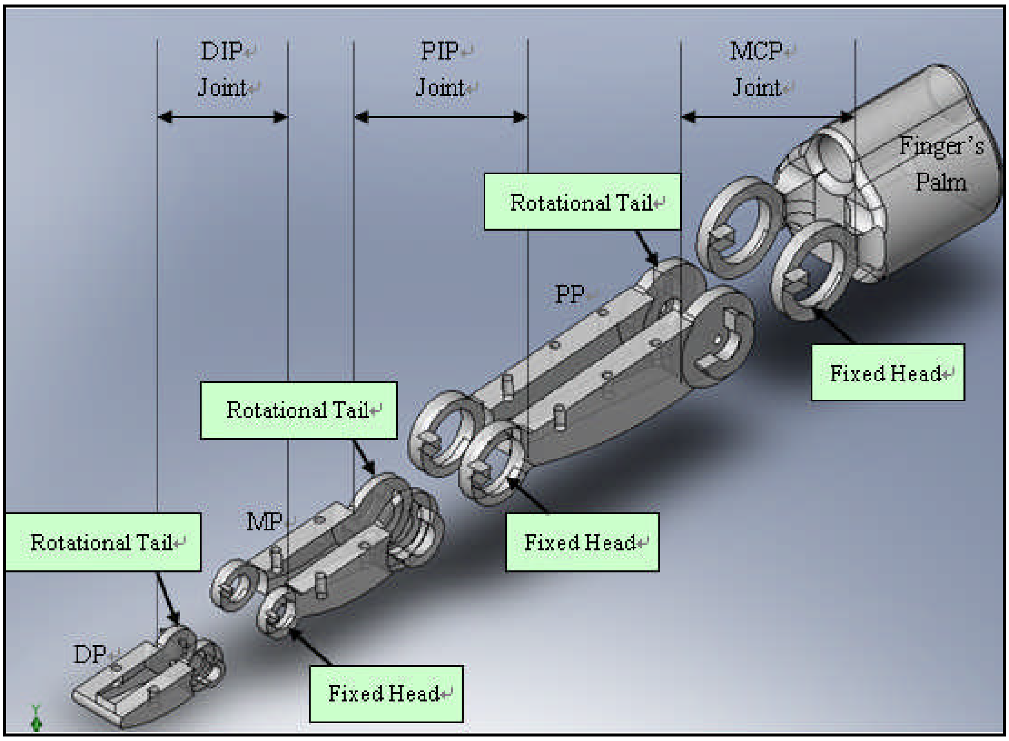


Figure 3.4: Finger joint.

3.3.3 Material Selection

Material selection is very important in order to design a safe grasping functionality of the hand/finger. Before the finger design is manufactured by rapid prototype machine, some relevant calculation is needed, such as the stopper on the fixed head (see Figure 3.4), and the contact surface between the finger for the palm housing and the finger palm housing fixed head. According to [58], the Maximum Voluntary Contraction (MVC) force in the proximal phalanx of the middle finger is around 40N for the 1-finger tasks of adult. Hence, this 40N will be used as reference in choosing a suitable material for the prosthetic finger design.

The finger, shown in Figure 3.2, was manufactured using a Vanguard HS HiQ SLS (Selective Laser Sintering) rapid prototype machine.

Firstly, the author decided to use the material called Duraform PA plastic, which is the Nylon 12 material, to construct the finger. The tensile (Young's) modulus and the specific gravity of this material are 1586MPa and 1.00g/cm^3, respectively [59]. According to Equation 3.1, and some relevant values shown in the Tables 3.2 and 3.3, we know that the displacement $\Delta L_{Stopper}$, on the stopper of the fixed head, and the displacement, $\Delta L_{Surface}$, on the surface between the palm and the palm's fixed head, are 0.2113 mm and 0.0192 mm, respectively.

Table 3.2: Relevant values to evaluate displacement of stopper.

Value for counting the displacement, $\Delta L_{Stopper}$, of the stopper.	
Torque transmission from DC actuator to the stopper, FL_0	4.2624Nm*
Original cross-section area, A_0	1.272×10^{-5} m^2

(* DC motor stall torque (1.48mNm) x DC actuator gear head (1:64) x Number of teeth of work gear (45)).

Table 3.3: Relevant values to evaluate displacement of surface between finger palm housing and palm fixed head.

Value for counting displacement $\Delta L_{Surface}$ of surface between palm and palm fixed head.	
Original length L_0, for PP to the area	0.07296 m
Maximum force F applied to area	40N **
Original cross-section area A_0	9.584×10^{-5} m^2

(* Maximum Voluntary Contraction (MVC) force in PP of middle finger = 39.69 ± 7.16 N in 1-finger tasks of Young [58]).

$$E = \frac{\text{Stress}}{\text{Strain}} = \frac{\sigma}{\varepsilon} = \frac{\frac{F}{A_0}}{\frac{\Delta L}{L_0}} = \frac{FL_0}{A_0 \Delta L} \quad \text{(Equation 3-1)}$$

<u>where</u>: E = Tensile (Young's) Modulus,

F = Force applied to object,

A_0 = Original cross-section area through which force applied,

ΔL = Amount by which length of object changes,

L_0 = Original length of object.

Since the displacement on the stopper $\Delta L_{Stopper}$, was considered a very large value, the second material was changed to Duraform GF plastic, which is a glass-filled nylon based on nylon 12. The tensile (Young's) modulus is 4068MPa, and its specific gravity is 1.49g/cm^3 [60]. Consequently, this material was make the displacement on the stopper $\Delta L_{Stopper}$ and the displacement on the surface between the palm and the palm's fixed head $\Delta L_{Surface}$, became 0.0823 mm and 0.00749 mm.

Nylon was preferred over aluminium because it has a lower friction coefficient, 0.15~0.25 and 1.05~1.34, respectively. An additional drawback of using aluminium-based bearing designs would increase the overall complexity of the system without any benefits.

3.4 Prosthetic Finger Kinematics

The kinematic model of the hand is designed to form the basis of a controller rather than to be a comprehensive anatomical model. A compromise is needed between model accuracy and computational complexity, particularly regarding the inverse kinematics solution.

The kinematics model plays an important role in the work as it provides the geometric constraints on the x-y plane positions of hand features. Using the D-H (Denavit-Hartenberg) representation, we start by solving the forward kinematic problem where individual link transforms are concatenated to yield a single transform that fully describes the position and orientation of the fingertip with respect to the base reference frame (finger palm housing). After that, the inverse kinematics equations are analysed from the quadrilateral.

The purpose of the finger kinematics analysis is to determine the final fingertip position relationship between the angular position of each joint. The finger design can be modelled as a robot hand fixed to the finger's palm housing, its schematic of a robot hand lying in the X-Y plane based on a prosthetic finger is shown in Figure 3-5.

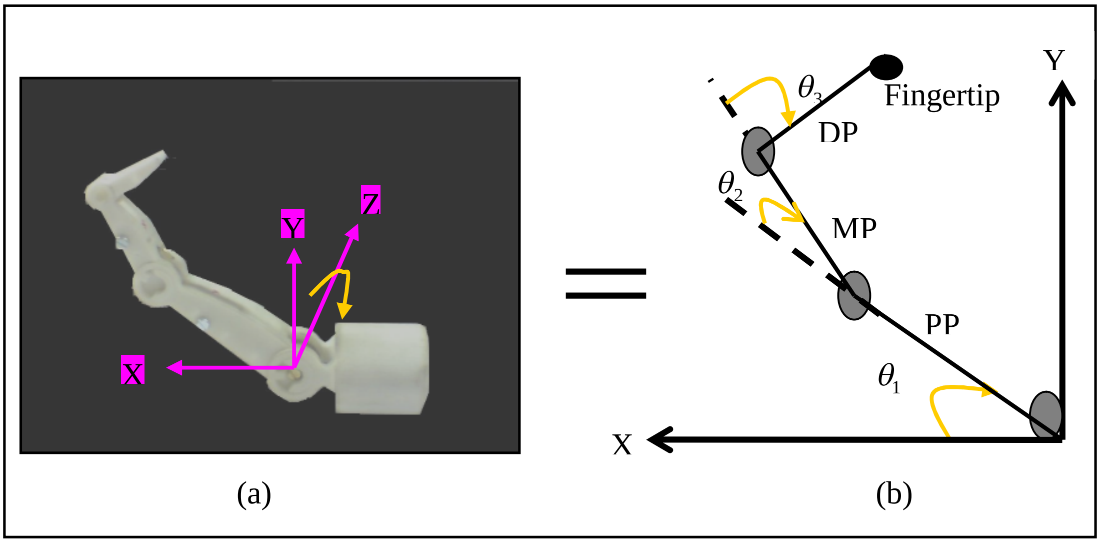


Figure 3.5: (a) Prototype of prosthetic finger, (b) Schematic of prosthetic finger lying in X-Y plane.

Kinematics is the study of the motion of robots. In a kinematic analysis, the position, velocity and acceleration of all the links are calculated without considering the forces causing this motion. The relationship between motion and associated forces and torque is studied in robot dynamics. For the kinematic analysis, there are two fundamental questions for the finger kinematics to evaluate the fingertip position [61]:

1. Given the desired position and orientation of the end-effectors of the manipulator and the geometric link parameters with respect to a reference coordinate system, one can calculate the desired joints angles. This fundamental problem is called inverse kinematics.

2. The second fundamental problem contrasts with the first. Given each joint angles and the geometric link parameters, one can find the position and ori-

entation of the end-effectors of the manipulator with respect to a reference coordinate system. This fundamental problem is called forward kinematics.

A simple block diagram indicates the relationship between these two problems in Figure 3.6.

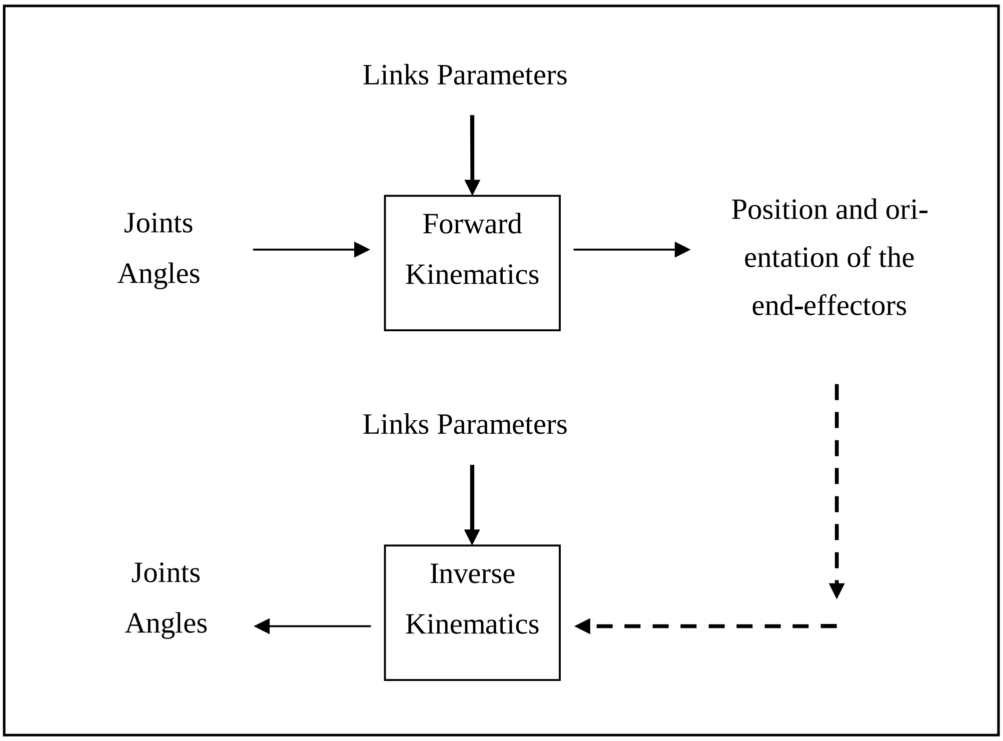


Figure 3.6: Forward and inverse kinematics problems [61].

3.4.1 Forward Kinematics

In our kinematics model, the prosthetic finger consists of three links corresponding to the three phalanges of the human finger (see Figure 3.5). The finger is equipped with two active DOF and one passive. The two active DOF are at the MCP and PIP joints and the one passive DOF is at the DIP joint coupled with the PIP joint.

Denavit-Hartenberg’s (D-H) is the most commonly used method to solve the forward kinematics problem, where the transformations of links connected in series are used to produce a single transformation showing the position or direction of the fingertip with respect to the palm or to the reference base [8, 61]. The finger kinematics can be considered in 2D kinematics and lying in the X-Y plane, as shown in Figure 3.5, because the abduction-adduction movement of the MCP joint was ignored, as previously explained.

The D-H theorem consist of four parameters, these being the joint angle θ_i, the link/phalanx offset d_i, the link/phalanx length L_i, and the link/phalanx twist α_i [61]. These four variables are used to calculate the position or direction of the fingertip.

The abduction-adduction movement of the MCP joint was ignored and makes no movement in the z-axis (Figure 3.5) to cause the link/phalanx offset d_i and the link/phalanx twist α_i can be eliminated. The finger link/phalanx coordinate parameters are shown in Table 3.4.

Table 3.4: Finger joints and links/phalanges coordinate parameters.

Joint i	Twist α_i	Angle θ_i	Length L_i	Offset d_i	Joint Angle Range
1	0	θ_1 (MCP)	58.59 mm (PP)	0	0^0~90^0
2	0	θ_2 (DIP)	36.03 mm (MP)	0	0^0~110^0
3	0	θ_3 (DIP)	23.93 mm (DP)	0	0^0~80^0

Once the D-H coordinate system has been established for each link, a homogeneous transformation matrix can easily be developed relating the *i*th coordinate frame to the (*i*-1)th coordinate frame [61].

$$^{i-1}T_i = T_{z,\theta}T_{z,d}T_{x,l}T_{x,\alpha} \quad \text{(Equation 3.2)}$$

$$^{i-1}T_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & L_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(Equation 3.3)

Since the link/phalanx offset d_i and the link/phalanx twist α_i are zero, then Equation 3.3 can be simplified to Equation 3.4.

$$^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & L_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i & 0 & L_i\sin\theta_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{(Equation 3.4)}$$

$$^{0}T_i = ^{0}T_1 \, ^{1}T_2 \, ^{2}T_3 ..... ^{i-1}T_i \quad \text{(Equation 3.5)}$$

Equation 3.5 is used to connect each individual link to a single transform that shows the position or direction of the fingertip with respect to the palm or reference base. In our case, Equation 3.6 will be derived from Equation 3.5 [61].

$$^{0}T_3 = ^{0}T_1 \, ^{1}T_2 \, ^{2}T_3 \quad \text{(Equation 3.6)}$$

$$^{0}T_3 = \begin{bmatrix} c_{123} & -s_{123} & 0 & L_1c_1 + L_2c_{12} + L_3c_{123} \\ s_{123} & c_{123} & 0 & L_1s_1 + L_2s_{12} + L_3s_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \text{(Equation 3.7)}$$

$$^{0}T_3 = \begin{bmatrix} x_i & y_i & z_i & p_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{(Equation 3.8)}$$

where:

c_{123} = $\cos(\theta_1 + \theta_2 + \theta_3)$

s_{123} = $\sin(\theta_1 + \theta_2 + \theta_3)$

c_{12} = $\cos(\theta_1 + \theta_2)$

s_{12} = $\sin(\theta_1 + \theta_2)$

c_1 = $\cos\theta_1$

s_1 = $\sin\theta_1$

[x*i*, y*i*, z*i*] = Orientation matrix of *i*th coordinate system established at link *i* with respect to base coordinate system. Upper left 3x3 partitioned matrix of $^{0}T_i$ [61].

p*i* = Position vector pointing from origin of base coordinate system to origin of *i*th coordinate system. Upper right 3x1 partitioned matrix of $^{0}T_i$ [61].

Substituting parameter values given in Table 3.4 into Equation 3.4, together with Equation 3.6 to Equation 3.8, the solution for the fingertip position can be found as given in Equation 3.9.

$$P_{X_{FingerTip}} = L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3)$$
$$P_{Y_{FingerTip}} = L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3) \quad \text{(Equation 3.9)}$$
$$P_{Z_{FingerTip}} = 0$$
$$\phi = \theta_1 + \theta_2 + \theta_3$$

<u>where</u>:

$\mathrm{P}_{\mathrm{X}_{\mathrm{Fingertip}}}, \mathrm{P}_{\mathrm{Y}_{\mathrm{Fingertip}}}, \mathrm{P}_{\mathrm{Z}_{\mathrm{Fingertip}}}$ = Position of fingertip in x, y and z co-ordinate, (mm).

L_1, L_2, L_3 = Lengths of 1st link (PP), 2nd link (MP) and 3rd link (DP), (mm).

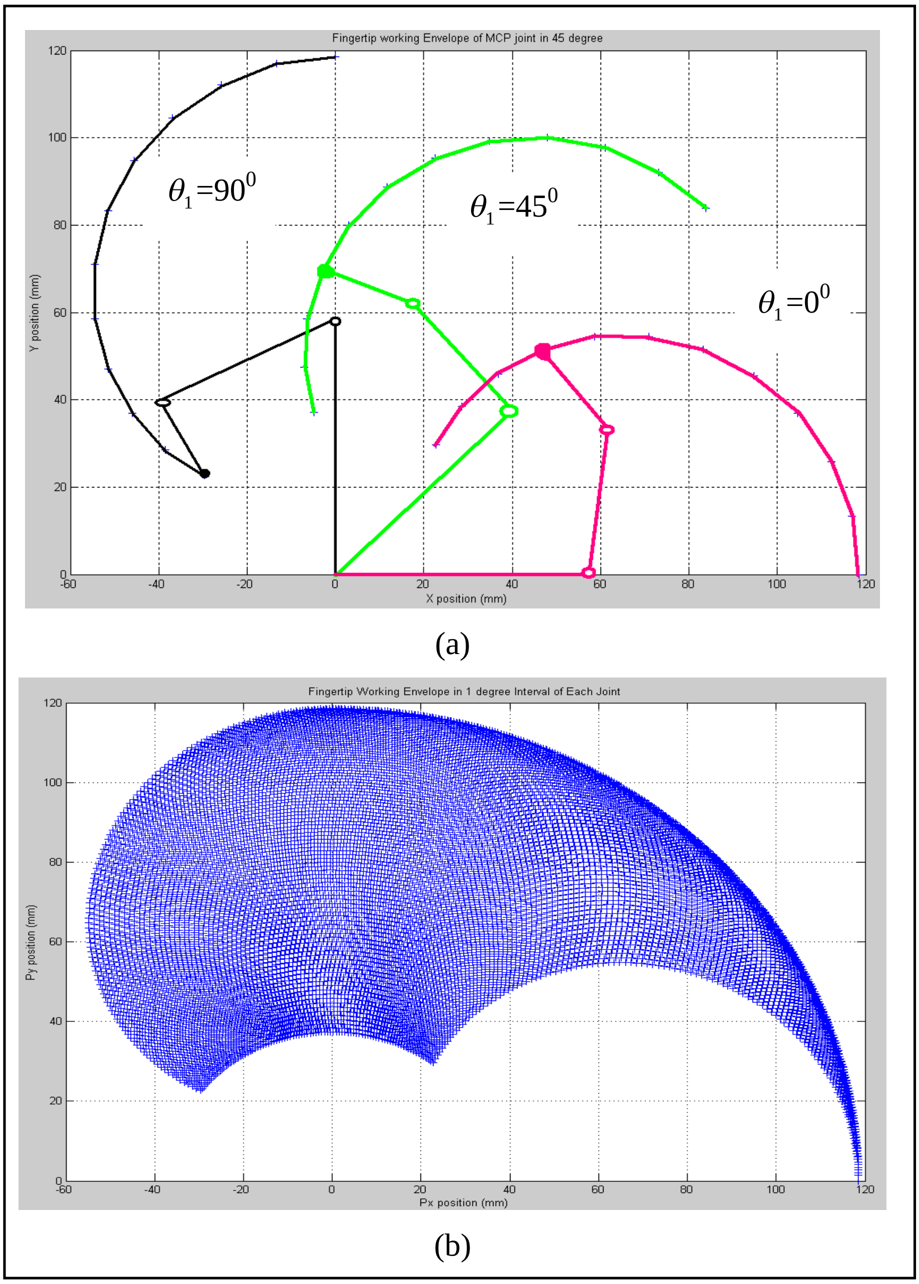
$\theta_1, \theta_2, \theta_3$ = Joint angle of 1st joint (MCP), 2nd joint (PIP), and 3rd joint (DIP), radian

ϕ = Angle between reference or orientation frame and fingertip.

In our finger mechanism design, the angle of the DIP joint, θ_3, is coupled with the PIP joint θ_2, by using 8/11 ratio pulley.

$$\theta_3 = \frac{8}{11}\theta_2 \quad \text{(Equation 3.10)}$$

By Equation 3.9 together with Equation 3.10, the working envelope for the fingertip position can be plotted as in Figure 3.7. The plot shows all the X-Y data points generated by cycling through different combinations of θ_1 and θ_2 (θ_3 was coupled with θ_2) and predicting the fingertip position in x and y co-ordinates for each.



(a)

(b)

Figure 3.7: (a) Fingertip working envelope of MCP joint (θ_1) in 45^0 interval with three finger links, (b) Fingertip working envelope of MCP joint (θ_1) in 1^0 interval.

Figure 3.7(b) illustrates that Y position ranges from $0mm \le P_{Y_{FingerTip}} \le 118.55\text{mm}$, while X position ranges from $-54.80mm \le P_{X_{FingerTip}} \le 118.55\text{mm}$. Tables 3.5 and 3.6 show the corresponding joint angles with respect to minimum and maximum of ($P_{X_{FingerTip}}, P_{Y_{FingerTip}}$). An important conclusion is that the prosthetic fingertip working envelope has the same range of movement as the author's middle finger, upon which the finger design was based (in this research, the small movement of abduction-adduction of the MCP joint is neglected). Therefore, it can be claimed that it provides a level of functionality close to that of an actual human middle finger.

Table 3.5: Min. and Max. of $P_{X_{FingerTip}}$ values.

	$P_{X_{FingerTip}}$	θ_1	θ_2	θ_3
Min.	-54.8mm	90^0	65^0	47^0
Max	118.55	0	0	0

Table 3.6: Min. and Max. of $P_{Y_{FingerTip}}$ values.

	$P_{Y_{FingerTip}}$	θ_1	θ_2	θ_3
Min.	0	0	0	0
Max	118.55mm	90^0	0	0

3.4.2 Inverse Kinematics

Inverse kinematics is the process of determining the parameters of the joint angle in order to achieve a desired position. From the 1st fundamental problem of finger kinematics above, the values of the $P_{X_{FingerTip}}$, $P_{Y_{FingerTip}}$ and $\theta_{Fingertip}$ are given, in order to evaluate the value of the desired angle, such as angle of MCP joint θ_1, PIP joint θ_2, and the DIP joint θ_3, are to be solved.

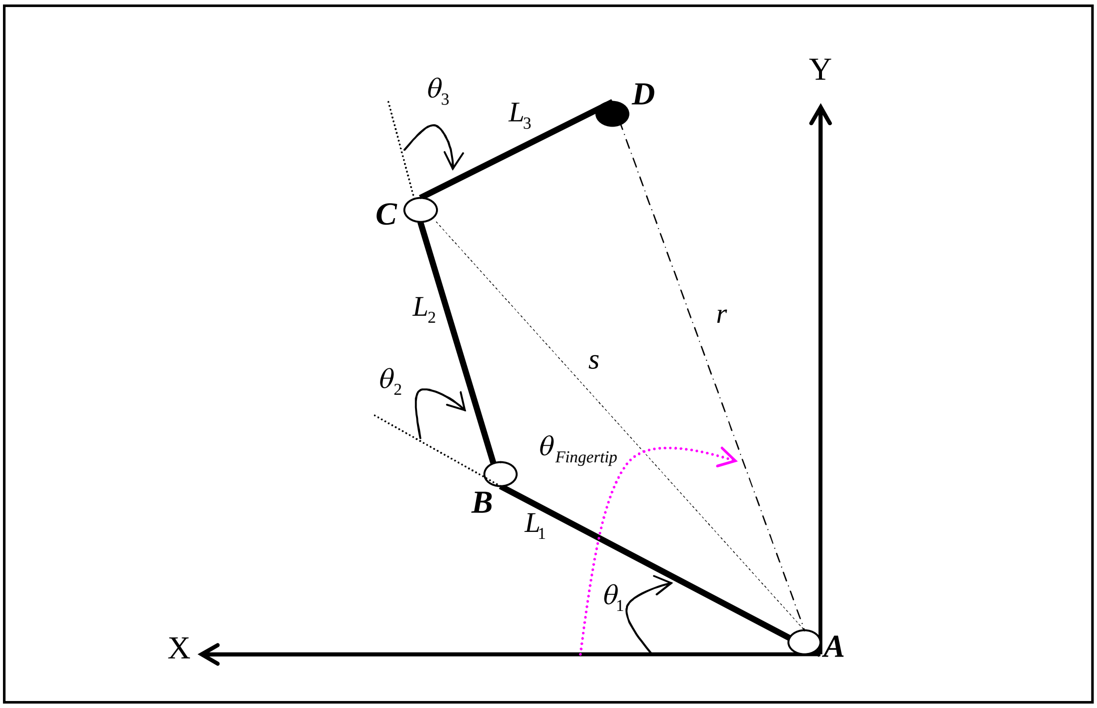


Figure 3.8: Schematic solution for θ_2.

By analysis of the quadrilateral, shown in Figure 3.8 above, the following equations are obtained:

$$s^2 = L_1^{\ 2} + L_2^{\ 2} - 2L_1L_2\cos(\pi - \theta_2) \quad \text{(Equation 3.11)}$$

$$\cos\angle ACD = \cos(\pi - \theta_3 - \angle ACB) \quad \text{(Equation 3.12)}$$

$$\frac{\cos\angle ACB}{1} = \frac{s^2 + L_2^{\ 2} - L_1^{\ 2}}{2sL_2} \quad \text{(Equation 3.13)}$$

$$\frac{\sin\angle ACB}{L_1} = \frac{\sin(\pi - \theta_2)}{s} \quad \text{(Equation 3.14)}$$

$$r^2 = s^2 + L_3^{\ 2} - 2sL_3\cos\angle ACD \quad \text{(Equation 3.15)}$$

where:

L_1, L_2, and L_3 = Lengths of PP, MP and DP, respectively.

r^2 $= P_x^{\ 2} + P_y^{\ 2}$

PIP and DIP Joint Angle

By using the trigonometric identities laws of sine and cosine to derive Equation 3.11 to Equation 3.15, and together with Equation 3.10, the following can be found:

$$r^2 - L_1^{\ 2} - L_2^{\ 2} - L_3^{\ 2} = 2L_1L_3\cos 19\alpha + 2L_1L_2\cos 11\alpha + 2L_2L_3\cos 8\alpha \quad \text{(Equation 3.16)}$$

where:

$$\alpha = \frac{1}{11}\theta_2 \quad \text{(Equation 3.17)}$$

By using binomial analysis technique, shown in Equation 3.18 below, to expand and derive Equation 3.16 to Equation 3.19.

$$\cos n\alpha \quad = \sum_{k}^{n} \binom{n}{k} \left(\cos^{k}\alpha\right)\left(\sin^{n-k}\alpha\right)\left(\cos(\tfrac{1}{2}(n-k)\pi)\right) \quad \text{(Equation 3.18)}$$

where:

k = 0, 1, 2, …., n

$\binom{n}{k} = {}^{n}C_{k}$

$$ax^{19} + bx^{17} + cx^{15} + dx^{13} + ex^{11} + fx^{9} + gx^{8} + hx^{7} + qx^{6} + jx^{5} + kx^{4} + lx^{3} + mx^{2} + nx + o = 0$$

(Equation 3.19)

where:

	x	$= \cos\alpha$ (Equation 3.20)
x^{19} :	a	$= 2^{19} L_1 L_3$
x^{17} :	b	$= -19 \times 2^{17} L_1 L_3$
x^{15} :	c	$= 19 \times 2^{18} L_1 L_3$
x^{13} :	d	$= -665 \times 2^{13} L_1 L_3$
x^{11} :	e	$= 1729 \times 2^{11} L_1 L_3 + 2^{11} L_1 L_2$
x^{9} :	f	$= -2717 \times 2^{9} L_1 L_3 - 11 \times 2^{9} L_1 L_2$
x^{8} :	g	$= 2^{8} L_2 L_3$
x^{7} :	h	$= 627 \times 2^{9} L_1 L_3 + 11 \times 2^{9} L_1 L_2$
x^{6} :	q	$= -2^{9} L_2 L_3$
x^{5} :	j	$= -627 \times 2^{6} L_1 L_3 - 77 \times 2^{5} L_1 L_2$
x^{4} :	k	$= 5 \times 2^{6} L_2 L_3$
x^{3} :	l	$= 285 \times 2^{3} L_1 L_3 + 5 \times 2^{3} L_1 L_2$
x^{2} :	m	$= -2^{6} L_2 L_3$
x :	n	$= -19 \times 2^{1} L_1 L_3 - 11 \times 2^{1} L_1 L_2$
	o	$= L_1{}^{2} + L_2{}^{2} + L_3{}^{2} + 2 L_2 L_3 - r^{2}$

From Equation 3-19, nineteen solutions for variable x can be obtained, containing real and imaginary values. Only one real value can be selected by recalculating

Equations 3.13 and 3.14, and requiring that all angles are positive. Besides, the value is also needed to recalculate Equation 3.17 and the value of θ_2 must be less than or equal to 110^0.

After the x value of Equation 3.19 was chosen, then the value of α was obtained from Equation 3.20. The PIP joint angle θ_2 is obtained by applying the value of α into Equation 3.17. After the value θ_2 was known, then the value of the DIP joint angle, θ_3, could be derived from Equation 3.10.

<u>MCP Joint Angle</u>

According to the quadrilateral of Figure 3.8, the angle of the MCP joint θ_1 can be obtained as:

$$\theta_1 \quad = \theta_{\text{Fingertip}} - \angle\text{DAC} - \angle\text{CAB} \quad \text{(Equation 3.21)}$$

<u>where</u>:

$$\theta_{Fingertip} \quad = \text{a}\tan 2(\text{Py}, \text{Px})$$

By using the trigonometric identities law of cosine, the value of $\angle\text{DAC}$ and $\angle\text{CAB}$ can be obtained by analysing the two equations below.

$$\text{L}_3{}^2 \quad = \text{r}^2 + \text{s}^2 - 2\text{rs}\cos\angle\text{DAC} \quad \text{(Equation 3.22)}$$

$$\text{L}_2{}^2 \quad = \text{L}_1{}^2 + \text{s}^2 - 2\text{L}_1\text{s}\cos\angle\text{CAB} \quad \text{(Equation 3.23)}$$

By reconstructing the Equations 3.21 to 3.23, the following is obtained.

$$\theta_1 \quad = \theta_{\text{Fingertip}} - (\text{a}\cos\angle\text{CAB} + \text{a}\cos\angle\text{DAC}) \quad \text{(Equation 3.24)}$$

The value of s can be obtained from Equation 3.11 since the angle value of DIP joint θ_2 was obtained from Equation 3.19, and the value of distance r is given by $\sqrt{P_x{}^2 + P_y{}^2}$. If the value of $\theta_{Fingertip}$ is negative, it means that the fingertip is located on the right hand side of Figure 3.8, because $\theta_{\text{Fingertip}} = \text{a}\tan 2(\text{Py}, \text{Px})$.

3.4.3 Verification of Closed Form Formula of Inverse Kinematics

This section is to verify how well the deduced closed form formulas of θ_1, θ_2 and θ_3 of inverse kinematics compare to the predicted value in forward kinematics. Figure 3.10 shows the error between the closed form value in inverse kinematics and predicted value in forward kinematics. This error was measured by a chain of transformations using a variety of upper limb configurations (see Figure 3.9).

Forward Kinematics
Inverse Kinematics
Forward Kinematics

Figure 3.9: Configuration to validate the equation deduced in inverse kinematics.

The verification is needed to know how well the formula in inverse kinematics, which was deduced from the quadrilateral (see Figure 3.8). The verification is used to verify the formula deduced from inverse kinematics with the predicted formula in forward kinematics.

Figure 3.10 is to validate the error values of the close form formulas of inverse kinematics and forward kinematics by using Equations 3.25 to 3.27. The angles θ_{1D} , θ_{2D} and θ_{3D} are the variables holding the values of θ_1 ,θ_2 and θ_3, deduced using the inverse kinematics Equations 3.24, 3.19, and 3.10, respectively. The θ_{1P} , θ_{2P}, θ_{3P} are the variables' values predicted by the forward kinematics.

$$\theta_{1\mathrm{Error}} = \theta_{1\mathrm{D}} - \theta_{1\mathrm{P}} \quad \text{(Equation 3.25)}$$

$$\theta_{2Error} = \theta_{2D} - \theta_{2P} \quad \text{(Equation 3.26)}$$

$$\theta_{3Error} = \theta_{3D} - \theta_{3P} \quad \text{(Equation 3.27)}$$

The maximum errors in θ_{1Error}, θ_{2Error} and θ_{3Error}, shown in Figure 3.10, are in the 1x10^{-4} rad (0.00573^0) range, which is a very low error. This means that the formulas deduced in inverse kinematics are correct.

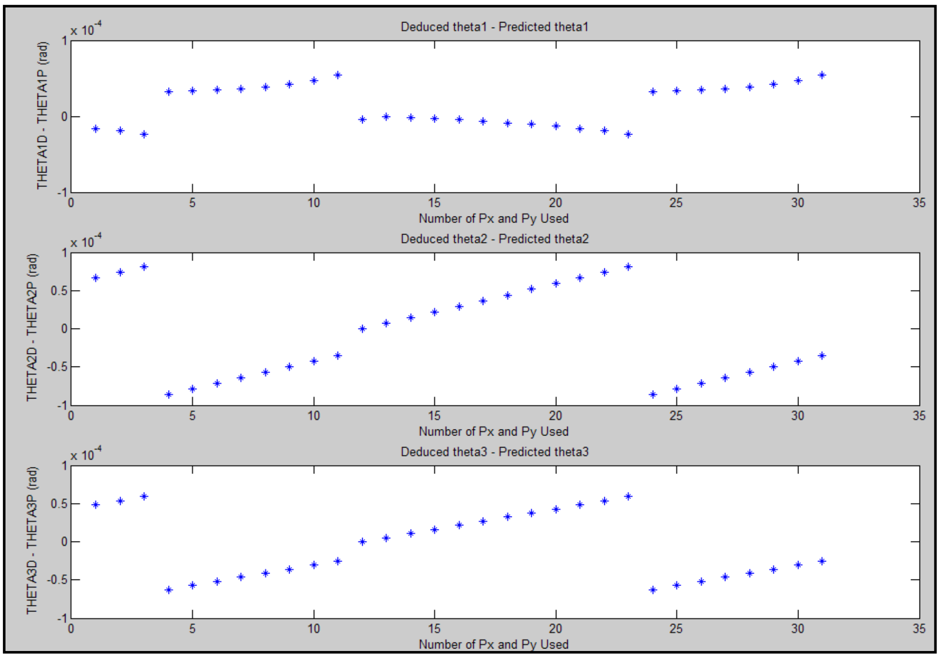


Figure 3.10: Error between the closed form θ_{1D} , θ_{2D}, θ_{3D} and predicted θ_{1P} , θ_{2P}, θ_{3P} .

3.5 Summary

A bio-mimetic prosthetic middle finger designed to mimic the kinematics architecture of the natural middle finger has been presented in this chapter. This finger design is according to the human hand and finger anatomy in order to allow the hybrid actuation mechanism to be found to increase the number of active DOF, and it is custom made due to large variation of every single human hand/finger.

The prosthetic finger has 3 DOF in contrast to 4 DOF of the human finger, since the movement of abduction-adduction in the MCP joint is not necessary for the middle finger to accomplish grasping functionality.

The material used to manufacture by rapid prototyping also presented in this chapter. Material selection is very important in order to design a safe grasping functionality of the hand/finger.

Kinematic analysis of the prosthetic finger was carried out to determine the relationship between the angular positions of each joint with the fingertip position. The analysis showed that the requirements of the DC actuator number of cycle turns and the requirement of SMA contraction will be achievable. The finger design is modelled as a robot hand fixed to the finger palm. Denavit-Hartenberg (D-H) theorem and homogeneous transformation matrix are used to describe the fingertip position with respect to the base reference or orientation frame. The fingertip working envelope is plotted after the fingertip position formula is achieved. This fingertip working envelope shows all the X-Y data points generated by cycling through different combinations of joint angles and predict the fingertip position in x and y co-ordinates for each for each of them and it also illustrates finger tip position range.

A quadrilateral method associated with trigonometric identities and binomial are used to evaluate the angle of each joint given a desired position and orientation of the end-effectors of the manipulator and the geometric link parameters with respect to a reference coordinate system (inverse kinematics) is also presented.

The simulation carried out shows that the errors between the deduced formula in inverse kinematics and the predicted value in forward kinematics are very small for the application it is being used in.

CHAPTER 4
Hybrid Actuation Mechanism

4.1 Introduction

The purpose of a prosthetic hand/finger is to imitate the grasping capabilities of the human hand/finger, with obvious improvements in the quality of life of amputees. Hence, to be effective it should allow holding and grasping of objects with sufficient force/torque, as these two are important functions of a human hand.

The force/torque of each joint and each phalanx needed to maintain grasp is presented in this chapter. Section 4.2 provides an overall specification for the main parameters of the hybrid actuation system under consideration as well as an overview of the hybrid actuation design. Section 4.3 briefly considers the background information on some actuators used in prosthetic hands. The design of the DC actuator, such as the torque/force analysis, based on the relevant gears is discussed in Section 4.4. Section 4.5 presents the SMA actuator design together with its torque/force analysis.

4.2 Specification

In the past, electric motors were the primary actuators used in prosthetic devices as they have a high torque and are easy to control. However, their heavy weight, size and shape have been restrictive and uncomfortable to the users [3]. Due to these limitations, some researchers have used Shape Memory Alloys (SMAs), or Electroactive Polymers (EAPs) to replace the electric motor and the pneumatic actuator, as they are operationally similar to human muscles [8]. Both are lightweight and direct-driven actuators able to provide ways to increase the functionality of artificial hands without adding mechanical complexity [8]. EAP materials have higher response speeds and lower densities compared to SMAs [9]. However, they have low actuation forces, mechanical energy density and lack of robustness compared to the SMAs [9]. Furthermore, the big advantages of SMAs are their incredible small size and volume, their high force to weight ratio and their low cost [3, 19]. One drawback with SMAs is that in prosthetic devices large lengths of SMAs are needed, something that limits their life cycle [3].

Hence, this chapter presents the actuation mechanism used which is a hybrid system consisting of a DC actuator and an SMA actuator.

The hybrid actuation mechanism must be designed so that it is as light and compact as possible, placing all actuators into palm in order to leave the prosthetic socket empty for other components such as battery and electronics cable. The list of specifications for the actuation mechanism is presented in Table 4.1.

Table 4.1: Specifications of hybrid actuation mechanism design.

Weight	It must fit into150 ~ 200 gram
Force	The output actuation force must fit of: 40N or above on the end of PP [59], and 32N or above on the fingertip (middle of DP) [59]
Volume / Size	Two actuators must fit into a volume of palm, that is 47mm length x 22mm width x 30mm height, in order to empty the volume of the socket for some other purpose, such as power cable, etc..
Cost	The cost of the actuation mechanism design as low as possible. Manage in £500 for one finger.
Time response	Time to fully open the finger/hand: 1.5 second [62]. Time to fully close the finger/hand: 2 seconds [62].

4.3 DC Actuator

The movement of the flexion-extension in the MCP joint is controlled by a DC actuator because the proximal phalanx(PP) of the middle finger in one finger tasks can create a higher force than the distal phalanx (DP), which is about 40N in the PP of an adult male‘s middle finger [58].

Maxon DC motor RE10 1.5Watt was used in the design. This motor is extremely light weight and small in size, just around 20g, 10mm diameter and 57mm total length. All these parameters are including metal gearhead and encoder, as shown in Figure 4.1. The absolute reduction of the metal gearhead is 64 and the counts per turn of the encoder is 256, as shown in the Table 4.2. It is placed in the palm to control the flexion-extension of MCP joint, as this joint needs more force than the PIP and DIP joints [58].

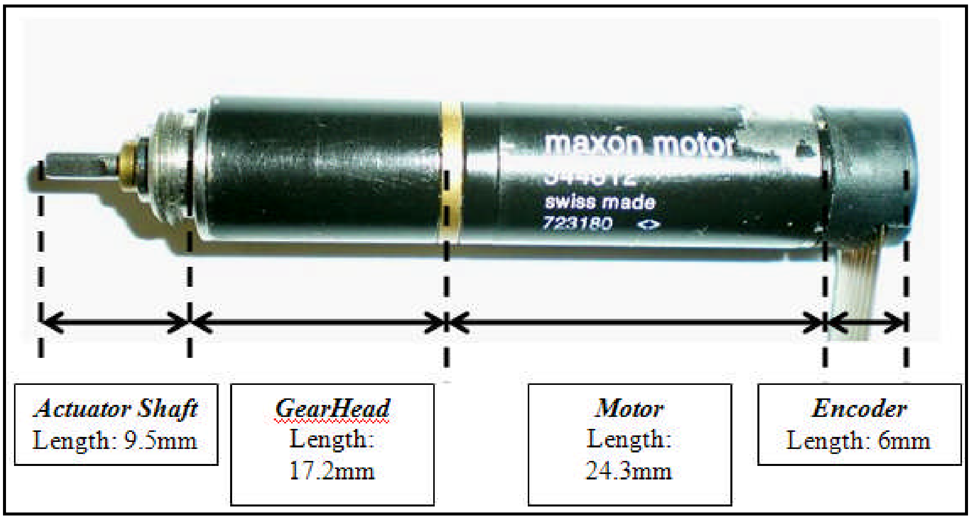


Figure 4.1: DC actuator and dimensions.

Table 4.2: Specification of DC actuator.

Nominal Voltage	5V
No Load Speed	12800rpm (1340rad/s)
Nominal Torque	1.48mNm
Reduction of Gearhead	64:1
Encoder Count per Turn	256 pulse

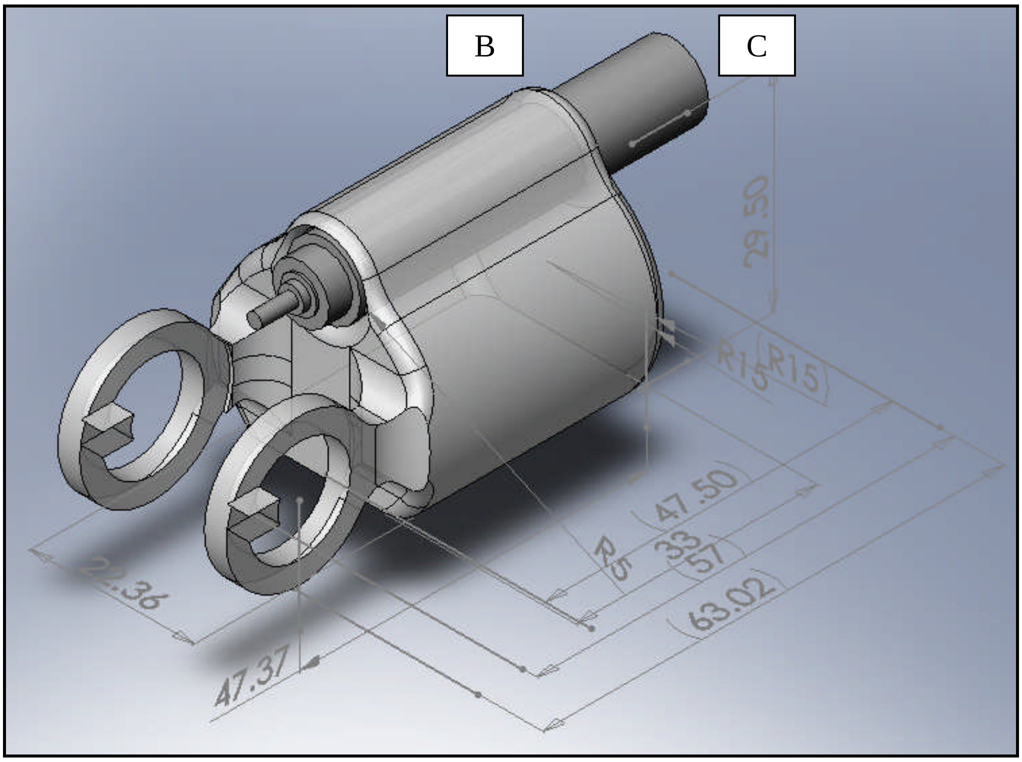


Figure 4.2: DC actuator and finger housing.

The result of applying the DC actuator of Figure 4.1 into the finger design of Chapter 3 (Figure 3.3), is shown in Figure 4.2. The DC actuator is 14.5mm (B to C) outside/away from the finger palm housing due to the finger's need to be 'hang up' with a rob, as shown in Figure 4.3(a).

Consequently, the total length of the finger palm design from 33mm, as shown in Figure 3.3, to 47.5mm, as shown in A to C of the depicted finger (Figure 4.3(b)). This length is the same as that of the author's palm. Therefore, the total length of the finger design (including palm) is 181.57mm.

4.2.1 Force/Torque of the MCP Joint and Proximal Phalanx

The driving gear selection is of crucial importance as it not only converts the rotation of the DC actuator to flexion-extension of the finger but it also needs to have high enough transmission torque to the MCP joint.

At first, 21:12 bevel gear was used. However, the torque of the MCP joint is just 0.166Nm and the force transmission to the end of the PP, F_{pp}, (see Figure 4.2) is just

2.83N, as shown in Equations 4.1 and 4.2. It is extremely low compared to the force one adult male finger can generate [58].

$$\tau_{\mathrm{MCPjoint}} = \tau_{\mathrm{DCactuator}} \times \mathrm{Gear\_ratio} \quad \text{(Equation 4.1)}$$

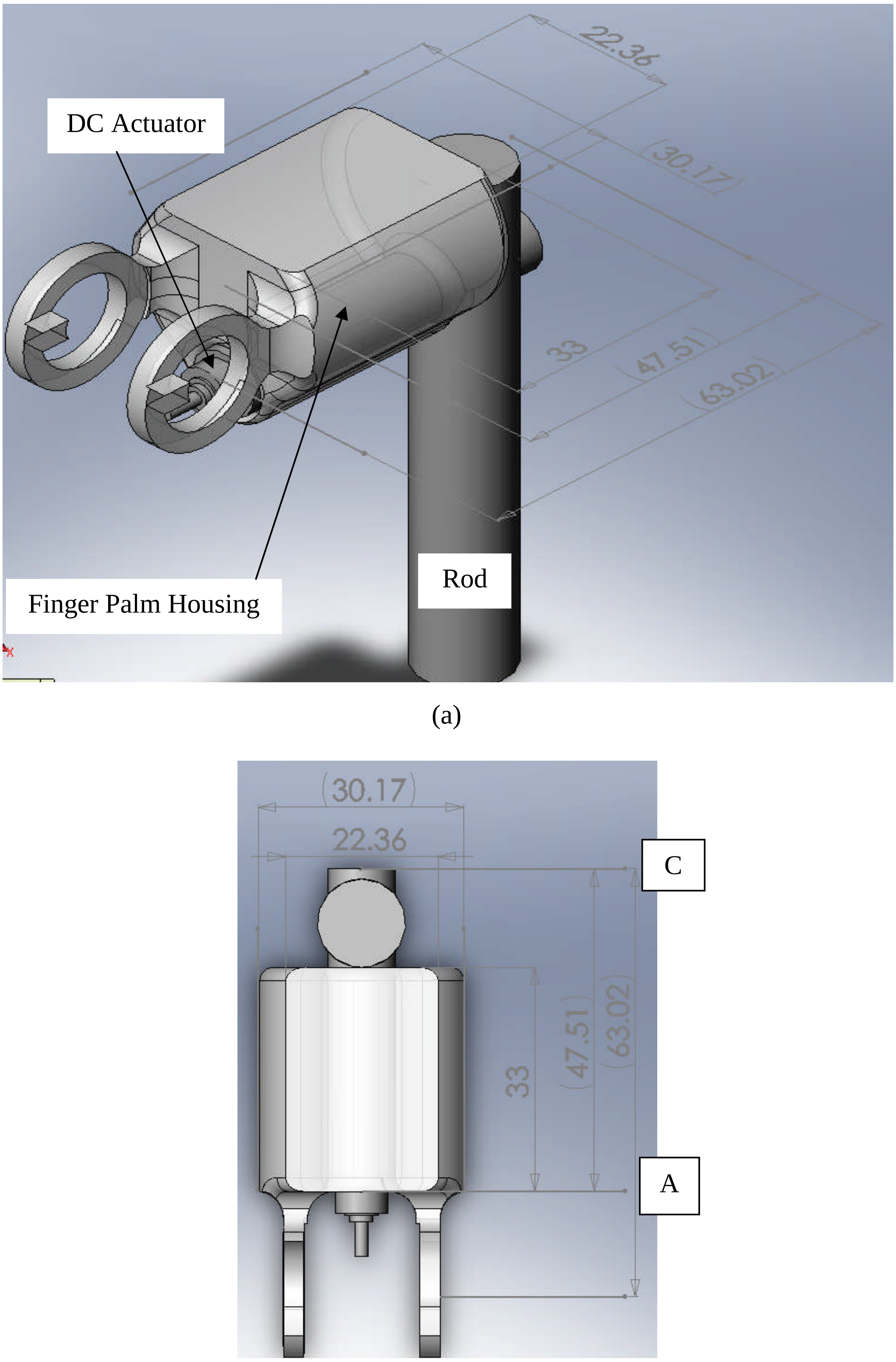
where:

Gear_ratio	= External Gear ratio
$\tau_{DCactuator}$	= Torque of DC actuator
$\tau_{MCPjoint}$	= Torque of MCP joint

$$F_{\mathrm{PP}} = \frac{\tau_{\mathrm{MCPJoint}}}{L_{\mathrm{PP}}} \quad \text{(Equation 4.2)}$$

where:

F_{pp}	= Force at end of proximal phalanx (PP)
τ_{MCPJoint}	= Torque of MCP joint
L_{pp}	= Length of PP



(a)

(b)

Figure 4.3: DC actuator, finger palm housing and rod, (a) Isometric view, (b) Top view.

Hence, a higher gear ratio of worm gear is used instead of a bevel gear. This is because the gear ratio is directly proportional to the torque of the MCP joint, as illustrate in Equation 4.1. The maximum transmission torque in the MCP joint is 4.26Nm by using a 45:1 worm gear, and the force at the end of the Proximal Phalanx, F_{pp} (see Figure 4.4) is 72.71N.

The design shown in Figure 3.2 was computed based on a worm gear. Room for the placement of the DC actuator is found on the metacarpal bone (palm).

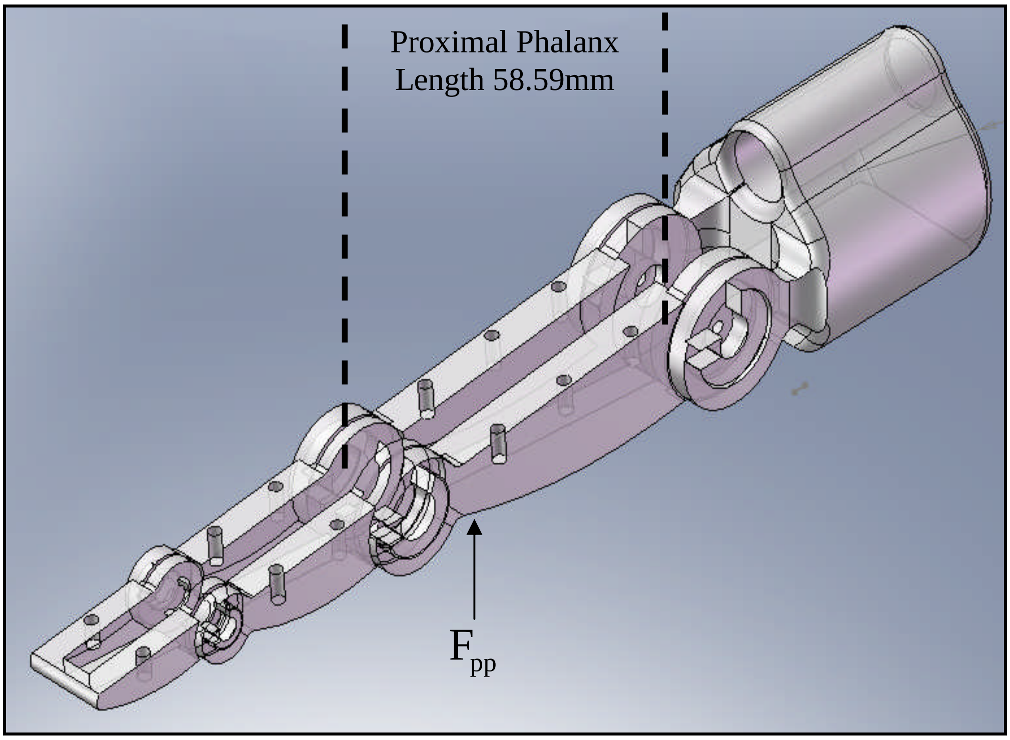


Figure 4.4: Transmission force at PP.

4.4 SMA Actuator

The SMA actuator design is the most challenging part in order to produce the smallest size possible to achieve a large linear stroke or displacement for large angular motion and to produce enough grasping force. We propose a mechanism associated with SMAs wire in order to create large linear displacement through a small change of SMAs wires. The SMA actuator associated with a spring is used to control the flexion-extension of the PIP joint.

4.3.1 Linear Displacement of SMA Actuator

The displacement/stroke of the SMA actuator is very important and challenging, as it serves to pull the PIP joint sufficiently from 0^0 to 110^0.

Before designing the SMA actuator, the limited dimension to place the SMA actuator is the crucial issue to be considered. The limited dimensions for the SMA actuator are 47.50mm (A to C of Figure 4.3(b)) length × 37mm width, as shown in Figure 4.3(b).

The second crucial issue to be considered is the way to connect the tendon cable within the finger and the SMA actuator, as shown in Figure 4.8. The tendon cable of the finger as shown in Figure 4.5 is 10mm from the centre of the PIP joint, and the other end of the cable will connect to the SMA actuator. The SMA actuator is positioned on the palm (as shown in Figure 4.5). The tendon cable of the SMA actuator will be in between 85mm (A to D of Figure 4.5) to 130mm (A to E of Figure 4.5).

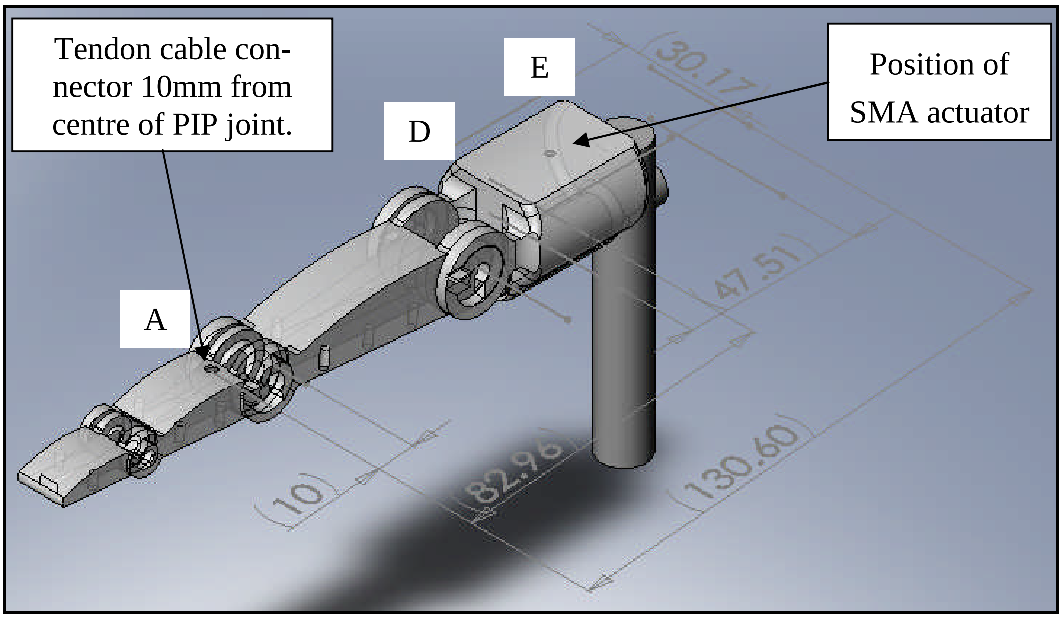


Figure 4.5: Dimension to position SMA actuator.

The purpose of Figure 4.6 is to evaluate the linear displacement ΔL needed for the SMA actuator in order to rotate the PIP joint from 0^0 to 110^0, where 'O' is the PIP joint, A is the tendon cable connector on the finger when the PIP joint is at 0^0. C_1 and C_2 are the tendon cable connectors on the SMA actuator before and after SMA actuator activation. B is the final position of A after PIP joint turns 110^0 circular motion and L_1 is the length of the tendon cable. The circular distance $\hat{a}$ is 19mm.

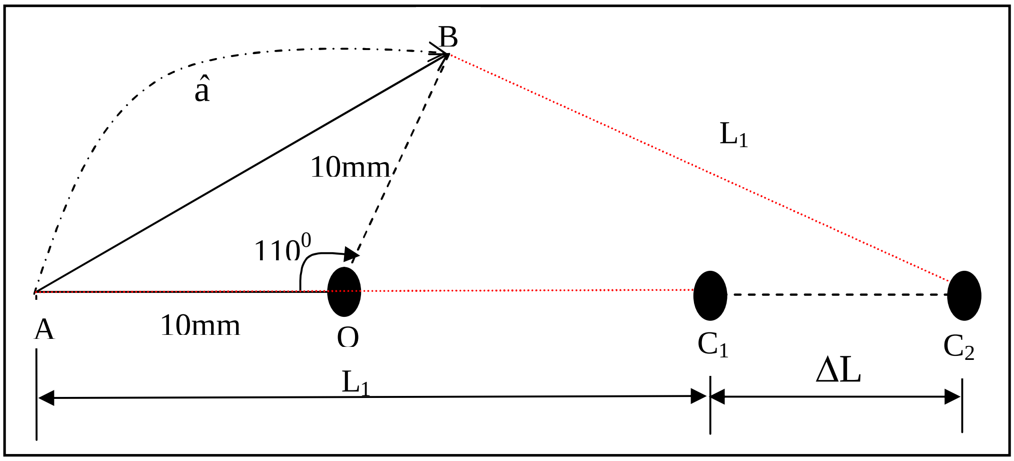


Figure 4.6: Schematic solution for linear displacement ΔL.

Using law of cosine, the values of AB and $\angle OAB$ are 16.38mm and 35^0, respectively. After that, using law of cosine again, yield:

$$L_1^2 = (L_1 + \Delta L)^2 + (AB)^2 - 2(L_1 + \Delta L)(AB)\cos\angle BAO \quad \text{(Equation 4.3)}$$

There are two answers for Equation 4.3; however, only positive value will be chosen.

According to the properties of Flexinol's nickel-titanium (Ni-Ti) SMA wire contract length is typically just 5% of its length, therefore it is impossible to directly connect the SMA wire to the limited dimensions of Figure 4.3. For example, if the length of L_1 is 100mm, then the linear displacement ΔL for the PIP joint to rotate from 0^0 to 110^0 is 12.98mm, which is evaluated by using Equation 4.3.This means 259.6mm of SMA wire is needed. It is extremely long and impossible to connect to the position of the SMA actuator, as shown in Figure 4.3(b).

Consequently, the principle of leverage, as shown in Figure 4.7, is applied in order to deduce the total length of the SMA wire. The purpose of using Figure 4.7 is to achieve large linear displacement from a small linear displacement change.

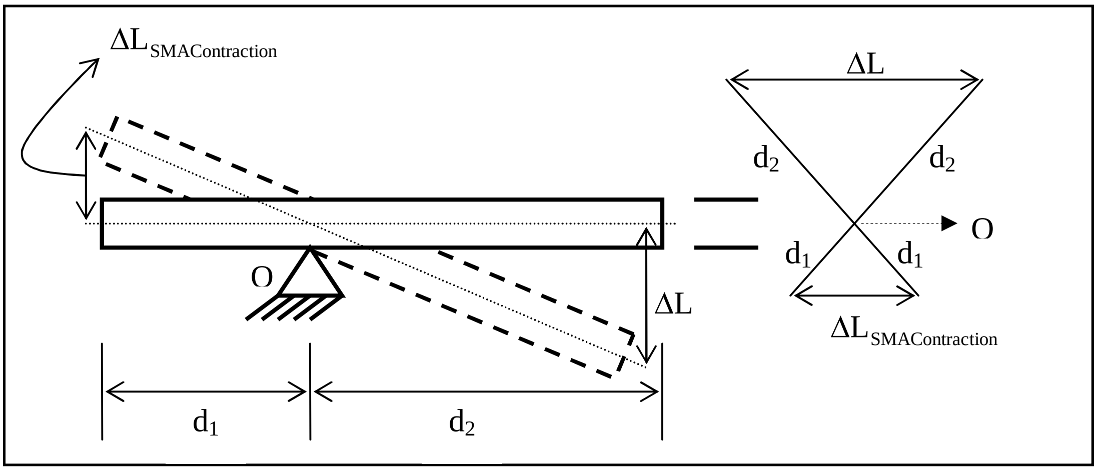


Figure 4.7: Principle of leverage.

Using law of cosine, the relationship between the $\Delta L_{SMAContraction}$ and ΔL is obtained as.

$$\Delta L_{SMAContraction} = \frac{d_1}{d_2} \times \Delta L \text{ (Equation 4.4)}$$

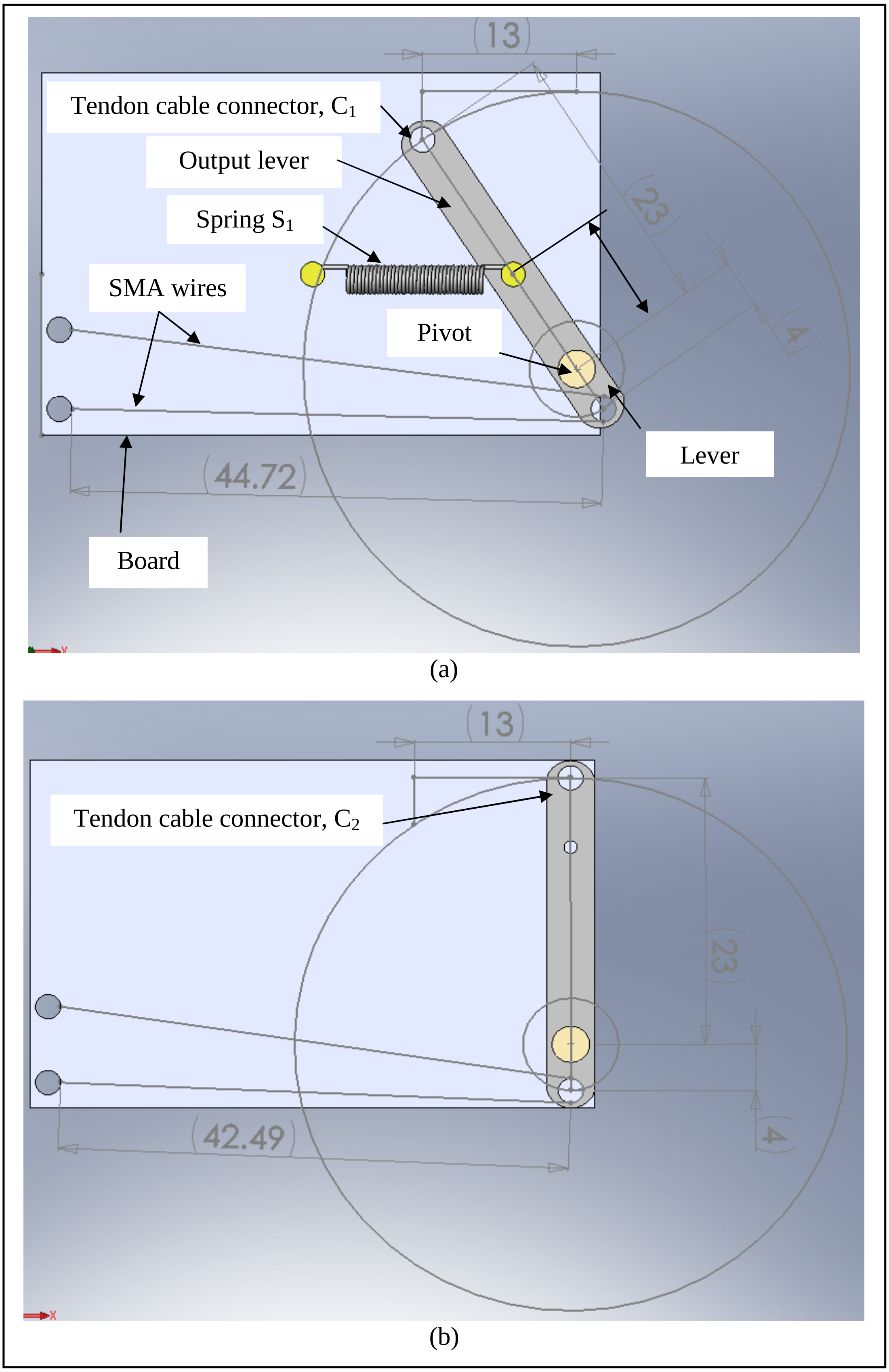
Equation 4.4 can be extended by applying the total length of the SMA;

$$L_{SMA} x R_{contraction} = \frac{d_1}{d_2} \times \Delta L \text{ (Equation 4.5)}$$

where: L_{SMA} is the total length of the SMA wire and $R_{contraction}$ is the SMA deformation/contraction rate.

45mm length of L_{SMA} (length of the SMA actuator location), 5% SMA deformation/contraction rate and ΔL =13mm are used to design an SMA actuator. According to Equation 4.5, the ratio of the d_1 / d_2 is 0.174, and the length of d1 and d2 can be evaluated and achieved in 4.00mm and 23mm, respectively. The design of the SMA actuator is shown in Figure 4.8.

The design of the SMA actuator functions as follows: When the SMA actuator is unactivated or the SMA wire is unheated or cooled after heating, the actuator looks like Figure 4.8 (a). If the SMA wires heat up, they contract and pull the output lever in the opposite direction to that of the SMA wire contraction, as shown in Figure 4.8 (b), due to the pivot used. This pulls the tendon wire, connected on one side to the ‘tendon cable connector’ and on the other to the MP of the finger, as shown in Figure 4.9, to allow the PIP joint to make an angular motion. The purpose of the spring S_1 in Figures 4.8 (a) and 4.9 is to bring the output lever back when the SMA wire is cooled or released after being heated. It also can reduce the recovery time so that the finger can be released faster.



(a)

(b)

Figure 4.8: SMA actuator wire (a) before contraction, (b) after contraction.

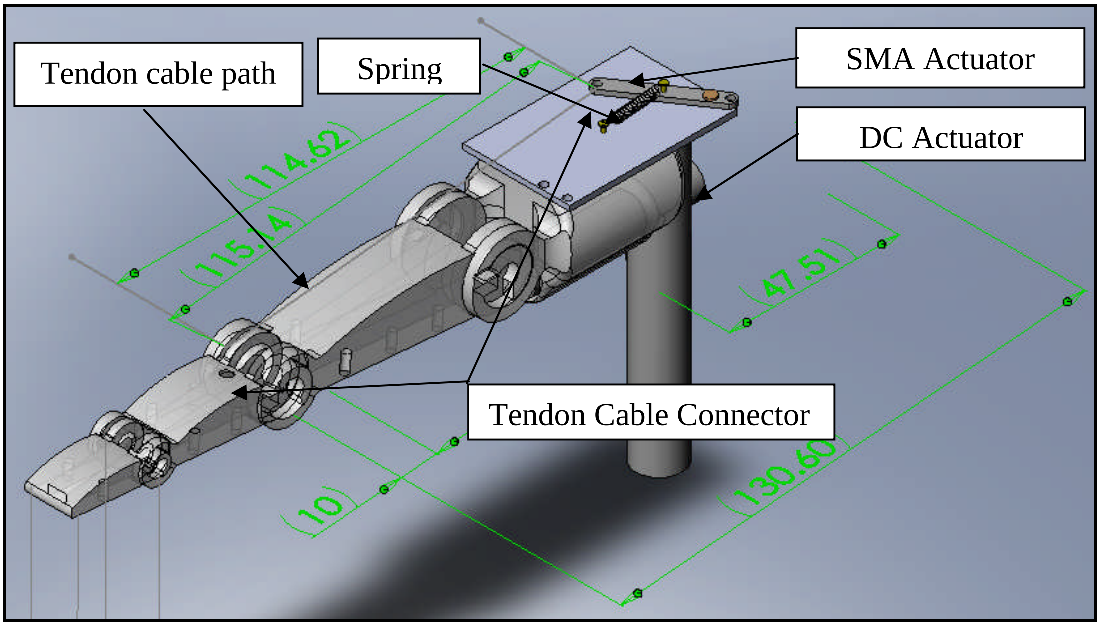


Figure 4.9: DC and SMA actuators on finger.

In order to know the relationship between the linear displacement ΔL and the PIP joint angle, Equation 4.3 can be extended to:

$$L_1^2 = (L_1 + \Delta L)^2 + (AO^2 + AO^2)(1 - \cos\theta_2) - (2 \times \sqrt{AO \times AO})(L_1 + \Delta L)\sqrt{(1 - \cos\theta_2)}\cos\left(\frac{\pi - \theta_2}{2}\right)$$

(Equation 4.6)

where L_1 is the length of the tendon cable and AO is the distance between the tendon cable connector on the finger and PIP joint.

Equation 4.6 shows the relationship between the linear displacement of SMA actuator ΔL and angular motion of PIP joint θ_2. Six solutions for variable θ_2 can be obtained; however, only one in the range $0 \le \theta_2 < 180$ can be selected.

Applying 100mm to the value of L_1 and 10mm to the value of AO, the relationship between the linear displacement of SMA actuator ΔL and angular motion of PIP joint θ_2 is shown in Figure 4.10. Since Equation 4.6 contains a cosine function, therefore Figure 4.10 shows the relationship between ΔL and θ_2 as non-linear relationship.

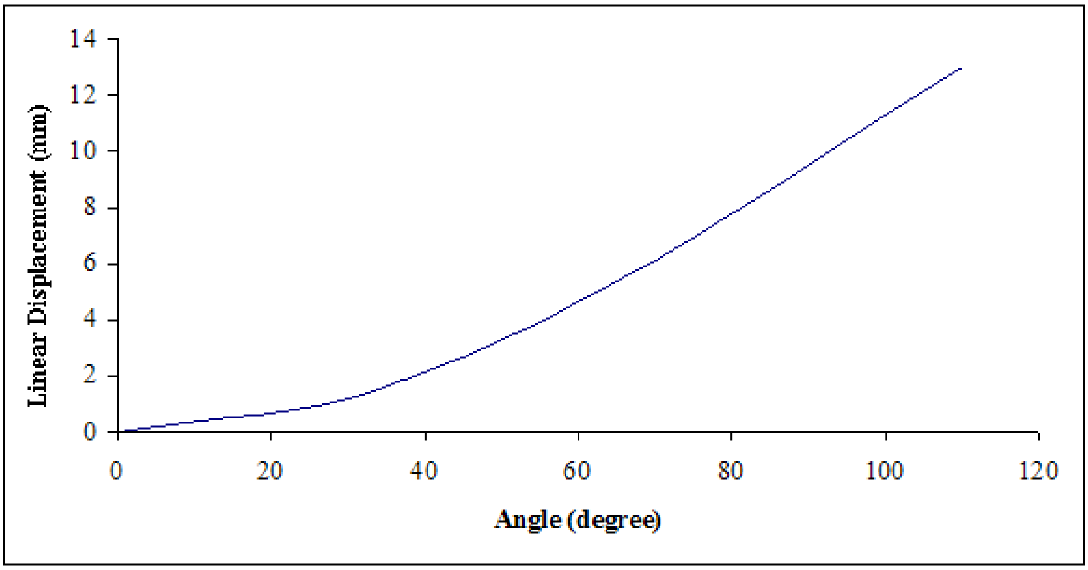


Figure 4.10: Relationship between linear displacement of SMA actuator and angle movement of PIP joint θ_2.

4.3.2 Force/Torque of SMA Actuator

The force/torque is the second challenging part for designing the SMA actuator in order for it to be sufficient for grasping. The previous section has just evaluated the linear displacement of the SMA actuator based on a single SMA wire according to the principle of leverage. However, this principle will reduce the SMA wire force ΣF_{SMA} for the lever output force ΣF_o of the SMA actuator.

$$\Sigma M = 0 \quad \text{(Equation 4.7)}$$

$$\Sigma F_{SMA} d_1 - \Sigma F_{S1} d_3 - \Sigma F_o d_2 = 0 \quad \text{(Equation 4.8)}$$

where: ΣM : Summation moment of output lever in SMA actuator,

ΣF_{SMA} : Summation force of SMA wire,

ΣF_{S1} : Summation force of spring S1 in SMA actuator,

ΣF_o : Summation output force of output lever / force of tendon cable (Figure 4.11),

d_1 : Distance between SMA and pivot (Figure 4.11),

d_2 : Distance between tendon cable connector of SMA actuator and pivot (Figure 4.11),

d_3 : Distance between spring of SMA actuator and pivot (Figure 4.11).

Assuming no load is connected to the tendon cable connector of the SMA actuator, maximum force and 5% deformation will be produced by the SMA wire and without friction between the output lever and board, the maximum ΣF_o in Figure 4.11 is obtained as:

$$(F_{SMA}(\cos 2.17 + \cos 8.18)) \times d_1 - (F_{S1} \cos 4.24) \times d_3 - \Sigma F_o d_2 = 0 \quad \text{(Equation 4.9)}$$

$$\Sigma F_o = \frac{(F_{SMA}(\cos 2.17 + \cos 8.18)) \times d_1 - (F_{S1} \cos 4.24) \times d_3}{d_2}$$

(Equation 4.10)

From the Flexinol wire properties given by Milford Instruments, the 0.010” ($250\mu m$) maximum force is 9.12N. The S1 spring constant is 0.3N/mm and the linear displacement is 5.44mm from the original position (Figure 4.8(a)) to where the SMA wire of the SMA actuator was fully contracted (Figure 4.8(b)). Therefore, applying all of the values to Equation 4.10, the value of ΣF_o is 2.48N.

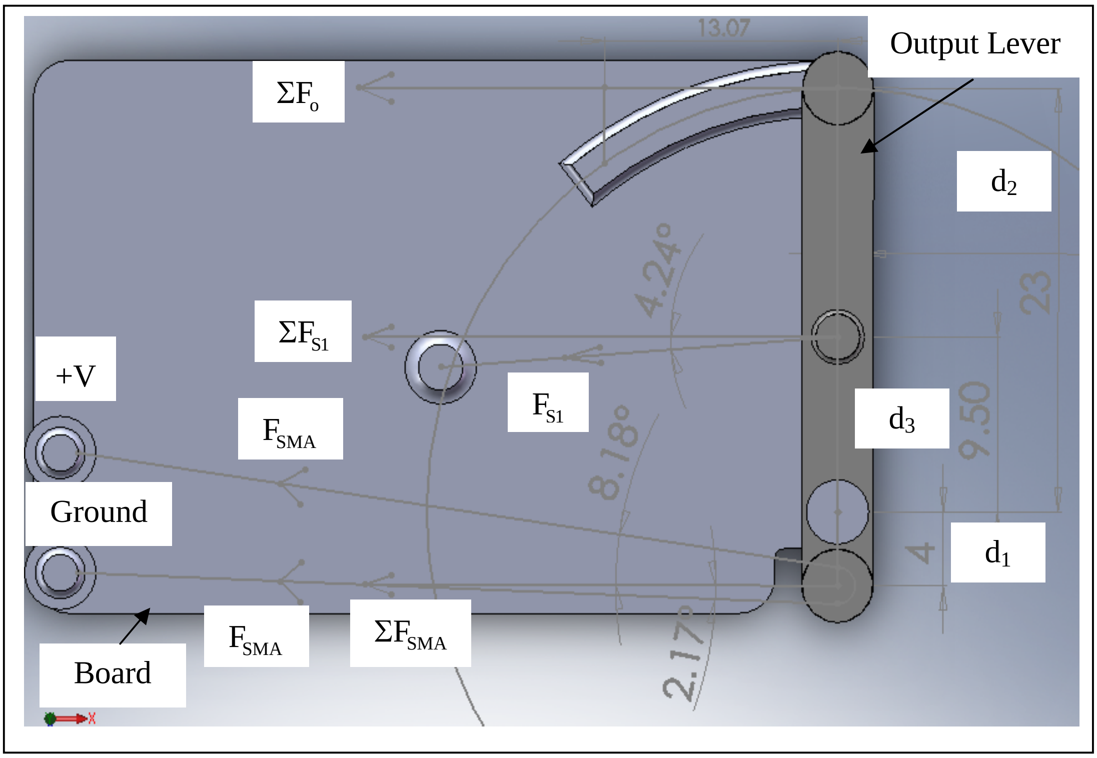


Figure 4.11: Static equilibrium of SMA actuator.

Applying this tendon force ΣF_o to the finger, as shown in Figure 4.12, and to Equation 4.11, the torque of the PIP joint θ_2 is 7.50e-3Nm (7.5Nmm) (The centre of mass of the MP and DP is 18.44mm from the PIP joint and the total mass of the MP, DP and some mechanism (such as pulley, round belt, etc.) is 0.017kg (17gram), the spring constant of the spring S2 is 0.1N/mm and the displacement from 0^0 to 110^0 is 29mm).

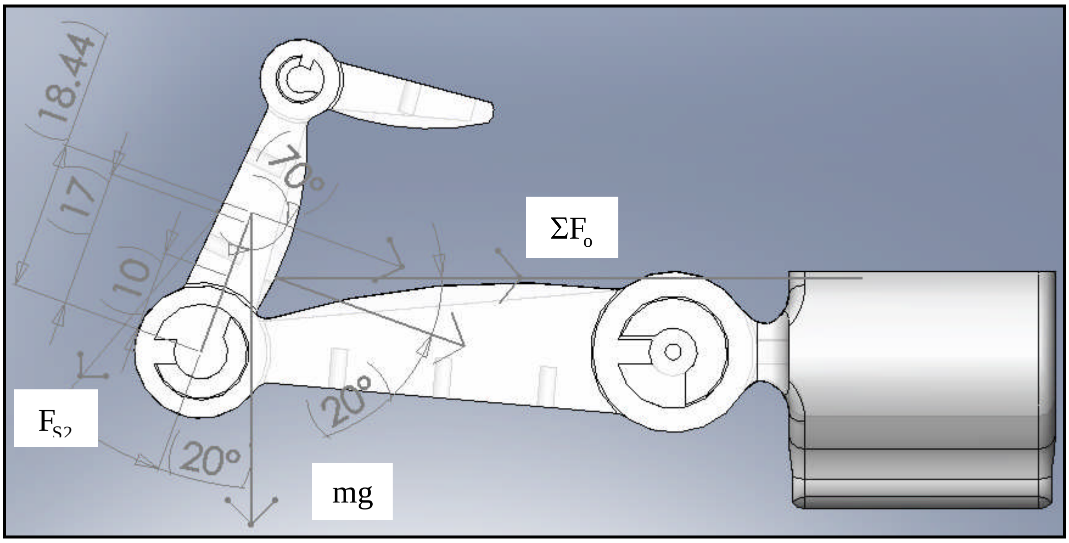


Figure 4.12: Static equilibrium of finger.

$$(\Sigma F_o \cos 20) \times L_c + (mg \cos 70) \times L_{cm} - (F_{S2} \sin 20) \times L_{S2} = M_{\theta_2}$$

(Equation 4.11)

where: M_{θ_2} : Moment or torque of PIP joint θ_2 (Nmm)

F_{S2} : Spring S2 force (N)

L_c : Distance between ‘tendon cable connector’ on finger and PIP joint θ_2 (mm)

L_{cm} : Distance between centre of mass of MP and DP and PIP joint θ_2 (mm)

L_{S2} : Distance between spring S2 connected to PIP joint θ_2 (mm)

Since the DIP joint is coupled with PIP joint by the 8/11 ratio pulley, the torque of the DIP joint θ_3 is hence 0.01Nm and the force of the middle of the DP is 0.836N. It is extremely low compared to [58], because it can give 32N for an adult male 1-finger task in the DP of the middle finger. Assuming this 32N is applying in the middle of DP which design in Chapter 3, a 0.38Nm flexor torque is needed in DIP joint θ_3.

Consequently, in order to increase the flexor torque of the DIP joint θ_3 to 0.38Nm, the torque of the PIP joint θ_2 must be increased from 7.50e-3Nm (7.5Nmm) to 0.278Nm (278.46Nmm). In order to increase the torque of the PIP joint, it can increase the number of SMA wires connected in parallel [63]; however the electric flow must be connect in series in order to have the same linear stroke, which evaluate from the section 4.3.1.

The Equation 4.10 can be Reformed to Equation 4.12:

$$\Sigma F_o = \frac{(nF_{SMA}(\cos 2.17 + \cos 8.18)) \times d_1 - (F_{S1} \cos 4.24) \times d_3}{d_2}$$

(Equation 4.12)

<u>where</u>: n is the number of layers of SMA wire in SMA actuator

Substitute Equations 4.12 to 4.11, the M_{θ_2} derived to Equation 4.13.

$$\left(\frac{(nF_{SMA}(\cos 2.17 + \cos 8.18) \times d_1) - (F_{S1} \cos 4.24) \times d_3}{d_2}\right) \times L_c \cos 20 + (mg \cos 70) \times L_{cm} - (F_{S2} \sin 20) \times L_{S2}$$

(Equation 4.13)

Substitute the torque of the PIP joint θ_2 in Equation 4.13, the number of layers *n* of SMA wire in SMA actuator is 10. Figure 4.11 shows only a single layer with 1 loop/route SMA wire and Figure 4.13 shows two layers with 1 loop/route.

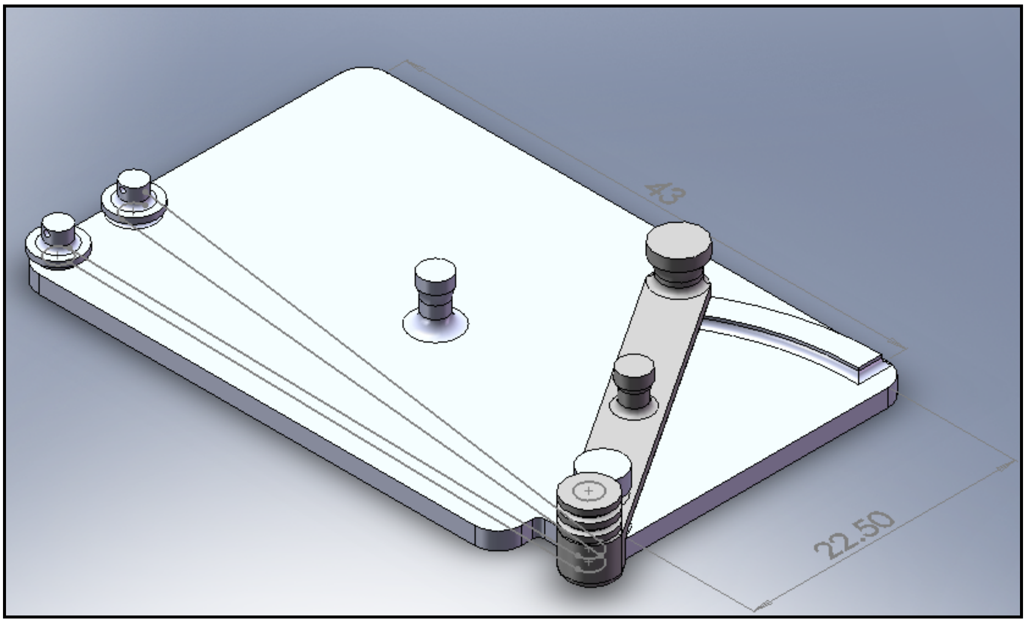


Figure 4.13: Two layers of SMA wires on SMA actuator.

4.3.3 Experiment of Result

The experimental result is based on the preliminary SMA actuator design (Figure 4.14) of a single layer with one loop/route 0.012” (300μm) SMA wire. The maximum deformation of the SMA wire in Figure 4.14 is 4.5%, and the distance between the tendon cable connector and SMA wire to the pivot is 17.83mm (d_2) and 4.5mm (d_1), respectively. It also means that the maximum linear displacement ΔL of the output lever is 8.0mm.

This SMA actuator is connected to a frame, as shown in Figure 4.15, and different bias load will ‘hang’ on the ‘bias load connector’. The experimental results are shown in Figures 4.16 to 4.20.

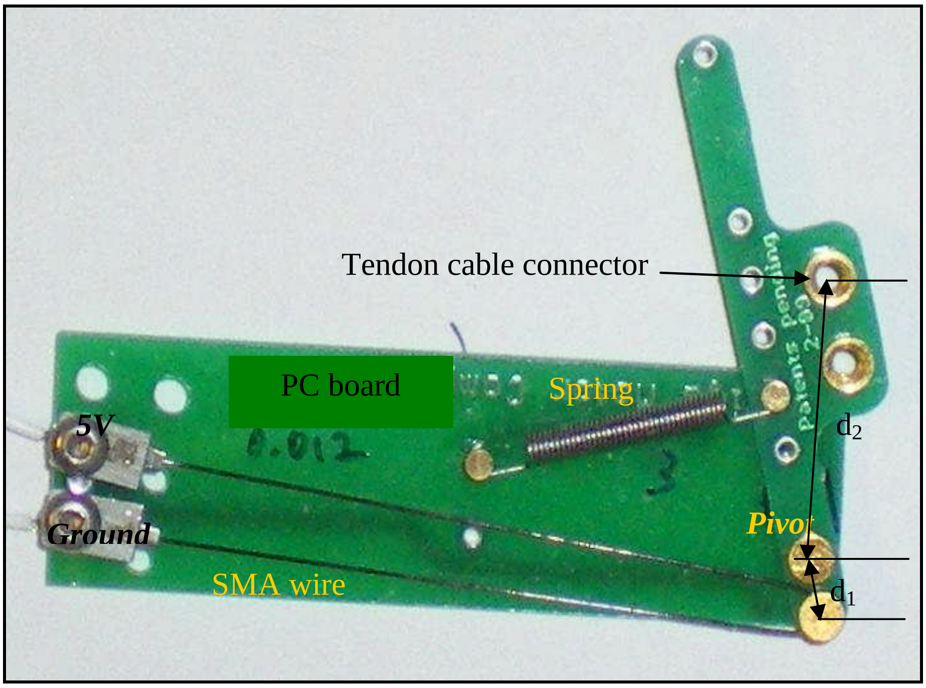


Figure 4.14: Preliminary design of single layer SMA wire actuator.

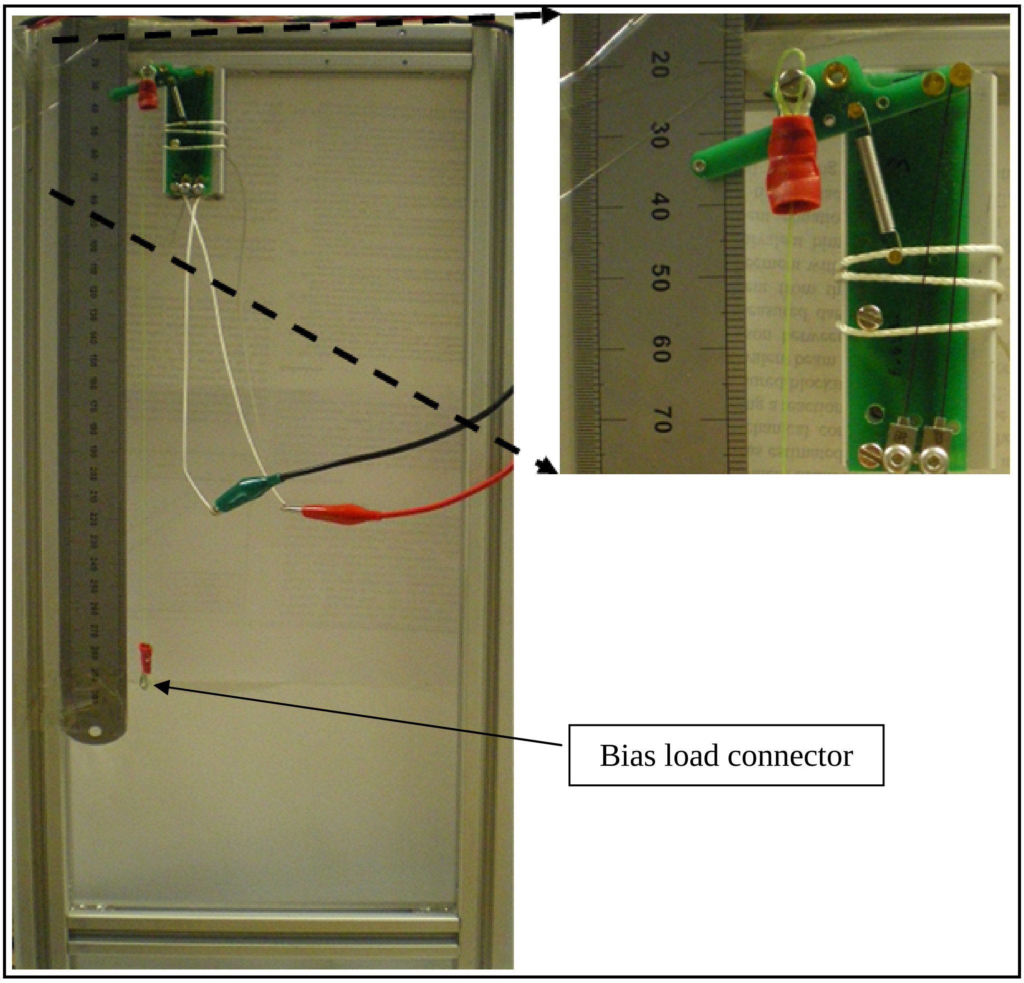


Figure 4.15: Frame for Preliminary experiment.

Different supply voltages from 0.1volt to 2.1volt and different bias loads from 0g to 350g will be used in order to get their relationship between the linear displacement of the output lever of the SMA actuator.

Due to the Flexinol SMA wire property, the safety range for using 0.012” SMA wire is below 1.75amp. Therefore, the maximum voltage is set to 2volts (total length of SMA wire used in Figure 4.14 is 90mm (45mm x 2) and total resistance is 1.2ohm before contraction and 1.13ohm after contraction).

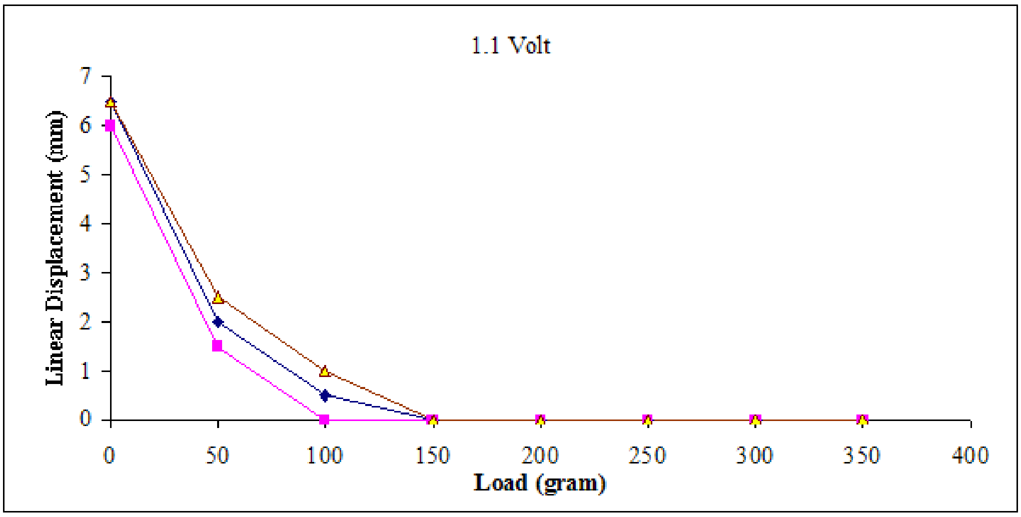


Figure 4.16: Relationship between linear displacement of output lever of SMA actuator and different supply loads for 1.1V supply voltage (0.92amp).

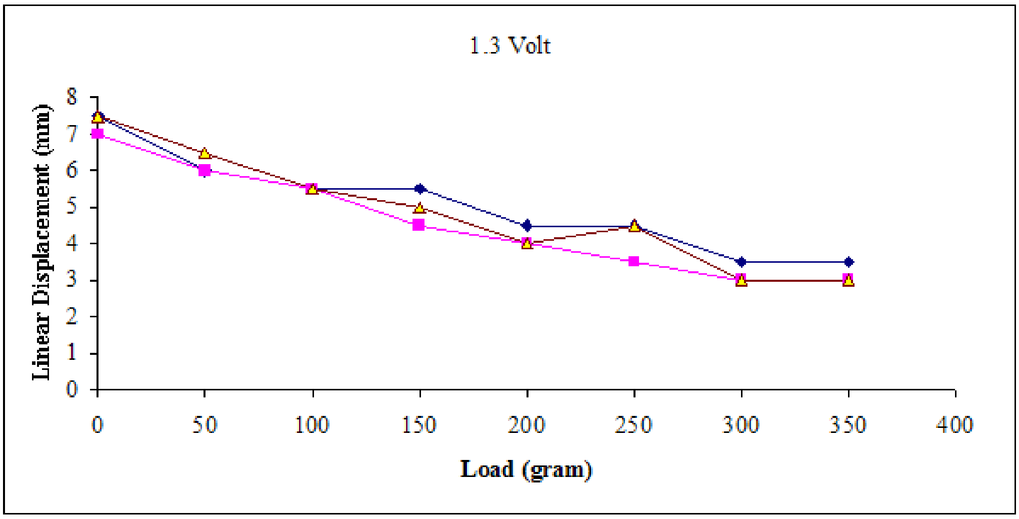


Figure 4.17: Relationship between linear displacement of output lever of SMA actuator and different supply loads for 1.3V supply voltage (1.08amp).

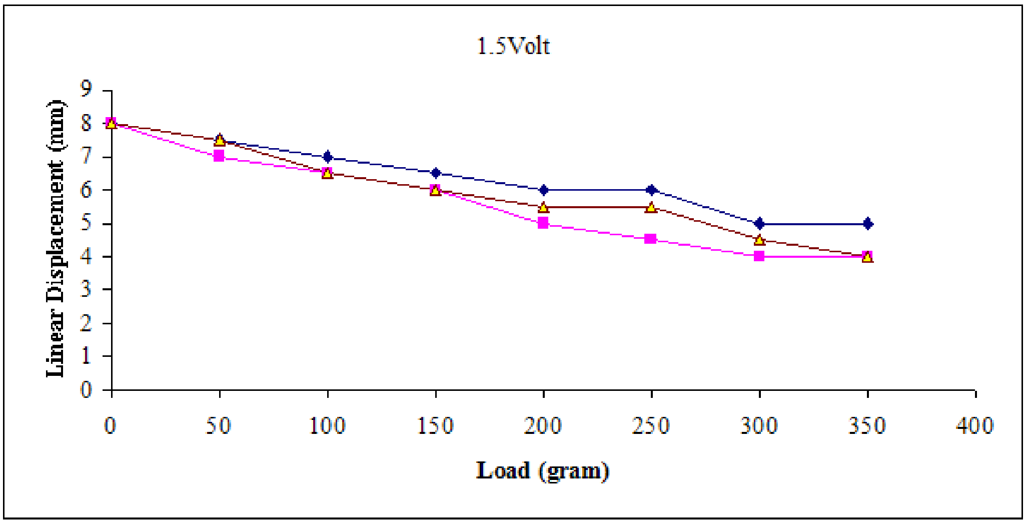


Figure 4.18: Relationship between linear displacement of output lever of SMA actuator and different supply loads for 1.5V supply voltage (1.25amp).

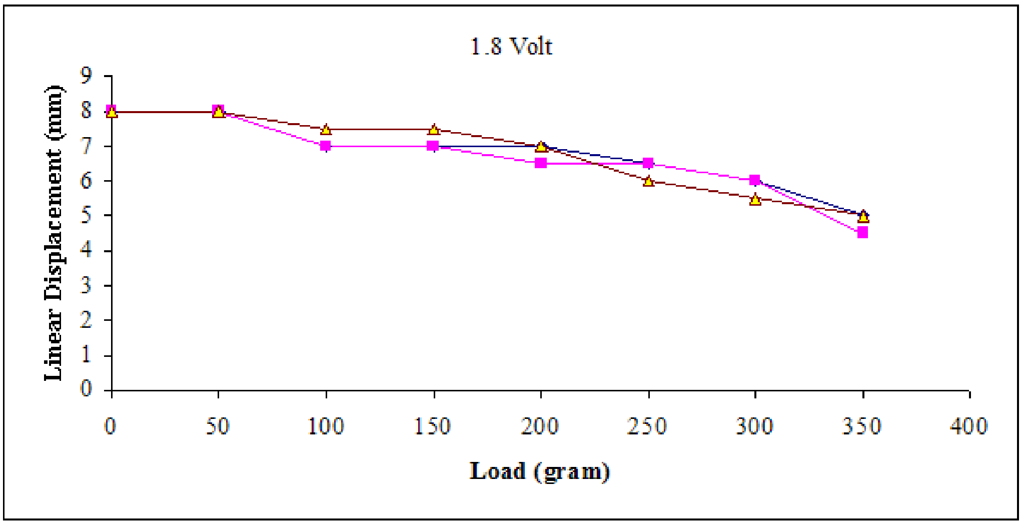


Figure 4.19: Relationship between linear displacement of output lever of SMA actuator and different supply load for 1.8V supply voltage (1.50amp).

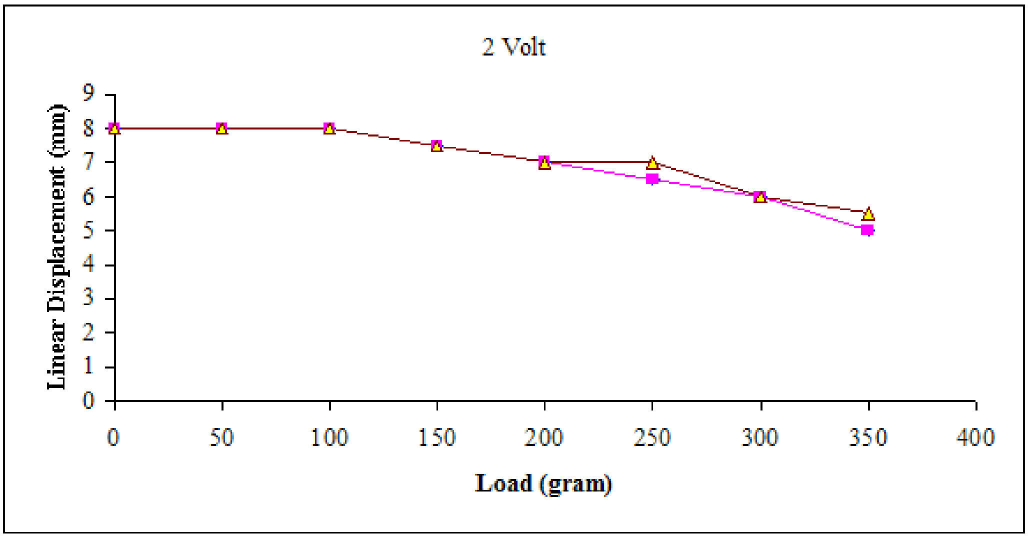
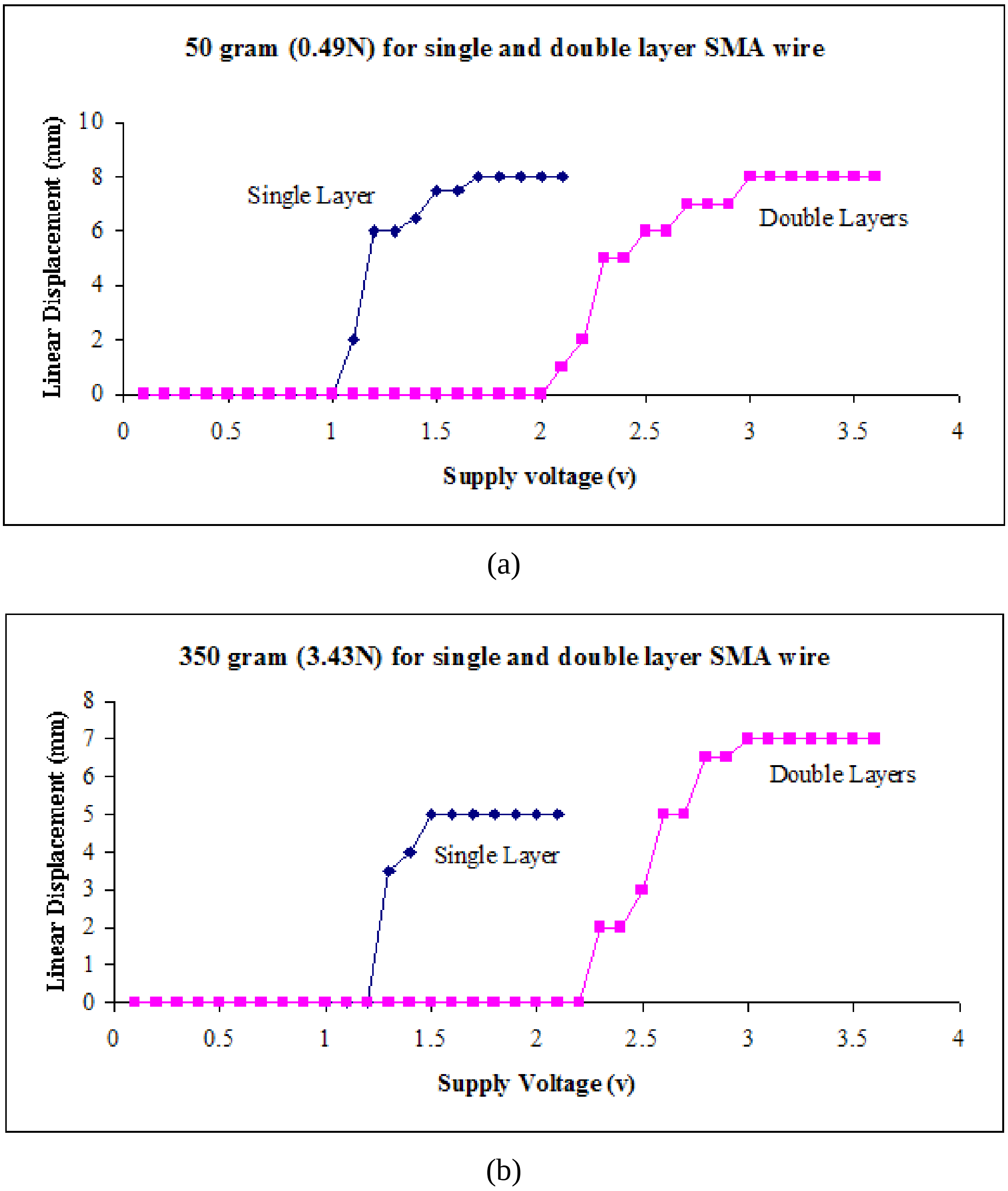


Figure 4.20: Relationship between linear displacement of output lever of SMA actuator and different supply loads for 2.0V supply voltage (1.67amp).

Three trials are carried out in Figures 4.16 to 4.20. These figures illustrate if the supply voltage is constant, higher bias load supplied will achieve lower linear displacement of SMA actuator. If bias load is constant, higher linear displacement of SMA actuator achieved in higher voltage supplied. However, the supply voltage or current cannot be over the limit recommended for the Flexinol wire properties, in order to avoid SMA wire destruction or reducing the number of cycles. (Note: The displacement is achieved at Austenite finish (Af) phase and the displacement from 0.1volt to 1.0volt for different load is zero).

Figure 4.21 shows the linear displacement relationship between single layer and double layer SMA wire used for 50g (0.49N) and 350g (3.43N), respectively. The linear displacement of the output lever of SMA actuator can reach up to 7mm in double layer SMA wire even higher than the single layer. As shown in Figure 4.21 (b). The reason for the SMA wire force's being increased that the wires are connected in parallel. The maximum displacement still stick on 8mm, as shown in Figure 4.21 (a), because of the current flow is connected in series.



(a)

(b)

Figure 4.21: Linear displacement relationship between single and double layer for (a) 50g and (b) 350g bias load in different supply voltage.

The supply voltage in the experiment for double layer SMA wires used is from 0.1 to 3.6v as the resistance of the SMA wire was increased due to increasing the length of the SMA wire from 90mm to 180mm, as shown in Figure 4.22. The figure shows two layers with two loop/route SMA wires, and each layer is 90mm (45mm x 2) before contracted. It means the SMA wire is connected in parallel but the current supply is in series.

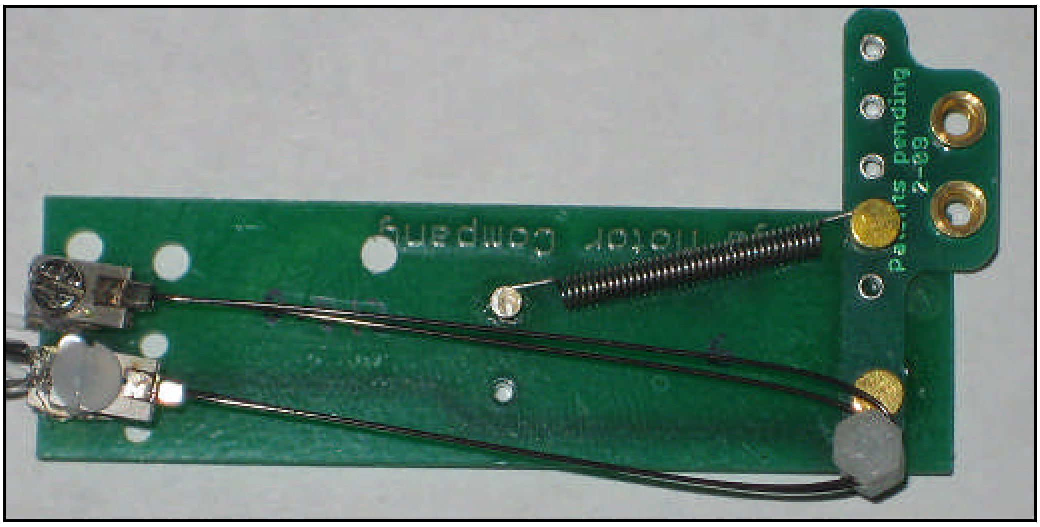
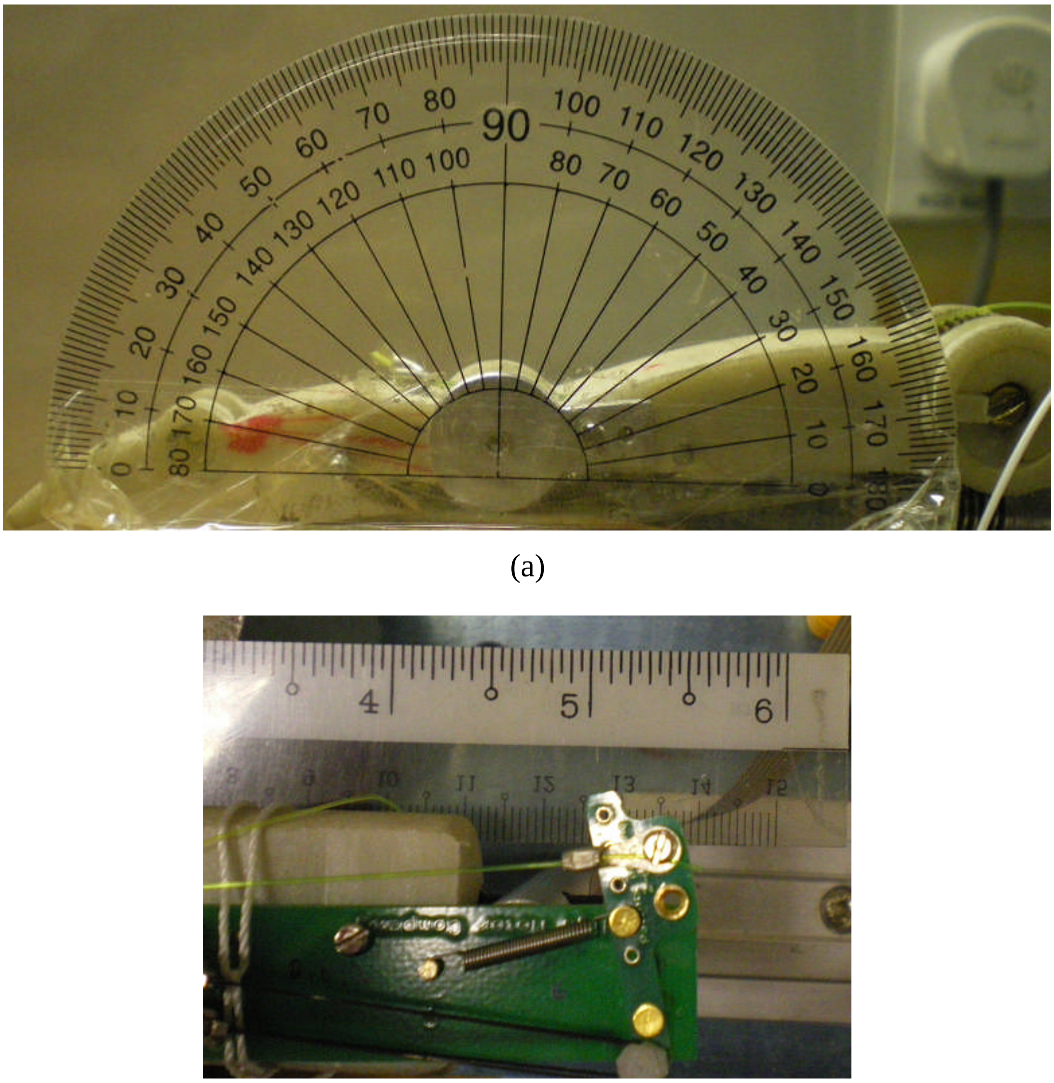


Figure 4.22: Preliminary design of double layer SMA wire actuator.

The double layer SMA wire actuator is connected to the finger, as shown in Figures 4.23 and 4.24 and defines the linear displacement of the SMA actuator and the rotational displacement of the finger PIP joint θ_2.

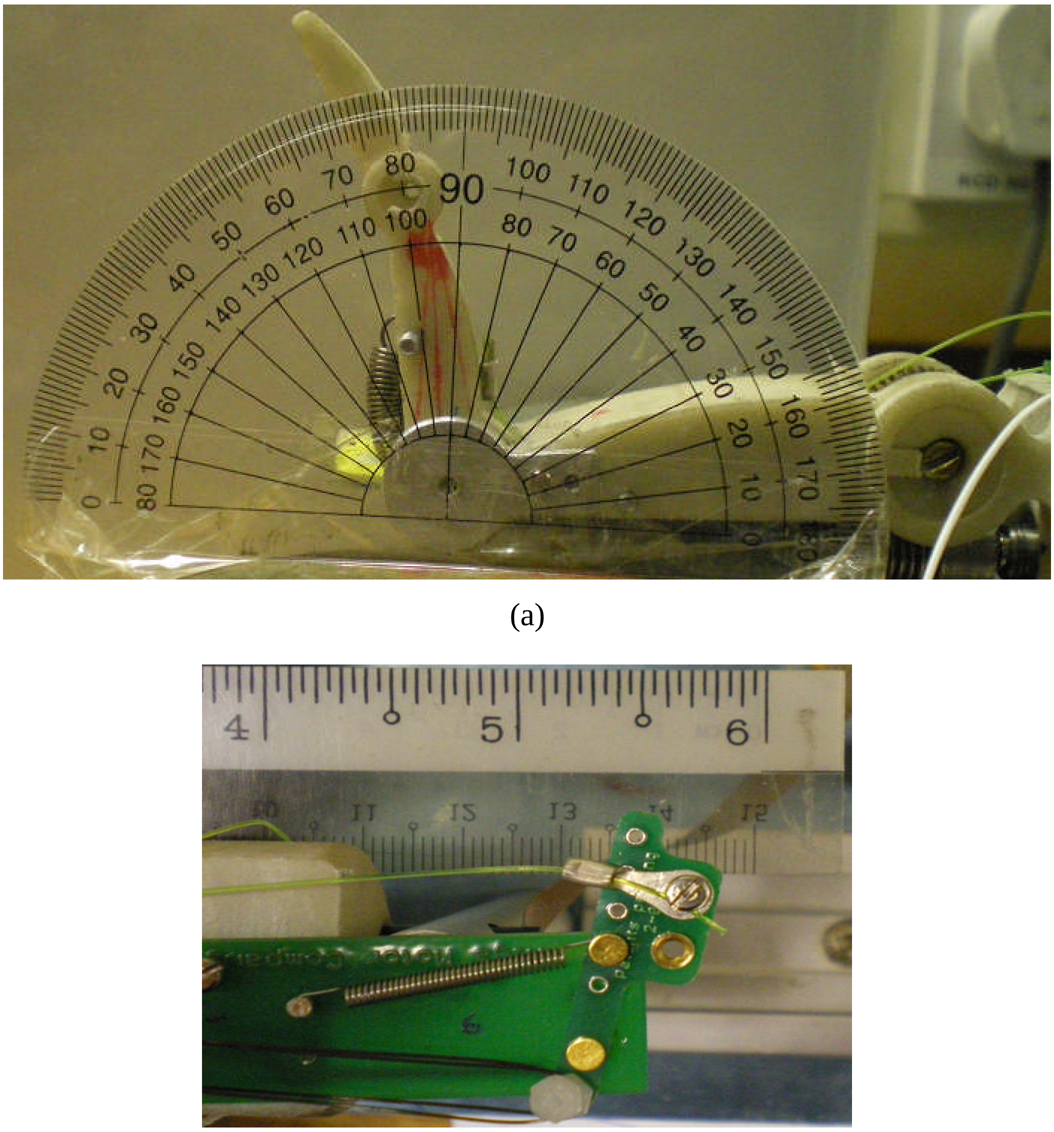
3v supply voltage is supplied to the double layer SMA wire actuator (Figure 4.22), which means 1.25amp current will flow to the 2.4ohm SMA wire. The 3v supply will bring the linear displacement of the SMA actuator from 13.5mm to 14.3mm (ΔL =8mm), as shown in Figures 4.23 (b) and 4.24(b). This displacement will bring the PIP joint θ_2 from 0^0 to 80^0, as shown in Figures 4.23 (a) and 4.24 (b). By Equation 4.6 and Figure 4.10, it is proved that the results achieved are reasonable and correct.



(a)

(b)

Figure 4.23: Double layer SMA wire actuator connected to finger at start position, (a) finger, (b) SMA actuator.



(a)

(b)

Figure 4.24: Double layer SMA wire actuator connected to finger at final position, (a) finger, (b) SMA actuator.

4.5 Summary

This chapter has focused on the forces/torques of each joint and phalanx needed to maintain enough grasping force by using hybrid actuation mechanism. The specification for designing this hybrid actuation mechanism prosthetic finger has been presented.

A suitable DC motor was used in our design and placed in the palm to control the flexion-extension of the MCP joint, as this joint needs more force than the PIP and

DIP joints [58]. From the design, it is known that the DC actuator can produce enough torque for the MCP joint through a suitable worm gear.

The SMA actuator is associated with a tendon cable and a spring to control the flexion-extension of PIP joint. Two challenges for designing this actuator were:

> Design the smallest size possible to achieve a large linear stroke or displacement for large angular motion. Principle of leverage was used for this application.
>
> Produce sufficient output pull force for grasping, at least 10 layers of SMA wire was used and connected in 1-loop. The current supplied is in series but the SMA wire pull force is in parallel.

From the experimental results for the SMA actuator, a higher supply voltage or current will give larger linear displacement of the SMA actuator; however, the supply voltage or current must be in the range of the recommended range. If the supply voltage is too large, it will reduce the number of cycles for which SMA wire can be used and possibility it will destroy the wire.

CHAPTER 5

Control System Design of Actuation Mechanism

5.1 Introduction

A control system is a device to manage, command, direct or regulate the behaviour of other devices or systems. The system is designed to control the hybrid actuation mechanism in order to position the finger, especially fingertip, correctly. A PID algorithm was used to control the movement of the DC actuator to 'bring' the MCP joint to the correct position. A simple open loop with the value of the input voltage and the output linear displacement change of the SMA actuator was used to control movement of the PIP joint.

Sections 5.2 and 5.3 will briefly discuss the design of the control for the DC and SMA actuators. Section 5.2 contains three subsections, which are actuator's plan estimation by using system identification, PIP control design, and DC actuator deployment. The time response and simple open loop control are the two subsections for section 5.3.

5.2 DC Actuator

There are three processes used to design the DC actuator control system. Firstly, modelling is used to identify a mathematical representation of the 'plant', the real-world, physical system that is to be controlled. The process of determining the equations governing the model's dynamics is called system identification. This is one of the techniques used to identify a mathematical representation of the plant based on a set of real-world stimulus (input signal) and response (output signal) data samples [64]. Control design is the second process and involves choosing a method of control and designing a controller. Deployment is the last stage in designing a control system that implements the finalised system. The general block diagram of the DC actuator control system is shown in Figure 5.1. The input and output of the figure can be any parameter, such as position, speed, etc..

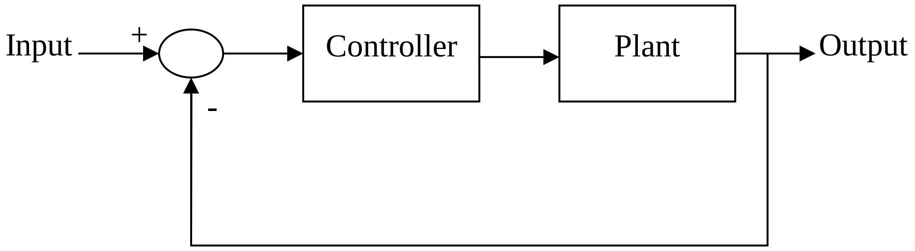


Figure 5.1: General block diagram of closed loop control system.

A DC actuator is placed in the palm to control the 90^0 flexion-extension movement of the MCP joint. Positioning the finger accurately needs a controller. In the initial phase of the design process, a mathematical model of the plant to be controlled must be obtained. One way to do this is by using a numerical process known as system identification. This involves acquiring data from a plant followed by numerical analysis of stimulus and response data to estimate the plant parameters.

5.2.1 Modelling – System Identification

System identification is a tool used to model a plant mathematically for control system design. The general system identification process might include the following stages [65] and Figure 5.2 demonstrates the typical system identification flowchart (Full programming except stage 1 is shown in Appendix A, Figure A.1):

1. Experimental design and data acquisition/collection.
2. Data analysis and pre-processing, including plotting data, removing offsets, filtering and selecting regions of interest.
3. Estimation and validation of models.
4. Model transformation and analysis, such as linear analysis, reducing model order and converting between discrete-time and continuous-time representations.

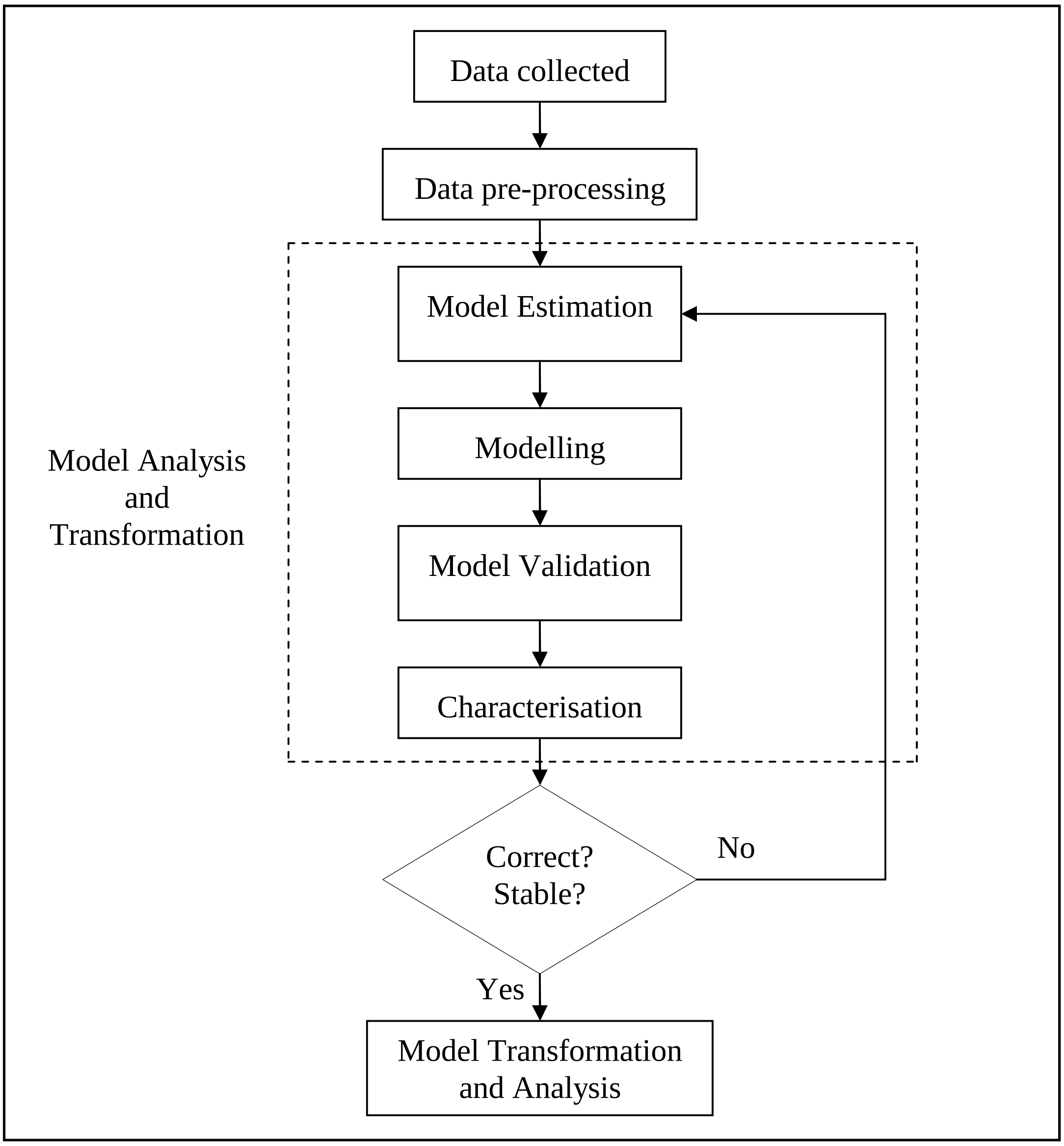


Figure 5.2: Typical system identification flowchart.

5.2.1.1 Acquiring and Pre-processing Data

The first step in identifying an unknown system is data acquisition. The raw input voltage and output angular velocity of the model are collected using a NI DAQ_mx card (NI PCI-MIO-16E-4) and a program developed in a LabView environment. The plant model is DC actuator connected to the MCP joint of the finger. In order to create a linear model, the output signal is choosing angular velocity to identify the model. Figure 5.3 shows the relationship between the input voltage and output angular velocity.

Since the output angular velocity contains some noise, as in Figure 5.4 (Left), a 2Hz lowpass cut-off frequency of Butterworth 1st order is used to filter out the noise, as in Figure 5.4 (Right). The stimulus and respond signals (to identify the model) use the input filtered voltage and the output filtered angular velocity shown in Figure 5.4(Left).

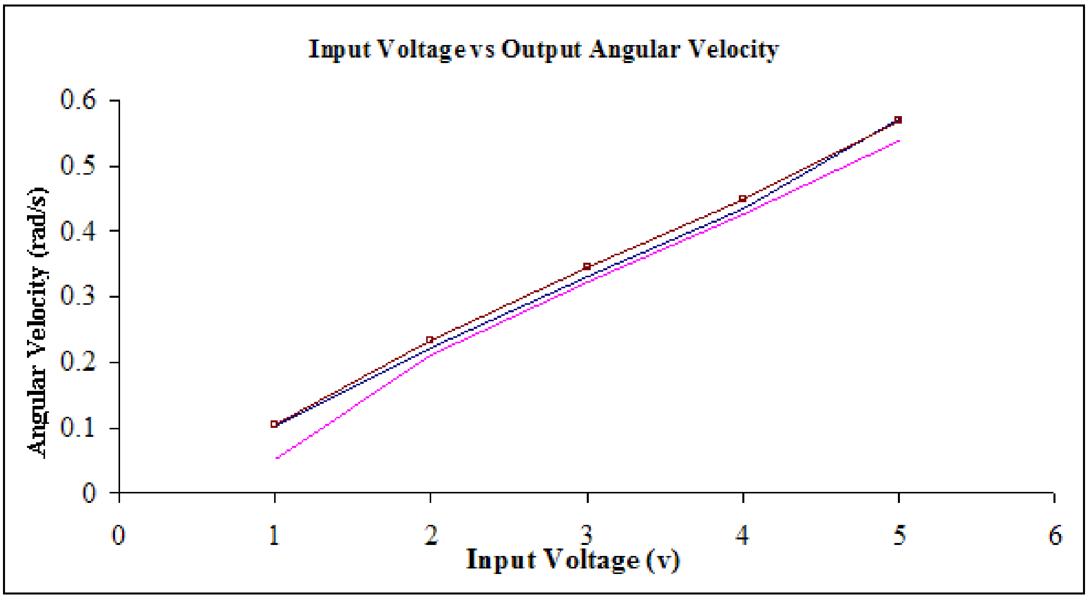


Figure 5.3: Input voltage versus output angular velocity (three trials).

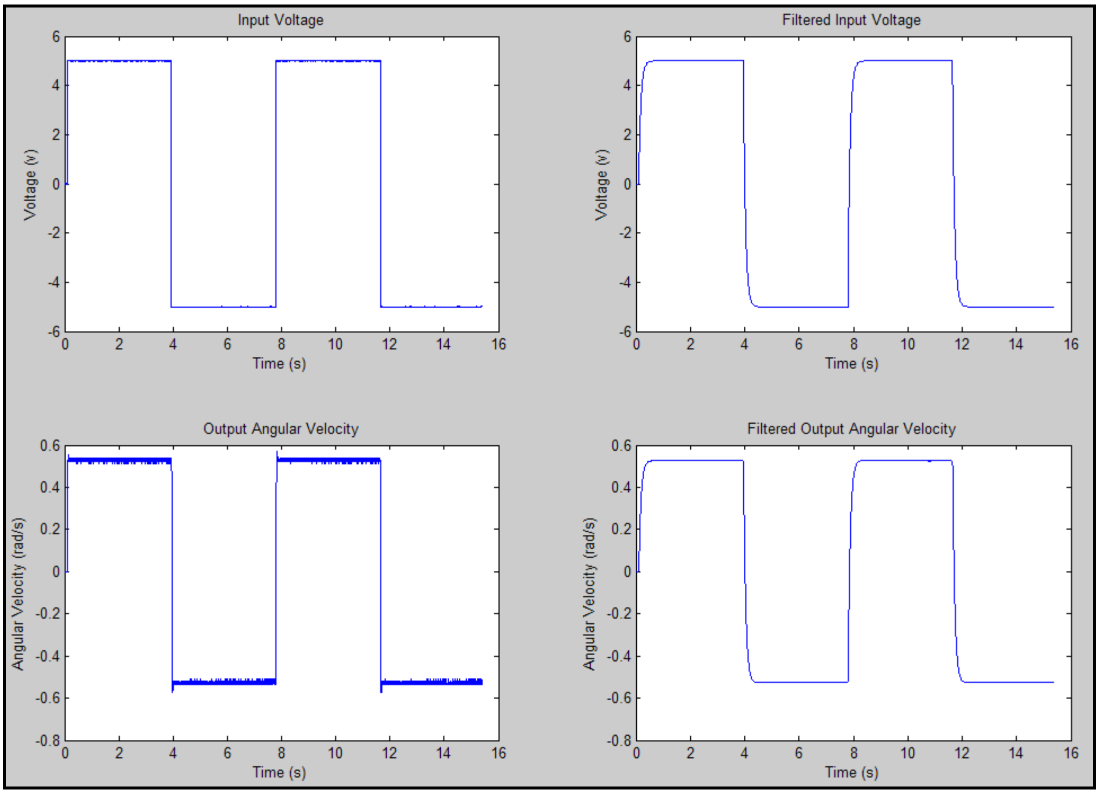


Figure 5.4: (Left) Input voltage and output angular velocity, (Right) Input filtered voltage and output filtered angular velocity.

5.2.1.2 Estimation and Validation of Model

After the stimulus and response signals are collected and pre-processed, the parametric model estimation method is used to estimate the model. One of the biggest challenges in model estimation is selecting the correct model. There are two categories of parametric models in a LabView system identification toolkit, polynomial and state-space.

1. Choosing Model

Polynomial category is used rather than state-space as the control system is single-input-single-output (SISO). The polynomial category is simpler and easier than state-space, but state-space models are often preferable to polynomial, especially in modern control applications that focus on multivariable systems [64].

The following equation describes a general-linear polynomial model and Figure 5.5 shows the signal flow.

$$A(z)y(k)=\frac{z^{-n}B(z)}{F(z)}u(k)+\frac{C(z)}{D(z)}e(k)=\frac{B(z)}{F(z)}u(k-n)+\frac{C(z)}{D(z)}e(k) \quad \text{(Equation 5.1)}$$

where: $A(z)=1+a_1z^{-1}+a_2z^{-2}+....+a_{k_a}z^{-k_a}$

$$B(z)=b_0+b_1z^{-1}+b_2z^{-2}+....+b_{k_b-1}z^{-(k_b-1)}$$
$$C(z)=1+c_1z^{-1}+c_2z^{-2}+....+a_{k_c}z^{-k_c}$$
$$D(z)=1+d_1z^{-1}+d_2z^{-2}+....+d_{k_d}z^{-k_d}$$
$$F(z)=1+f_1z^{-1}+f_2z^{-2}+....+f_{k_f}z^{-k_f}$$

z^{-n} defines the number of delay samples between the input and output and the k_a, k_b, k_c, k_d, and k_f are the model orders.

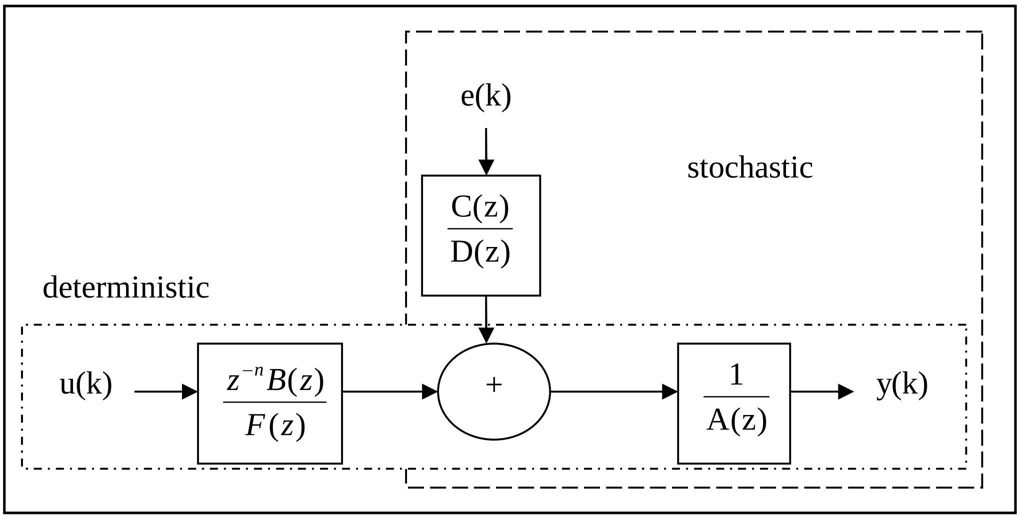


Figure 5.5: Signal flow of general-linear polynomial model [64].

The general-linear polynomial model is a combination of the deterministic and stochastic parts of the system. The former specifies the relationship between the output and the input signal and the latter specifies how the random disturbance affects the output signal. Often these parts of a system are referred to as system dynamics and stochastic dynamics, respectively [64].

When C(z), D(z) and F(z) settings are equal to 1, it can create simpler models, such as an autoregressive with exogenous (ARX) model. When D(z) and F(z) are equal to 1, it can create an autoregressive-moving average with exogenous (ARMAX) model. When A(z), C(z) and D(z) are equal to 1, it can create an output-error model. When A(z) is equal to 1, it can create a Box-Jenkins Model. When B(z), C(z), D(z) and F(z) are equal to 1, it can create an autoregressive (AR) model [64]. Table 5.1 describes the characteristics of each different model and ARMAX model is decided to choose. Figure 5.6 and Equation 5.2 show the model of the ARMAX.

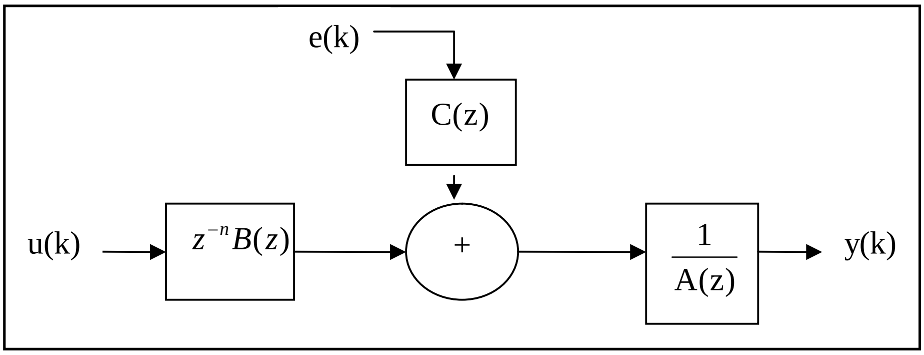


Figure 5.6: ARMAX model [64].

$$A(z)y(k) = z^{-n}B(z)u(k) + C(z)e(k) = B(z)u(k-n) + C(z)e(k) \quad \text{(Equation 5.2)}$$

Table 5.1: Characteristics of different models.

Model	Parameters equal to 1	Characteristics
ARX (Autoregressive with exogenous terms)	C(z), D(z) & F(z)	Transfer function of deterministic and stochastic parts of system has same set of poles. This coupling can be unrealistic. System and stochastic dynamics of a system do not share same set of poles all the time.
ARMAX (Autoregressive-moving average with exogenous terms)	D(z) & F(z)	Useful when dominating disturbance entering early in the process, for example at the input. More flexible than ARX model in handling models containing disturbances. Prediction error is identification method of output-error model.
Output-Error	A(z), C(z) & D(z)	Describes system dynamics separately from stochastic dynamics. Does not use any parameters for simulating disturbance characteristics.
Box-Jenkins	A(z)	Represents disturbance properties separately from system dynamics. Useful when disturbances enter late in the process, such as measurement noise on the output.
AR (Autoregressive)	B(z), C(z), D(z) & F(z)	Does not include dynamics between input and output. Therefore, more suitable for representing signals rather than a system because a system typically has input and an output.

2. Order of Plant Model

Since an ARMAX model was chosen in the previous section, another challenge in model estimation is the correct model order selection. This requires two data sets, one for estimation and one for validation. These two data sets can partition the original set of data into two, using the SI split signal VI. The VI divides the pre-processed stimulus-response (input-output) data samples into two portions, one for model estimation and the other for model validation, shown in Figure 5.7.

The data of the estimation portion are used to create the model of the plant whereas the data of the validation portion are used to predict 1-step or multi-step mean square error (mse). Figure 5.8 shows the current and 1-step ahead validated prediction signal, with mse value 5.0835e-06.

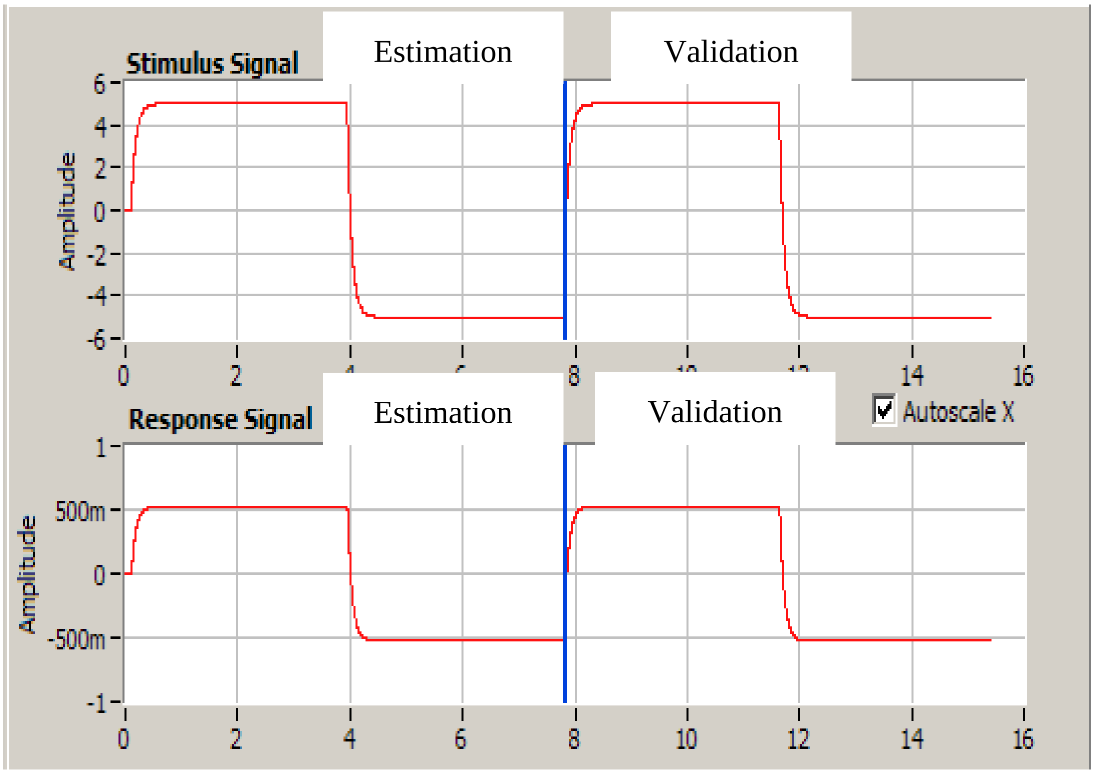


Figure 5.7: Stimulus and response signal splitter.

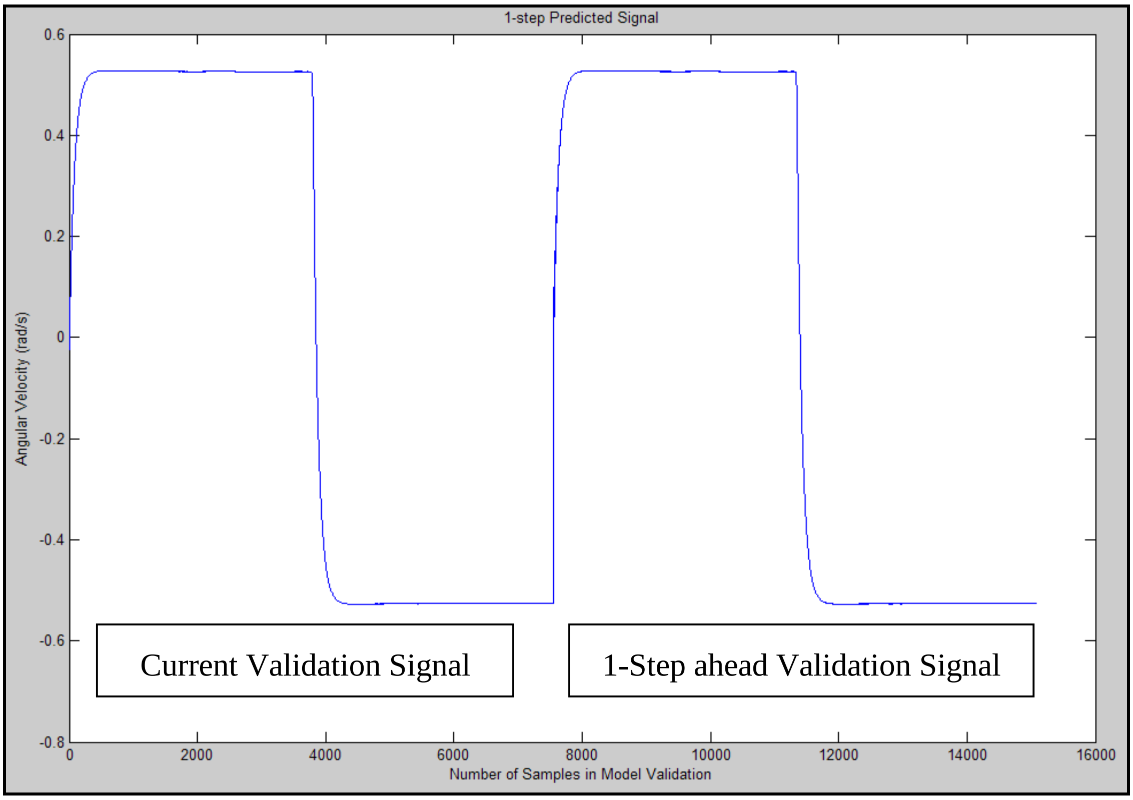
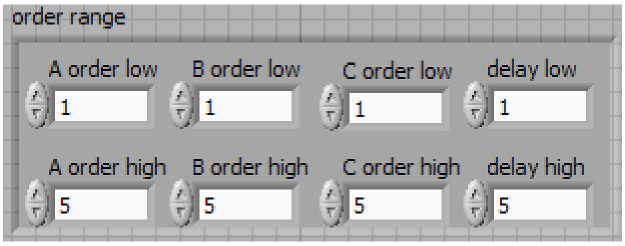


Figure 5.8: 1-Step predict signal.

There are three steps to help to choose the correct model order. The first step is to set a range of numbers for A(z), B(z), C(z), and delay the estimation portion of data, as shown in Figure 5.9. Figure 5.10 shows a prediction error plot generated by the SI Estimate Order of System Model VI for an ARMAX model. The y-axis is the prediction error and the x-axis is the model dimension.



order range

A order low	B order low	C order low	delay low
1	1	1	1
A order high	**B order high**	**C order high**	**delay high**
5	5	5	5

Figure 5.9: Number of range for A(z), B(z), and C(z).

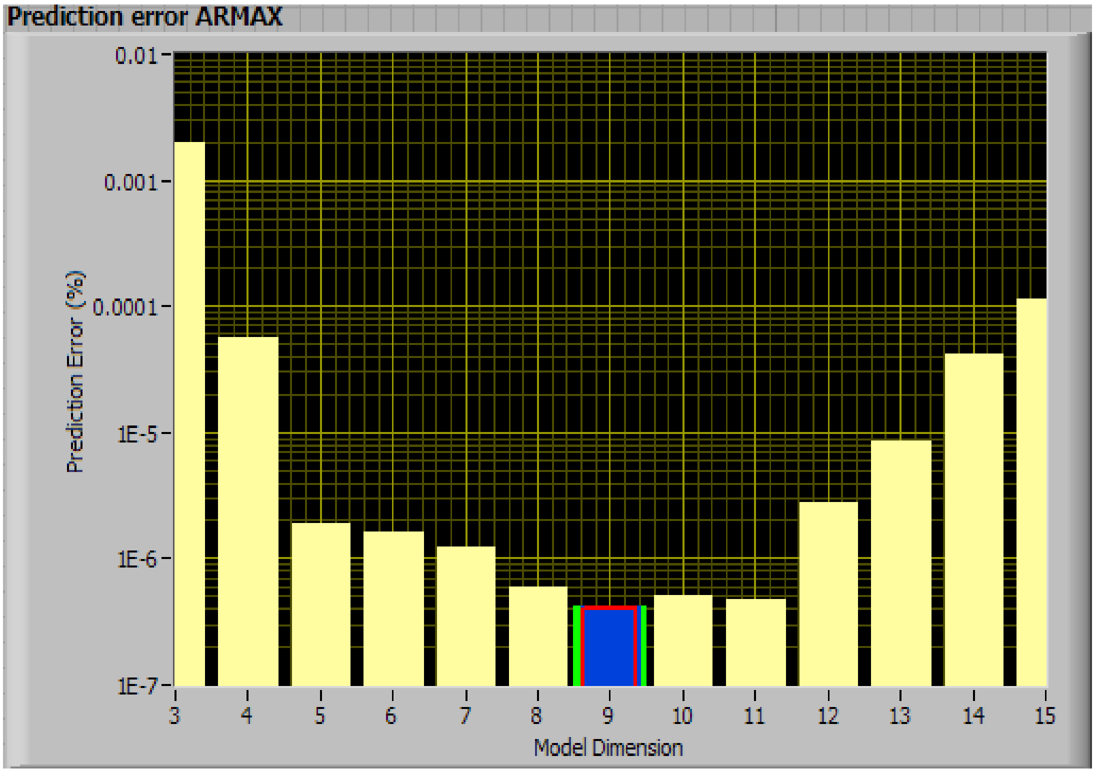


Figure 5.10: Prediction error plot.

The SI VI generates the prediction error plot for the ARMAX model and the optimal A, B, C and delay orders, which is on the lowest prediction error, as shown in the colour bar in Figure 5.10 for dimension 9. Hence, the number of parameters in the optimal order are A=5, B=3, C=1 and delay=1.

After determining the order of the model, the results must be verified to ensure the model describes the plant model with accuracy and stability. The second step to identify the correct model order is to plot a pole-zero map. This step is verified if the model created is stable [64]. If a pole and zero overlap, they cancel out each other, indicating that the estimated optimal order is too high, as shown in Figure 5.11 (pole and zeros at real axis 1). If the pole is out of the unit circle, the model created is considered unstable [64]. The red colour is zero and the green colour is pole.

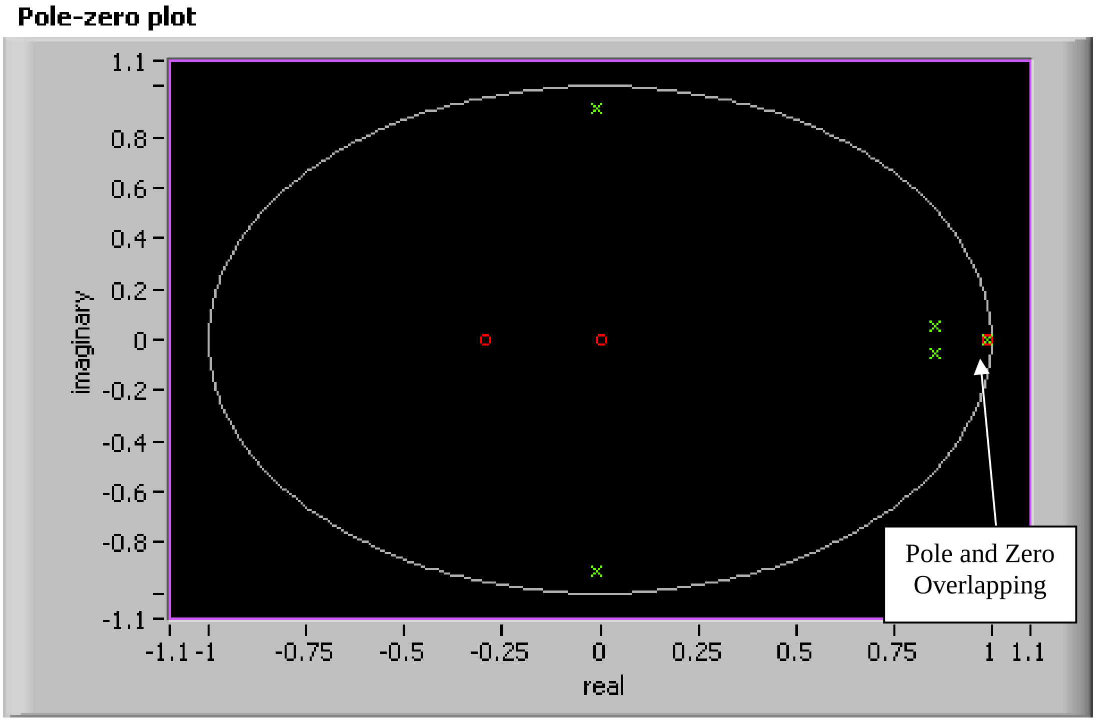


Figure 5.11: Pole-Zero map of A=5, B=3, C=1 and delay=1.

The last step is to choose the correct model order by plotting a residual analysis. This step is to verify whether the model created is accurate [64, 65] and the residual analysis tests whether the prediction error correlates to the stimulus signal. Figure 5.12 shows the model created at the parameters A=5, B=3, C=1 and delay=1, with cross-correlation of prediction error. The whole error signals are inside the confidence level of cross-correlation, which is 0.1423. This value is auto-achieved from the SI Model Residual Analysis VI. The confidence level corresponds to the range of residual values with a specific probability of being statistically insignificant for the system [64, 65].

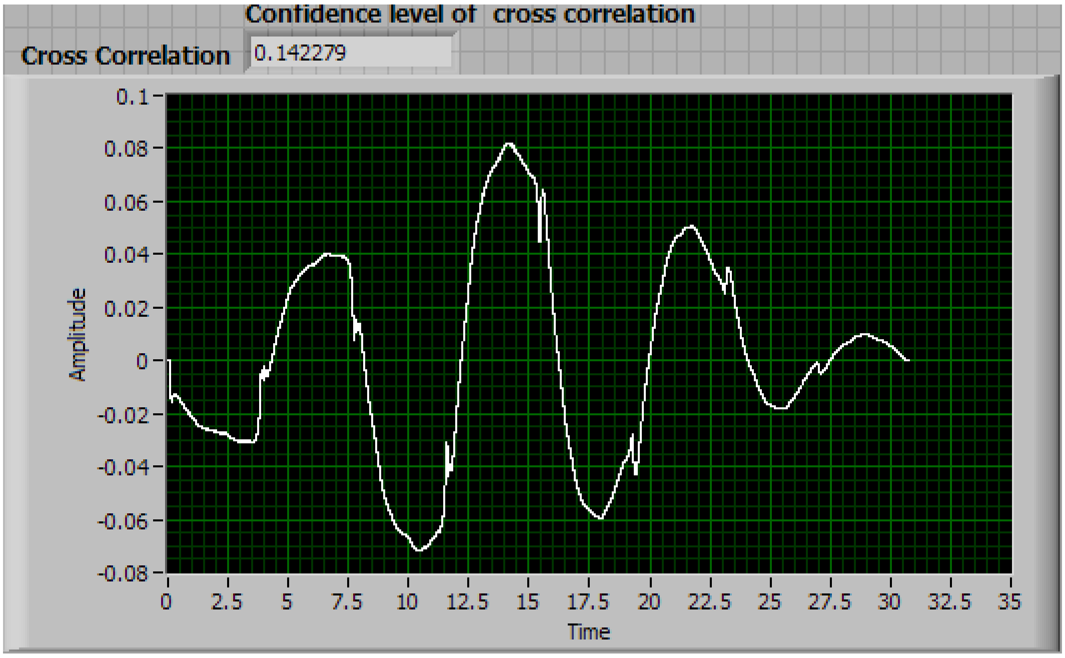
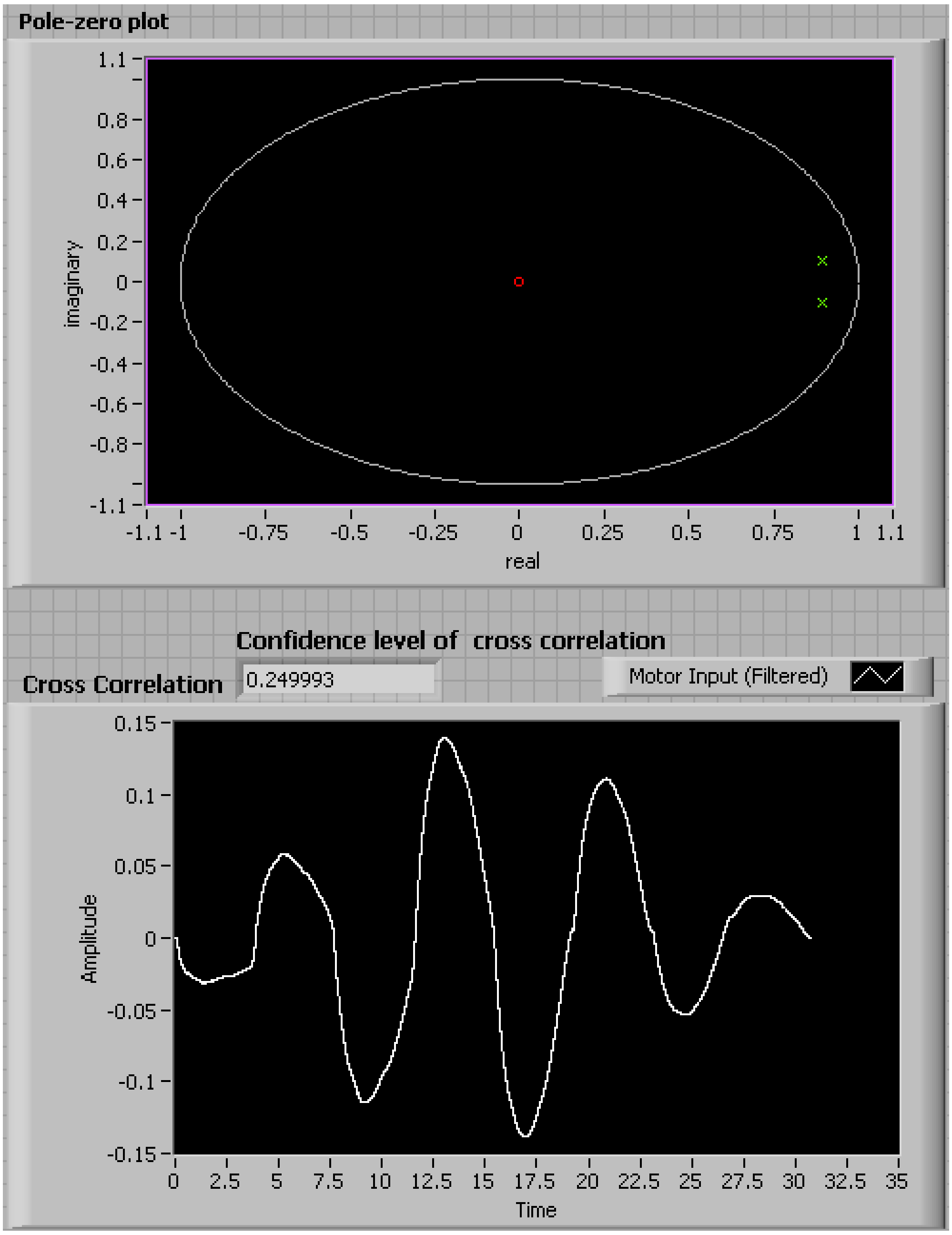


Figure 5.12: Residual analysis for A=5, B=3, C=1 and delay=1.

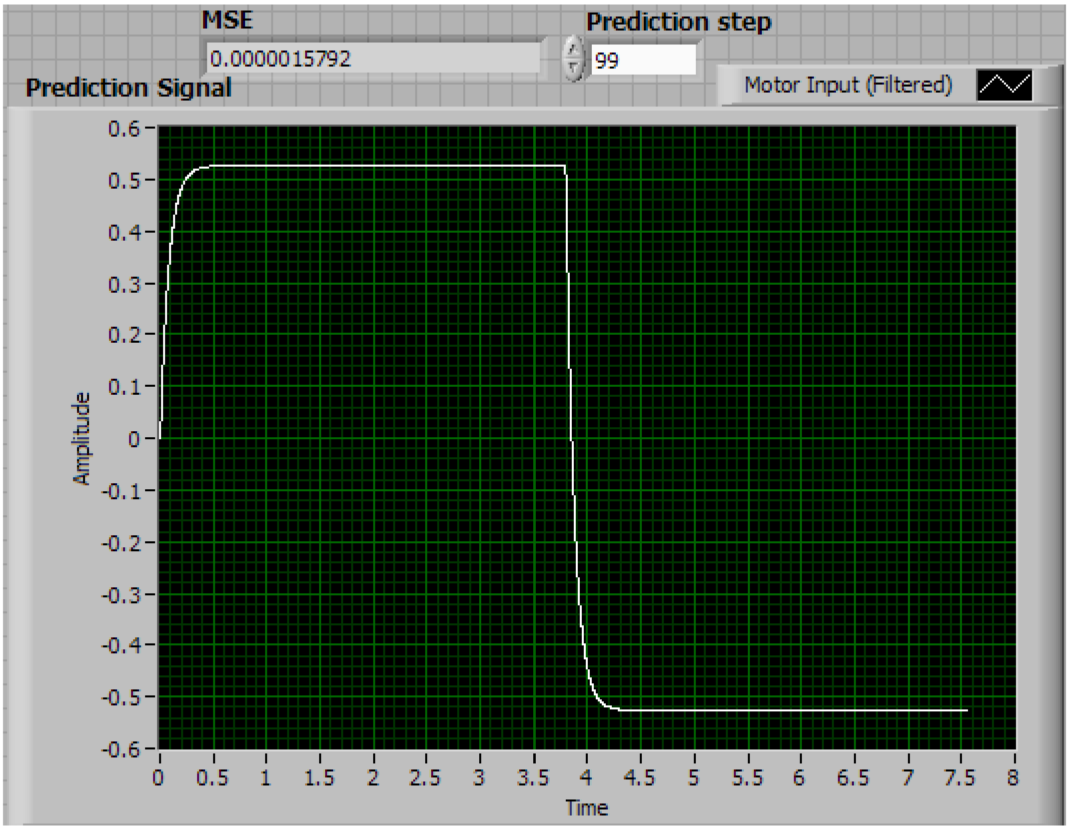
From the second and third steps above, we can conclude that the model created in parameters A=5, B=3, C=1 and delay=1 is accurate, but unstable.

Consequently, another model is created with parameters A=2, B=1, C=1 and delay = 1. This plant's model is accurate and stable since the pole is inside the unit circle with no overlapping and the whole error signal is inside the confidence level, as shown in Figure 5.13. Figure 5.14 shows the prediction plot at 99-step and its MSE is 1.5792e-6.



Confidence level of cross correlation: 0.249993

Figure 5.13: Pole-zero and residual analysis for A=2, B=1, C=1 and delay=1.



MSE 0.0000015792

Prediction step 99

Figure 5.14: Prediction plot at 99-step (Note: overlap of estimation and prediction).

5.2.1.3 Model Transformation and Analysis

Since we are using polynomial, the transfer function formed is a discrete transform:

$$(1-1.7882z^{-1}+0.8095z^{-2})y(k)=0.0022z^{-1}u(k)+(1+0.9716z^{-1})e(k) \quad \text{(Equation 5.3)}$$

Referring to Equations 5.1 and 5.2, we know the values of $A(z)=1-1.7882z^{-1}+0.8095z^{-2}$, of $B(z)=0.0022$ and of $C(z)=1+0.9716z^{-1}$.

Using the SI 'convert discrete to continuous model' VI, convert Equation 5.3 to continuous transfer function of the plant model. The continuous transfer function is shown in Equation 5.4. The value of $y(s)$ is the output angular velocity and the value of $u(s)$ is input voltage.

$$\frac{y(s)}{u(s)}=\frac{1.2016s+2486.04}{s^2+211.387s+23617.4} \quad \text{(Equation 5.4)}$$

This model plant clearly is a second order underdamped system, as shown in Figure 5.15, step response.

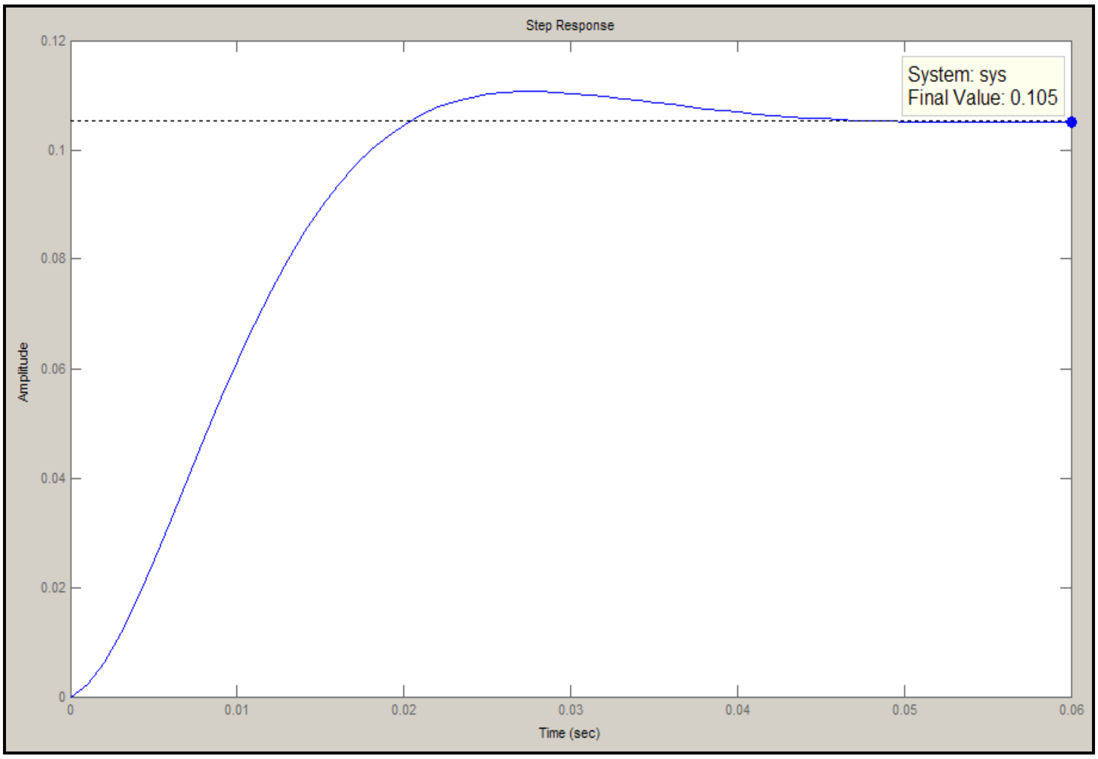


Figure 5.15: Step response of plant model.

The DC gain k, natural frequency ω_n, and damping ratio ξ can be found in the bode plot (Figure 5.16) and following Equations 5.5 to 5.7.

$$DC\_Gain(k) = 10^{\frac{M(0)}{20}} \quad \text{(Equation 5.5)}$$

$$Natural\_Frequency(\omega_n) = \omega_{-90^0} \quad \text{(Equation 5.6)}$$

$$Damping\_Ratio(\xi) = \frac{k}{2*10^{(\frac{M_{-90^0}}{20})}} \quad \text{(Equation 5.7)}$$

where: M(0) = Magnitude of bode plot when $j\omega=0$,

ω_{-90^0} = Frequency at which phase plot is at -90 degrees,

M_{-90^0} = Magnitude of bode plot when the phase is -90 degrees.

Hence, the values of DC gain k, natural frequency ω_n and damping ratio ξ of the created plant model are 0.1047, 163rad/s and 0.7310, respectively. Since the damping ratio ξ is below 1, the created plant model is underdamped second order. The plant model has one zero and two poles, as shown in Figure 5.17.

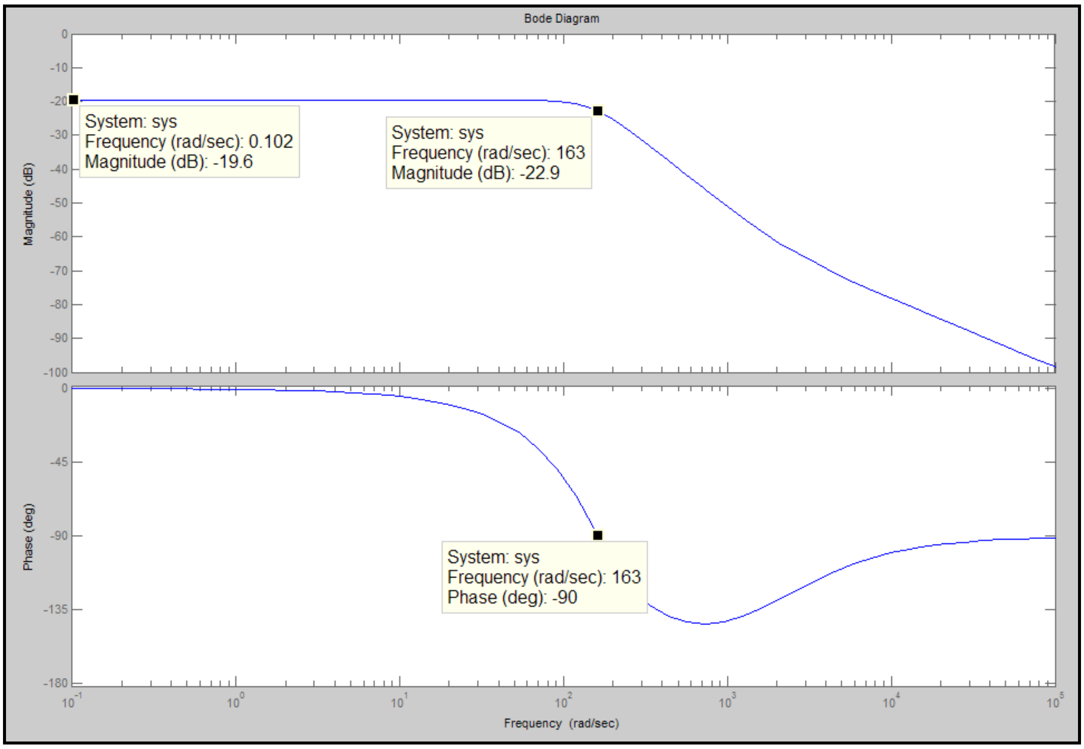


Figure 5.16: Bode plot of plant model.

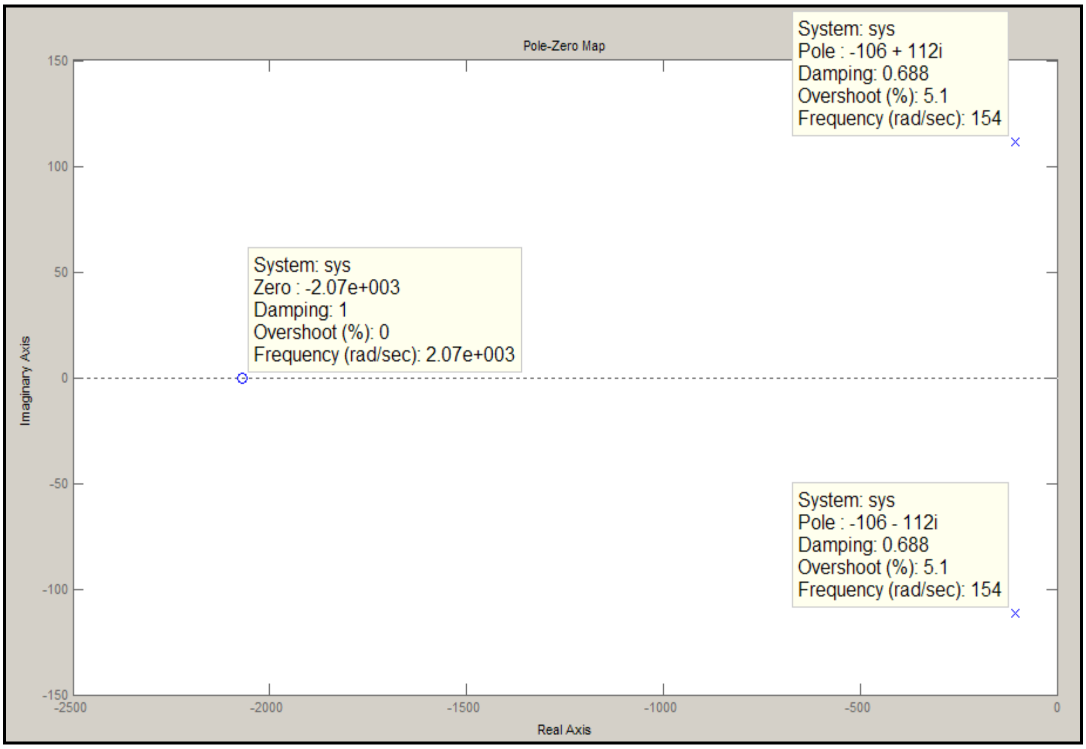


Figure 5.17: Pole-zero plot of plant model.

5.2.2 Control Design – Proportional Derivative

The PID control algorithm is used for the control of almost all loops in the process industries and is also the basis for many advanced control algorithms and strategies. To control the position of the finger, an integrator is needed to transform the angular velocity to an angular position, as shown in Figure 5.18. Since an integrator was used, the control algorithm of the system will become PD algorithm. In order for control loops to work properly, the PD loop must be properly tuned. The plant model of the Figure 5.18 is the DC actuator connected to the MCP joint of the finger.

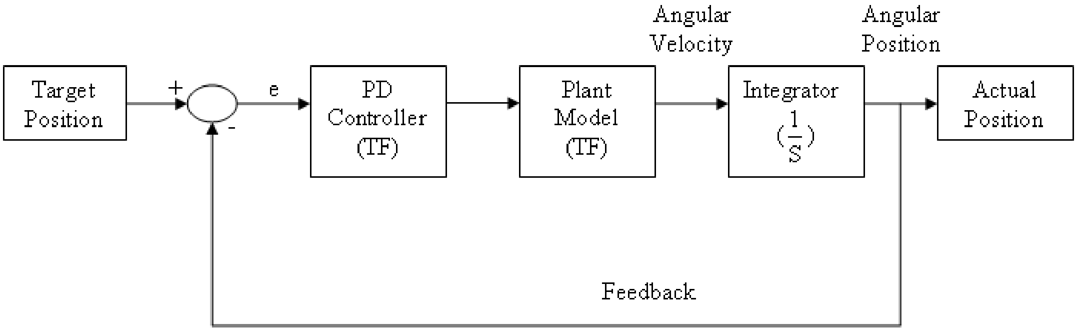


Figure 5.18: Block diagram of closed-loop system.

From Figure 5.18, the variable, e, represents the tracking error, the difference between the desired input value (Target Position) and actual output (Actual Position). This error signal, e, is sent to the PD controller, and the controller calculates the derivative of error signal. The output of the PD controller is now equal to the proportional gain $\mathrm{K_P}$, multiplied by the magnitude of the error plus derivative gain $\mathrm{K_D}$, times the derivative of the error, as shown in Equation 5.8 and Figure 5.19 [66].

$$PD\_Output = K_P e + K_D \frac{de}{dt} \quad \text{(Equation 5.8)}$$

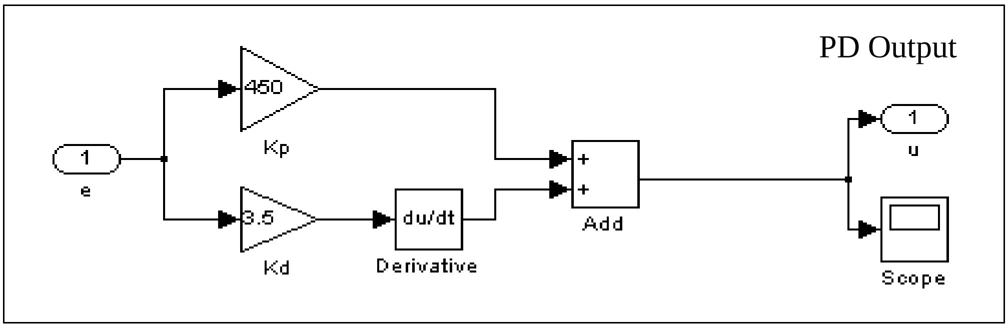


Figure 5.19: PD Controller.

The voltage of the PD output voltage is sent to the plant and the new output – actual position, is obtained. The new output is sent back to the sensor again to update the value of error e. The controller takes this updated signal and calculates its derivative and its integral again.

The characteristics of the proportional gain K_P and derivative gain K_D are manually tuned based on Table 5.2.

Table 5.2: Characteristics of P, I, and D controllers [67].

	Rise Time	Overshoot	Settling Time	Steady-State Error
K_P	Decrease	Increase	Small Change	Decrease
K_I	Decrease	Increase	Increase	Eliminate
K_D	Small Change	Decrease	Decrease	Small Change

The proportional gain K_P, will have the effect of reducing the rise time, but never eliminating the steady-state error. A derivative gain K_D will have the effect of increasing the stability of the system, reducing the overshoot and improving the transient response. Effects of each K_P and K_D of controllers on a closed-loop system are summarised in Table 5.2.

The control closed loop system's aims are: minimum rising time, minimum settling time, steady state error less than 1% and No overshoot.

The value of K_P and K_D are manually tuned until the above aims are reached. A typical procedure to turn a PID controller would be [66]:

1. K_P to 1, and K_I, K_D to zero. Keep increasing/decreasing K_P until response has some overshoot.

2. Modify K_D to make system faster and compensate overshoot.

Finally, the values of K_P and K_D are 450 and 3.5, respectively. Applying these values into the LabView programming, as shown in Appendix A (Figures A.2 and A.3), the transfer function of the desired closed-loop system is:

$$\frac{y(s)}{u(s)} = \frac{4.20556s^2 + 9241.86s + 1118720}{s^3 + 215.593s^2 + 32859.3s + 1118720} \quad \text{(Equation 5.9)}$$

where the value of $y(s)$ is the actual angular position of the MCP joint and the value of $u(s)$ is its target angular position.

From the step response (Figure 5.20) of the Equation 5.9 closed loop transfer function, the rise time, setting time, overshoot and steady-state error are 0.0438s, 0.0838s, 0.0000469% and 0, respectively.

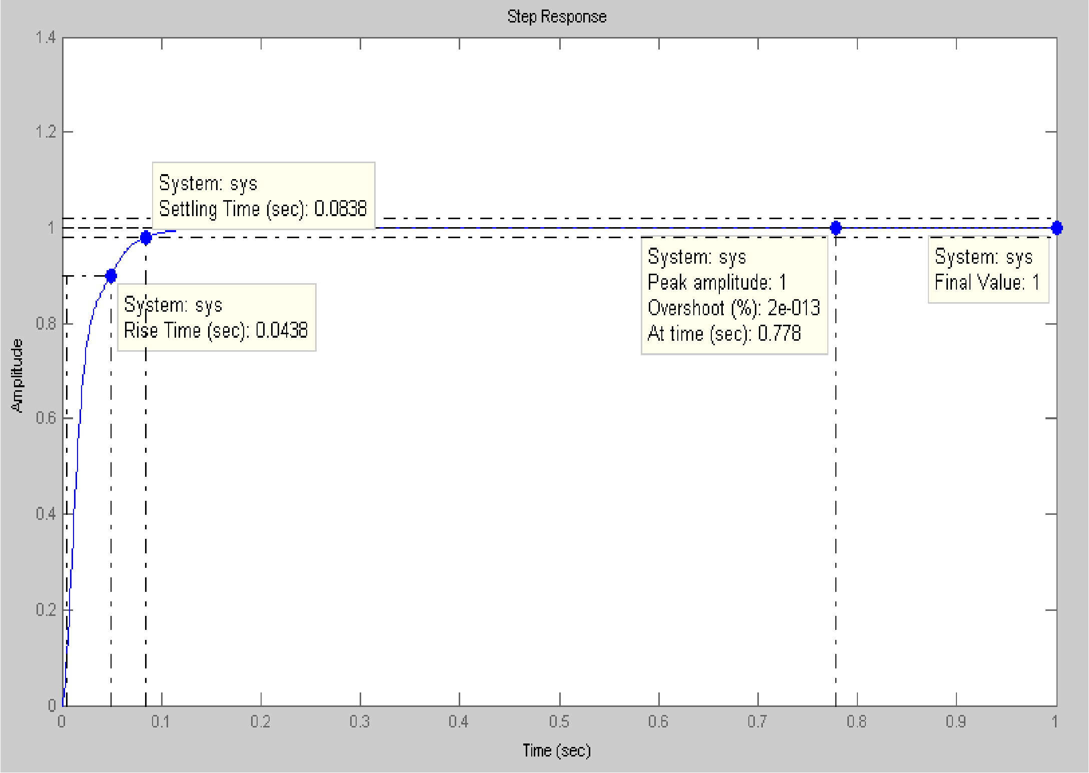


Figure 5.20: Step response of closed loop system.

Using Equations 5.5 to 5.7 and Figure 5.21 bode plot, the DC gain k, natural frequency ω_n and damping ratio ξ of the position control closed loop system are 1, 128 rad/s, and 1.02, respectively. The closed loop system is critically damped, since ξ =1.

The closed loop system has three pole and two zero on the left sides of the pole-zero map (Figure 5.22). Since, the pole and zero are located on the left sides of the pole-zero map, then it means the system is stable.

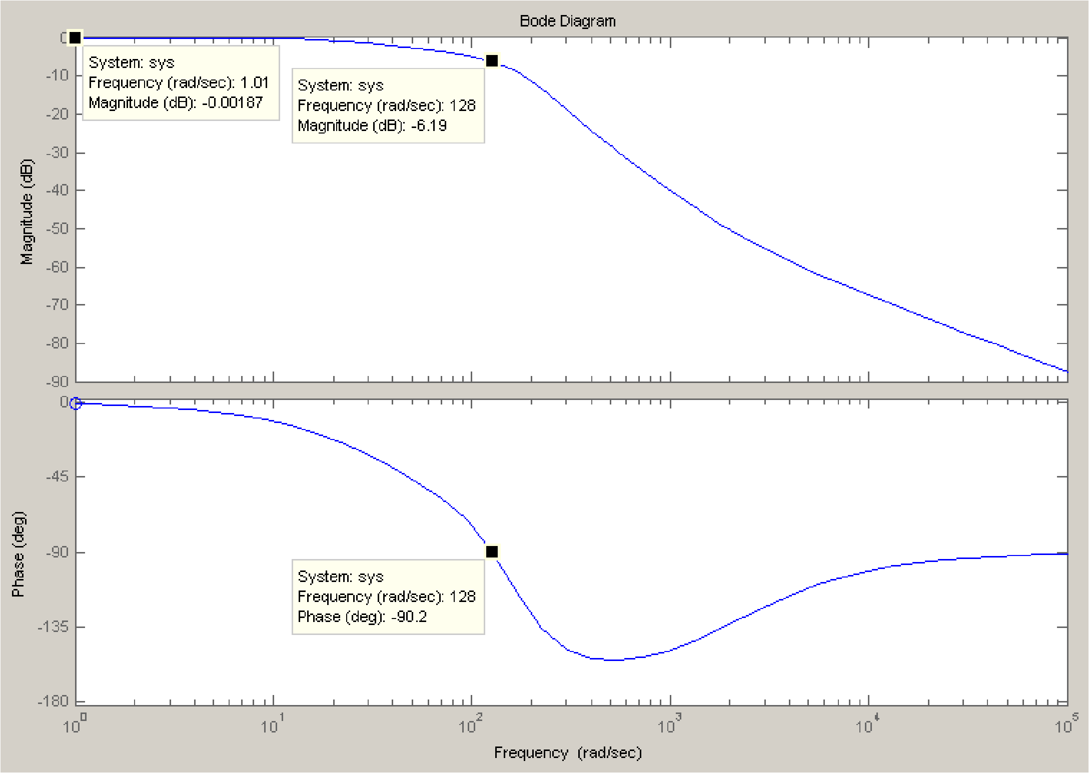


Figure 5.21: Bode plot of closed loop system.

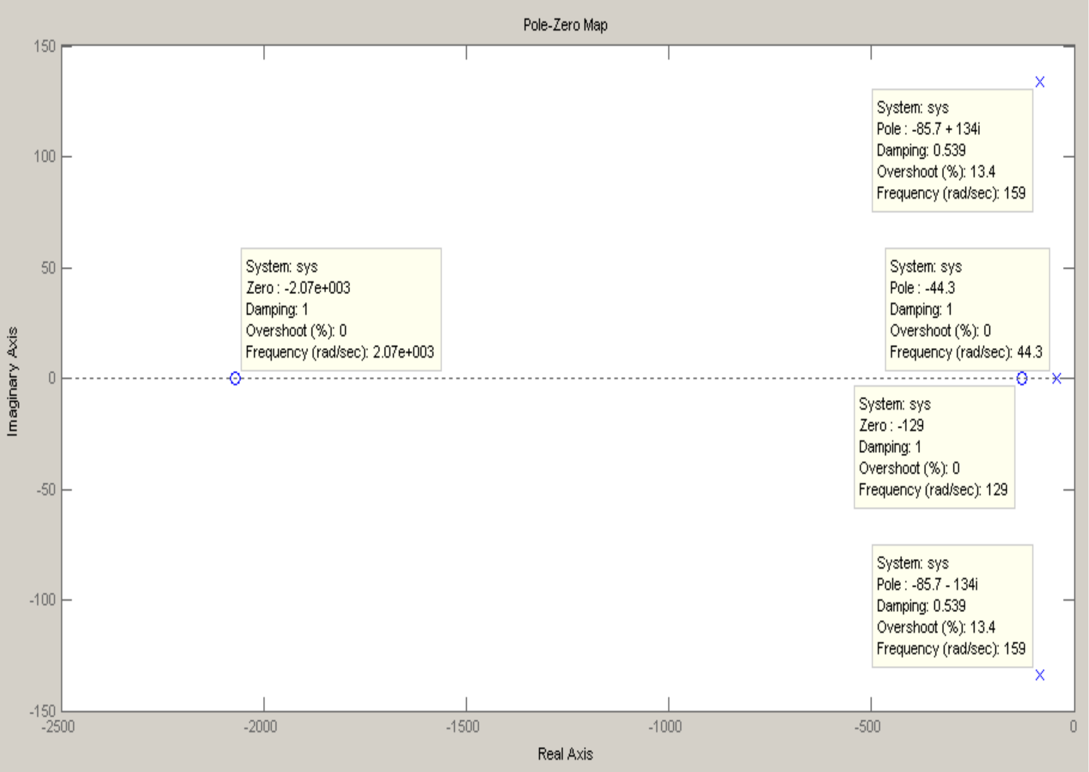


Figure 5.22: Pole-zero plot of plant model.

5.2.3 Deployment

Once a control system is designed, the algorithm can be deployed to control a real physical system. This is implemented in the finalised control system, which is deployed to the control algorithm on real hardware that controls the angular position of the DC actuator. As a result, it replaces the plant model with real hardware shown in Figure 5.23 and the full programming is shown in Appendix B.

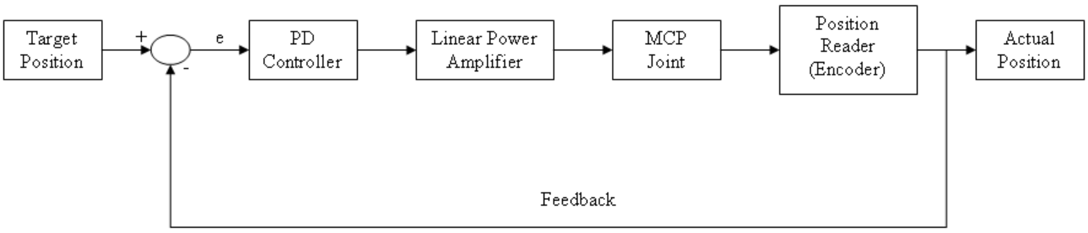


Figure 5.23: Block diagram of desired closed-loop system with real hardware.

When the target angular position is set to 1.57 rad, it will then compare to the actual angular position and send an error e to the PD controller (Figure 5.23). The PD controller will automatically adjust the PD output value based on K_P and K_D (Equation 5.8). After that, the PD output will send to the DAQ_mx to generate a signal to the DC actuator (which is connected to the MCP joint). When the DC actuator is activated, the encoder will start reading the number of pulses and use the Equation 5.10 to evaluate the actual angular position of the DC actuator. This process is iterative until the actual angular position is equal to the target angular position.

$$\text{Angular\_Position(rad)} = N_{pulse} x \frac{2 * pi}{256 * 64 * 45} \quad \text{(Equation 5.10)}$$

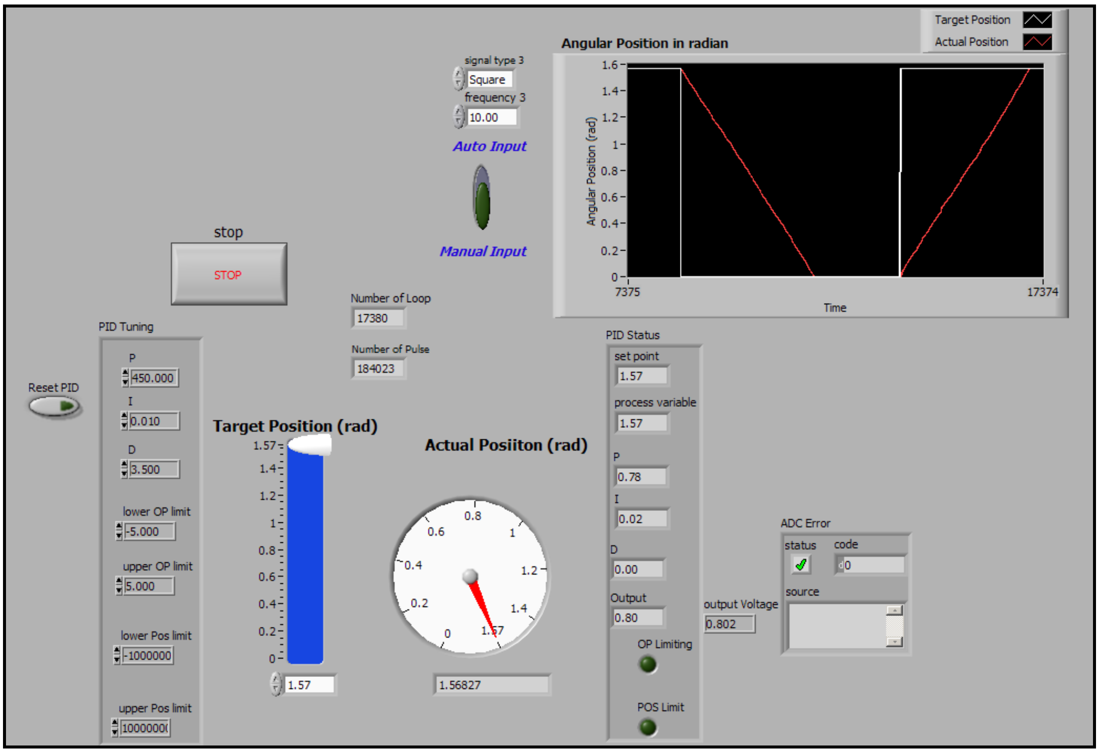
<u>where</u>: N_{pulse} = Number of pulses.

256 = Pulses per revolution

64 = Gear head reduction absolute

45 = External gear ratio (45:1 worm gear)

Figure 5.24 shows the front panel of LabView VI for real time DC actuator position control. When the target angular position was set to 1.57rad, the final actual angular was 1.56827rad. It is meant the steady state error is 0.00173rad (0.11%) compared to the target angular position. The 0.00173rad error is the DC actuator error, due to backlash, etc., in the joint.



signal type 3: Square
frequency 3: 10.00
Number of Loop: 17380
Number of Pulse: 184023

P: 450.000
I: 0.010
D: 3.500
lower OP limit: -5.000
upper OP limit: 5.000
lower Pos limit: -1000000
upper Pos limit: 1000000(

Target Position (rad): 1.57
Actual Posiiton (rad): 1.56827

set point: 1.57
process variable: 1.57
P: 0.78
I: 0.02
D: 0.00
Output: 0.80
output Voltage: 0.802

status: ✓
code: 0
source:

Figure 5.24: Real time DC actuator position control.

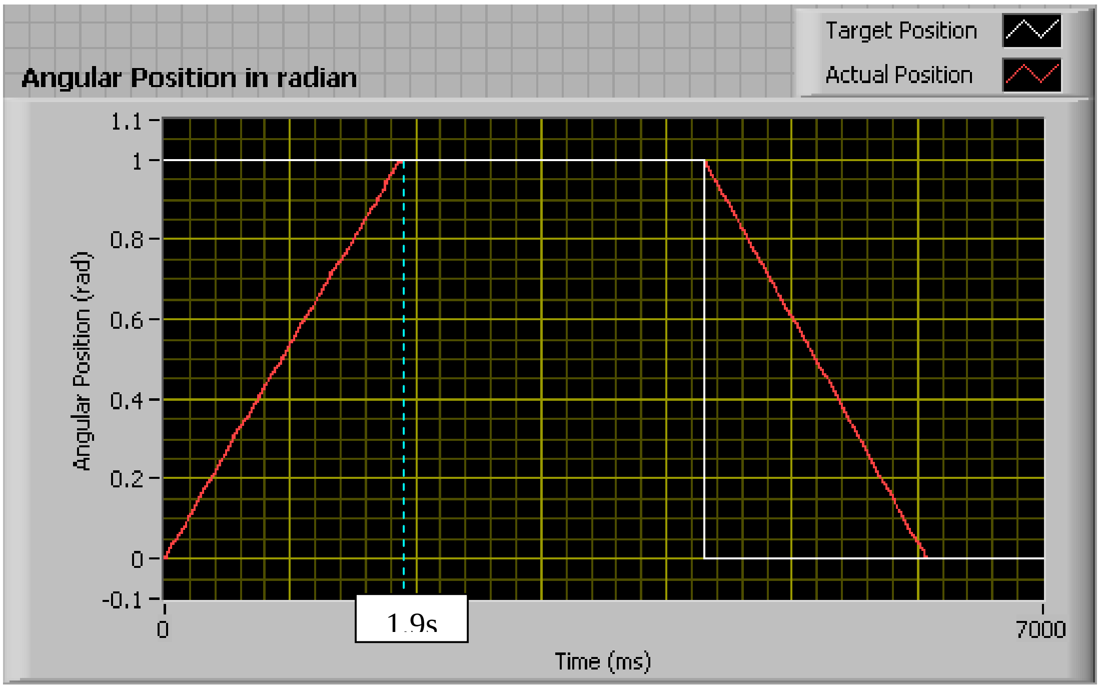


Figure 5.25: Experiment time response.

The experiment time response between the target and actual position is shown in Figure 5.25. The settling time takes around 1.9s and is longer than for the simulation result as shown in Figure 5.20 (0.0838s). The reason is that a voltage saturation was set on the experiment, which is from -5v to +5v, and no saturation was set in the simulation result of Figure 5.20, which means the range of the voltage (output of PD controller) can exceed +5v or go below -5v.

Figure 5.26 shows the simulation time response (1.8s) by using Simulink when the saturation voltage is set between -5v to +5v, and it is close to the experiment time response result (1.9s), as shown in Figure 5.25 with saturation voltage. The Simulink program is shown in Figure 5.27.

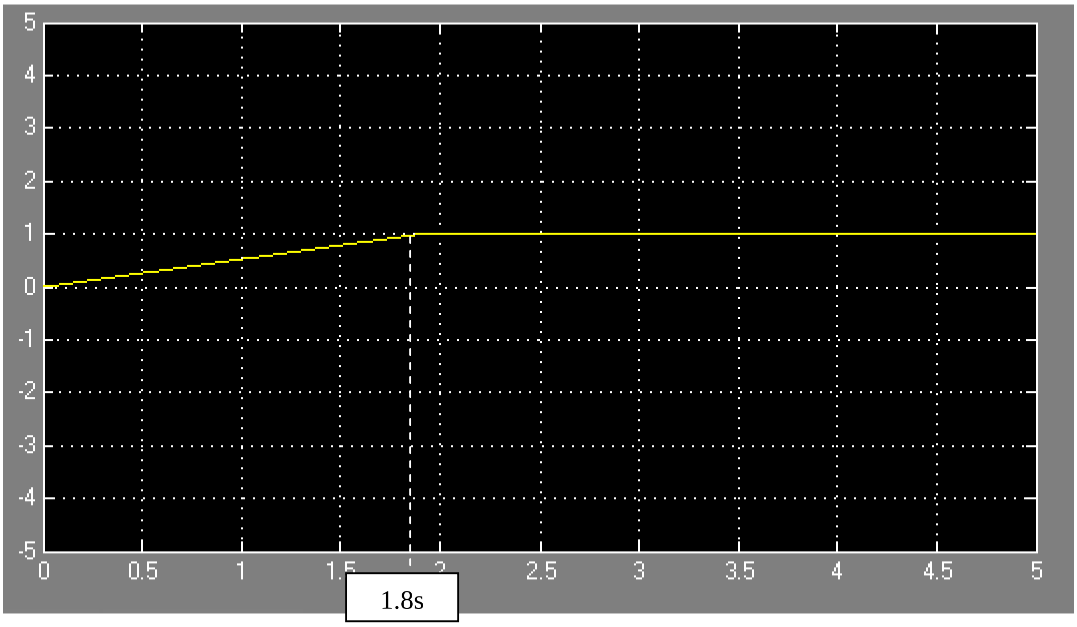


Figure 5.26: Simulation time response with a saturation voltage.

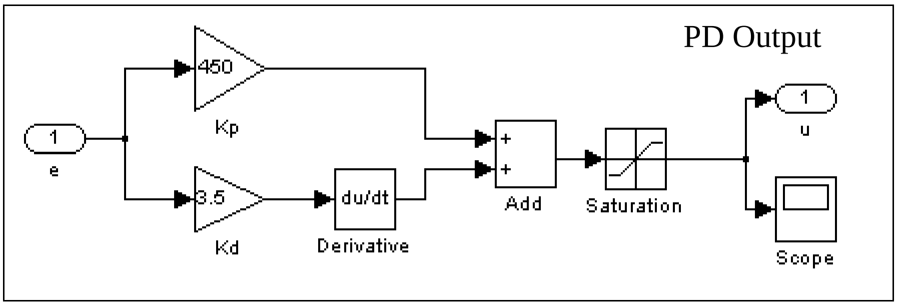


Figure 5.27: PD controller with a saturation.

5.3 Shape Memory Alloy Actuator

The SMA actuator is placed in the palm but different lever with the motor, as shown in Figure 4.9, to control the 110^0 flexion-extension movement of the PIP joint. In order to control the SMA accurate location, a understanding of SMA effects, its applications and the various consideration to be taken into account when using SMA.

Shape memory alloys display two distinct crystal structures or phases, which are martensite exits at lower temperatures and austenite exists at higher temperatures. When a SMA is in martensite form at lower temperatures, the metal can easily be deformed into any shape and when the SMA is heated it goes through transformation from martensite to austenite [62]. However, when the SMA is heated then cooled, the phases of the SMA will not straight away from austenite to martensite without any external load or bias force, it will change to a structure called twinned martensite, which is the configuration shown in Figure 5.28.

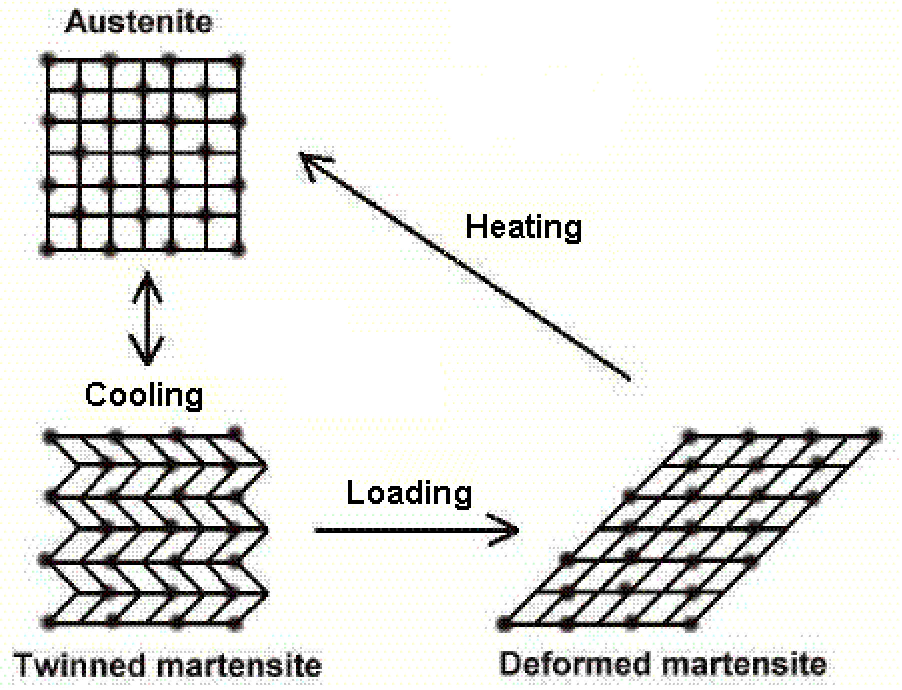


Figure 5.28: SMA crystal structure [68].

In Figure 5.28, in the twinned martensite structure, although the crystal structure has changed, its shape has not. From this point the material can be easily deformed by adding an external load.

Figure 5.29 illustrates a simple experiment using the structure/test-rig of Figure 4.15. The upper end is fixed and a weight or a bias load is attached to the other end. Starting the experiment with the material in its 70^0 Celsius austenite phase, one can expect the wire to show its original length (70mm). By increasing the temperature rises above Af, the SMA wire will fully contract to 67mm and then by steadily decreasing the temperature, the material will enter the twinned martensite state and the tensile load will elongate the wire. The transformation from one state to another does not occur at the same temperature on heating and cooling but follows a hysteresis cycle. This hysteresis gap is described by the transformation temperatures Ms (Martensite start), Mf (Martensite finish), As (Austenite start) and Af (Austenite finish).

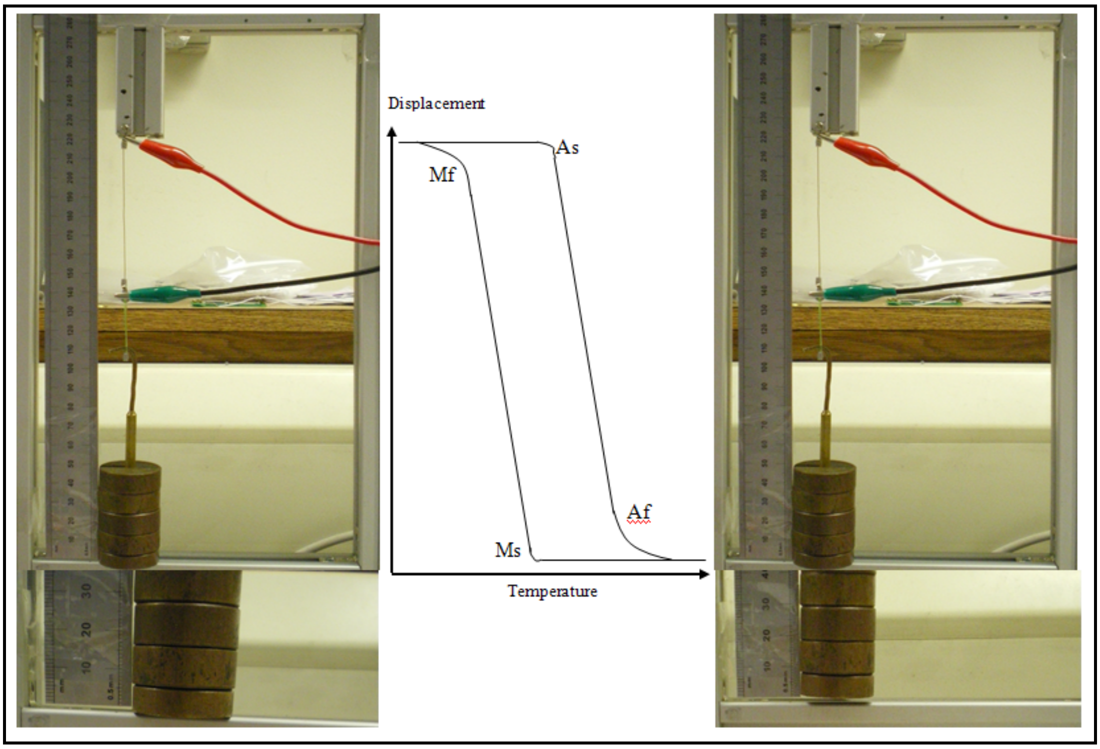


Figure 5.29: Changes of shape under constant load conditions.

5.3.1 Time Response

Since the myoelectric signal will be collected based on the nine wrist joint movements in order to control nine different movements of fingers only open and closed (for full details, see Chapters 6 and 7), therefore the position control of the SMA actuator just to control PIP joint θ_2 has 0 and 110 degrees.

The full contraction times for the 0.012” $(300\mu m)$ SMA single and double layer wire in the actuator are shown in Figures 5.30 and 5.31. These two figures are connected to 350g bias load to bring the SMA wire from Af back to Mf and the given voltage s are 2volt and 3.6volt, respectively. The heating started at 0 sec and the cooling at 3 sec. These experiment were contain three trials.

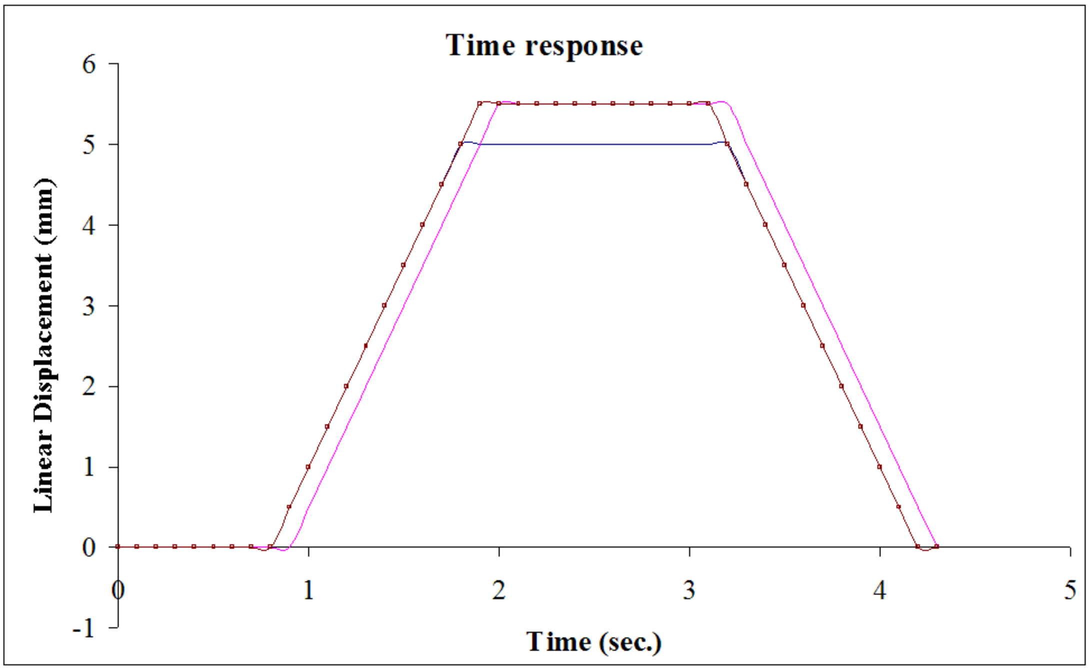


Figure 5.30: Time response for 0.012” SMA single layer wire actuator.

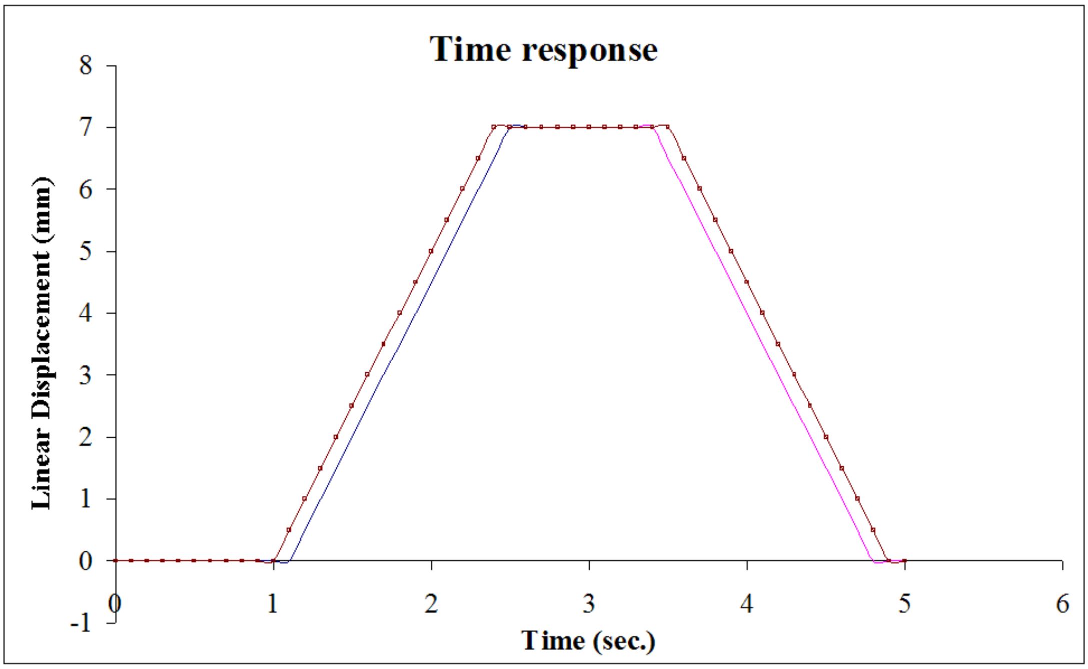


Figure 5.31: Time response for 0.012” SMA double layer wire actuator linear displacement.

This experiment is based on the preliminary SMA actuator design of Figures 4.14 and 4.22, due to the SMA wire used just 4.5% contraction rate, therefore the maximum linear displacement did not reach 13mm (that is 110 degrees of PIP joint). The cooling time from Af to Mf for the single layer SMA wire is faster than the double layer, averaging1.1sec and 1.4sec, respectively. The full contraction (As to Af) time is averaging 1.9sec and 2.1sec for the single and double layer SMA wire. It means the full contraction time takes longer than the fully cooled time. The time for start austenite (As) is take longer in double layers than single layer SMA wire, i.e. averaging 1.1 sec and 0.9 sec, respectively. The reason is because the length of double layers SMA wire is longer than single layer SMA wire.

Apply the relationship in Figure 5.31 to the Equation 4.6 and Figure 4.10 and the time response of the PIP joint angular motion is shown in Figures 5.32.

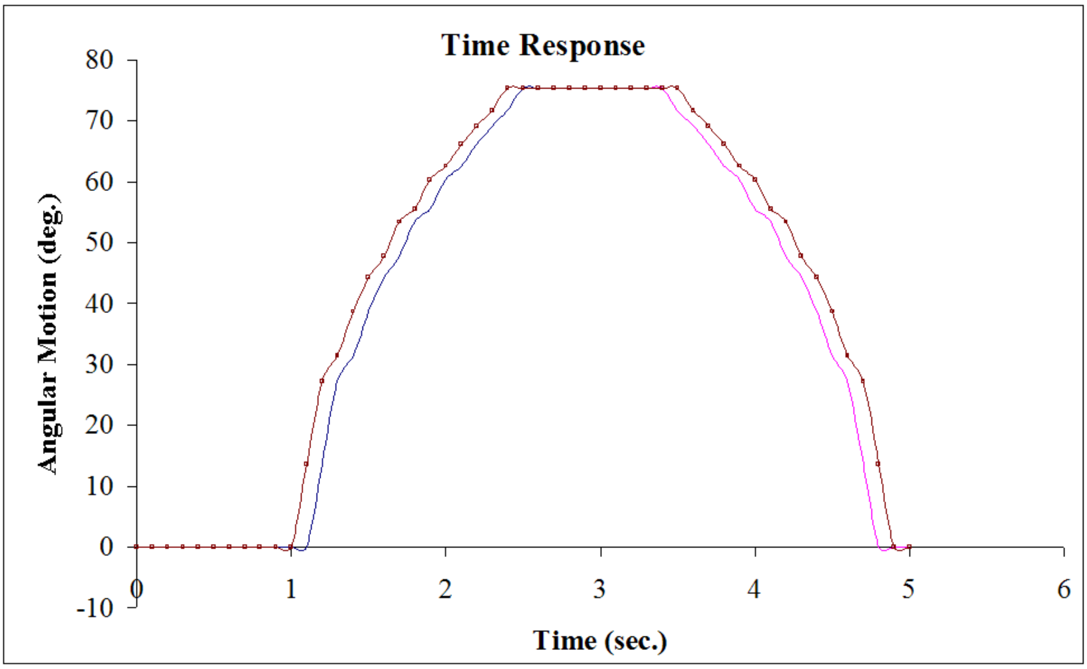


Figure 5.32: Time response for 0.012” SMA double layer wire actuator of PIP joint angular motion.

An assumption was made in Figure 5.32 that the mass of the finger (especially MP and DP) plus the spring S2 has 350g (3.43N) when the PIP joint turns to around 75^0 of PIP joint.

5.3.2 Control Design

Since the SMA transformation occurs as a continuous change over a temperature range, as shown in Figure 4.21, the latter is controlled by the input current, and the input current is controlled by the input voltage due to the resistance of the SMA being based on its length. By controlling the input voltage of the SMA, to control the rate of contraction/deformation in a continuous fashion.

However, the design of a control system for SMA actuators can be a very complex task if the device requires precise control. Many factors contribute to this problem, such as the contraction/deformation rate not precisely at 5%, spring response, different weight of object grasped, etc..

Due to no position sensor being connected on the PIP joint θ_2, the position control of the SMA actuator can not control similarly to the DC actuator in section 5.2. Therefore, a simple open loop control, as shown in Figure 5.33, was used.

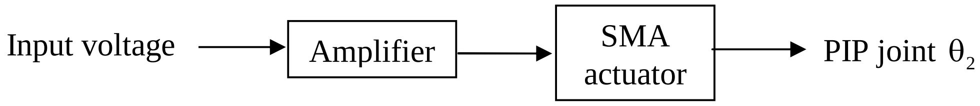


Figure 5.33: Open loop control system of SMA actuator.

Ideally, the open loop control associated with the inverse kinematics formula evaluated in Chapter 3 is good enough to control the middle finger. For example, from the inverse kinematics, we know that the MCP joint θ_1 is 90^0 (1.57rad) and the PIP joint θ_2 should be 65^0 (1.13rad) for position P_x = -54.8 and P_y = 64.75mm and θ_1 is 45^0 (0.785rad), the θ_2 should be 10^0 (0.1745rad) for position P_x = 73.23mm and P_y = 92.13mm. After the value of θ_2 is achieved, it will substitute this value to the Equation 4.6 to evaluate the value of linear displacement of the SMA actuator which should be performed. Then after that, based on Figure 4.21, to evaluate what input voltage needs to be supplied to the SMA actuator.

5.4 Summary

This chapter discussed the control system of the actuation mechanism of the prosthetic hand. Two main sections were covered in this chapter:

1. The control system of the DC actuator

   Three processes have been used to design the control system:

i) System Identification: to identify a mathematical representation of the plant based on a set of real-world stimulus and response data samples.

ii) Control design: A PD control algorithms was used for the control, due to its being the basis for many advanced control algorithms and strategies. Manually tuned technique was applied.

iii) Deployment: Once a control system is designed, the algorithm can be deployed to control a real physical system. This is implemented in the finalised control system, which is deployed to the control algorithm on real hardware that controls the angular position of the finger. As a result, it replaces the plant model with real hardware.

The experiment time response between the target and actual position is shown in Figure 5.25. The setting time takes around 1.85s.

2. The control system of the SMA actuator.

a. In order to control the SMA actuator, an understanding of SMA effects, its applications and the various considerations had to be taken into account.

b. The time response of a single and double layer SMA wire actuator was presented in section 5.3.1. The cooling time from Af to Mf and full contraction time from As to Af of the single layer SMA wire is faster than the double layer due to double layer SMA wire's having longer SMA wire. Besides, from the time response, the full contraction time takes longer than the full cooling time when a large bias load is applied.

c. Without any position sensor connected on the PIP joint, the position control of the SMA actuator cannot be controlled similar to the DC actuator. Therefore, a simple open loop control was used in this case.

CHAPTER 6
Surface Electromyography and its Signal Processing Techniques

6.1 Introduction

The concept of harnessing the remaining part of a limb to operate a terminal device remained central to all development in upper limb prosthetic until the practical introduction of myoelectrically controlled external power, beginning in 1958 [1]. Myoelectric control is the control of a prosthesis through the use of 'muscle electricity'. Muscle electricity is a by-product of muscle action and is picked up with electrodes on the surface of the body. The surface method of detection of muscle activity is appropriately illustrated in the standard electromyography (EMG) [1]. Many efforts have been to implement effective control algorithms based on the processing of EMG signals

The scope of this chapter is as follows. In Section 6.2, the definition of the EMG signals and their acquisition and processing techniques are presented. The specification of the myoelectrodes used in this research work are listed and elaborated in the Section 6.3. In Section 6.4, the author defines the methodology of EMG acquisition of the healthy human forearm based on rigorous literature review. Section 6.5 illustrates the sEMG signal features' extraction by the continuous wavelet transform, following their features' selection criteria. An example of sEMG acquisition and processing is illustrated in Section 6.6.

6.2 Electromyography

Electromyography (EMG) is an analytical tool used to record electrical activity of muscle for clinical or biomedical purposes [20]. There are two kinds of EMG in common use: surface EMG (non-invasive) and needle or intramuscular EMG (invasive) [20, 25]. Surface EMG (sEMG) is capable of providing information about neuromuscular activity in a non-invasive manner through electrodes placed on the skin surface and has thus become an important and effective control input for powered prostheses [25, 68]. Needle or intramuscular EMG on the other hand measures electrical activity by using electrodes inserted through the skin directly into the muscle. This application can cause a considerable amount of discomfort to the patient and thus it is not practicable in powered prostheses [25].

The main prosthesis control functions of interest are flexion and extension of the forearm to control the hand opening and grasping actions. In order to differentiate these control functions, investigators have developed various EMG features which include EMG signal amplitude, EMG frequency characteristic, etc. [50]. In any case, the myoelectric signal without any processing just permits the control of no more than one or two active DOFs, such as one for the gripper and one for the wrist. Hence, in order to be able to control more than two active DOFs, sEMG signal processing is needed. The raw signal contains valuable information in a particular form, which is not readily usable. This information however, can be made useful only if it is quantified by signal processing methods to achieve the accurate and relevant structures in the data, which could be utilised for further tasks, such as pattern classification.

Figure 6.1 shows the usual processing of sEMG signalS, which can be divided into three stages:

1) Data Acquisition

   This includes amplification, signal conditioning and analogue to digital conversion. More details will be discussed in Section 6.2.

2) Features Extraction

   Signal Processing stage to extract desired features from the sEMG signal. Further details will be covered in Section 6.3.

3) Feature Selection

   Selecting and retaining information that is important for the later application such as classification of signals using ANN. More details will be discussed in Section 6.3.

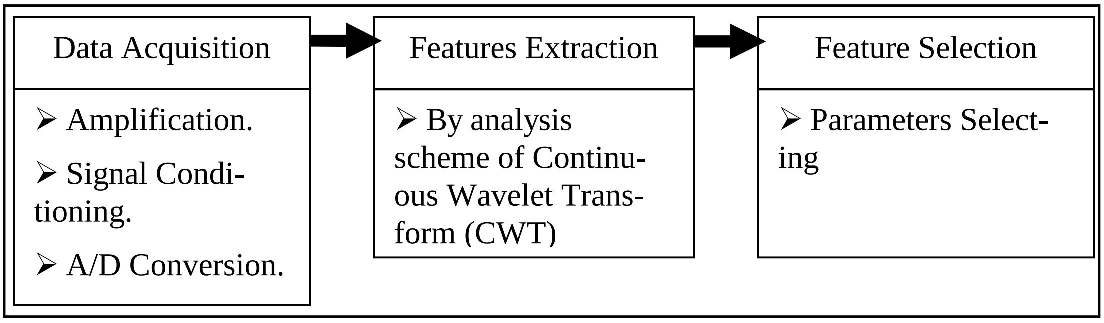


Figure 6.1: Stages of sEMG signal processing.

6.3 Myoelectrodes Sensor

Surface electrodes are currently the only practical way to receive myoelectric signals for prosthesis control because the electrodes are used for a long time every day and hence must be benign to skin and tissue.

The muscle electricity is monitored using the surface pre-amplifier type No. SX 230 (Biometric Ltd) with integral surface electrodes placed on the skin to measure potentials originating from muscle electrical activity. Its specifications are listed on the Table 6.1.

Table 6.1: Specifications of myoelectrodes.

Supply Voltage	+4.5Vdc to +5.0Vdc single sided
Gain Factor	1000
Current	10mA
Accuracy	+/- 2% full scale
Low Pass Filter	450Hz
High Pass Filter	20Hz
Noise	< μ5 V
Electrodes	Integral dry reusable material stainless steel

The high-pass filter is used to remove DC offsets due to membrane potentials and to minimise low frequency interference caused by the movement of the pre-amplifier on the skin surface and the low-pass filter to remove the unwanted frequencies above 450Hz. It is very importance to remove these high frequencies when interfacing to a computer sampling the signal since they would be converted to a lower frequency and mixed with the original signal.

Only two myoelectrodes were used in order to create a user-friendly system. These are placed on the skin over the muscles of the *extensor carpi radialis longus* and *flexor carpi ulnaris,* as shown in Figures 6.2 and 6.4. These two muscles are the thickest in the human hand.

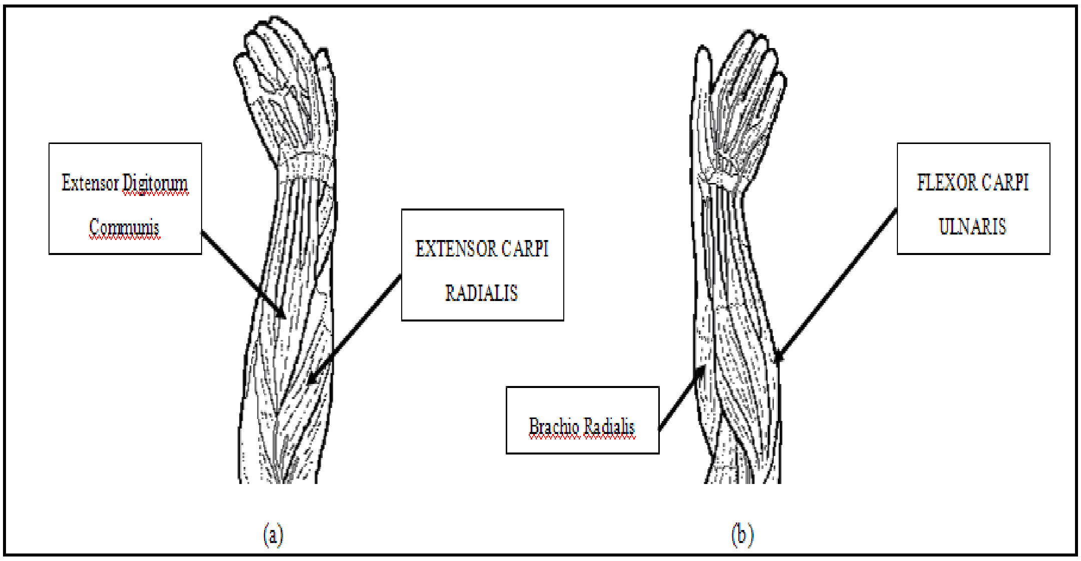


Figure 6.2: (a) Posterior, and (b) Anterior of left forearm muscles.

6.4 Data Acquisition

This section describes the equipment used, and a methodology that covers movement or exercise procedures to obtain forearm muscle maximum contraction.

6.4.1 Equipment and Programming Setups

In order to achieve correct or precise signal, free from any excessive noise or aliasing effect, equipment and programming setups are of crucial importance.

The raw sEMG signals in time domain were collected using NI DAQ_mx card (NI PCI-MIO-16E-4) and a program developed in LabView environment (see Appendix C). The sEMG signals obtained and monitored on a computer screen were filtered by a 20Hz - 450Hz bandpass filter and amplified by a factor of 1000, as these two functions were integrated in the myo-electrode sensors. In all experiments the sampling rate was set to 1024Hz and 1024 number were recorded. According to the Nyquist Theorem, to avoid signal aliasing, the sampling rate must be set at least two times the bandwidth, hence, considering sEMG signal characteristics, 1024Hz sampling rate was high enough to avoid signal aliasing.

6.4.2 Methodology for Collecting sEMG Signal

In order to achieve correct pattern recognition on the further use (Chapter 7), the methodology for collecting sEMG signal is very crucial since the ways for collecting them correspond to multiple fingers movement.

Only one subject was enrolled during signal collection in this study because different subjects would have different muscle strength, shapes and power. These variations could potentially cause problems in pattern classification when using the neural network. The pattern classification result of different subjects is shown in Appendix F.

One subject (the author) with healthy left forearm participated in this study. Signals were collected when the wrist moved in the direction of nine different wrist joint movements shown in Figures 6.3 to 6.12, without load and with relaxed grasp. These nine wrist joint movements correspond to nine different finger movement patterns.

The direction of the centre point in Figure 6.3 is the left forearm‘s front midpoint of the fist and the signs ‘+45^0’ and ‘-45^0‘represent the degrees of forearm turn in pronation and supination rotation, respectively. Fully up and fully down represent the fist moved vertically up and down when the forearm was at the central point of supination-pronation (see Figure 6.3). In the full extension and full flexion movements, when the forearm was in neither supination nor pronation rotation, the fist was fully ‘bend’ out from and towards the body, respectively. The shoulder and forearm had to be perpendicular, shown in Figure 6.4.

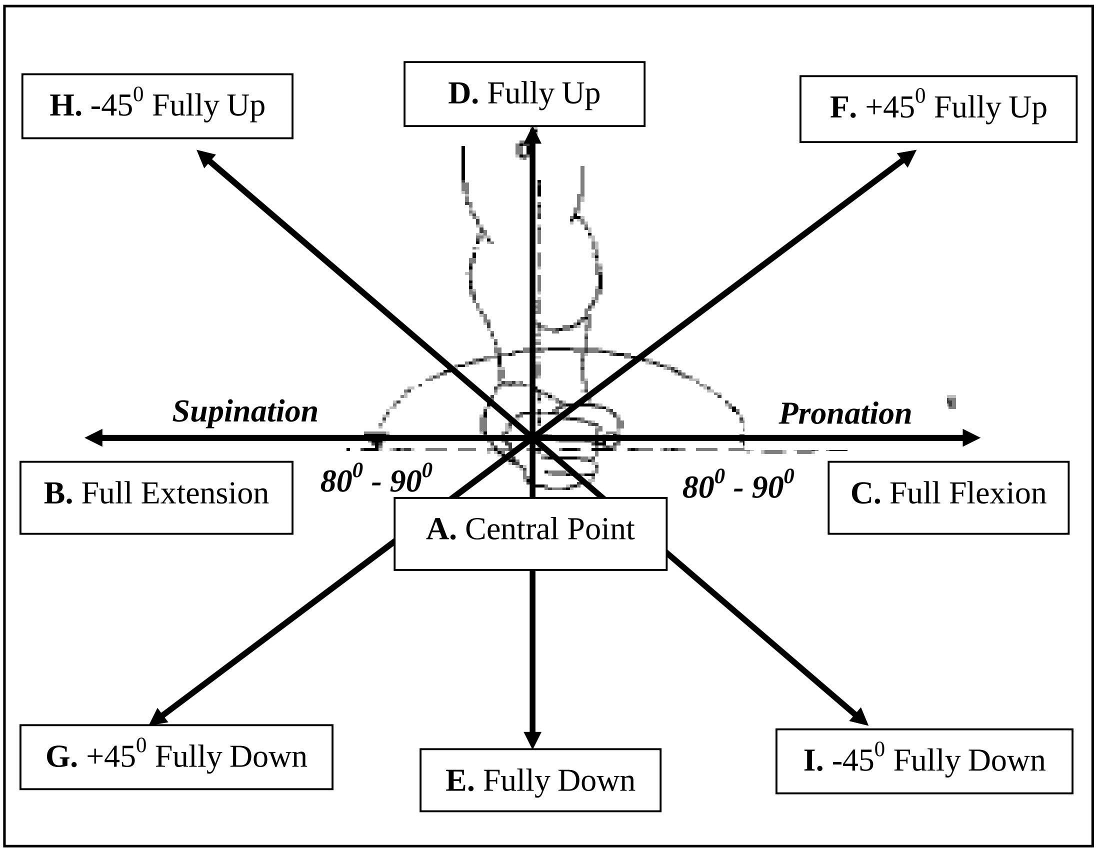


Figure 6.3: Nine different wrist joint movements (Positions A to I).

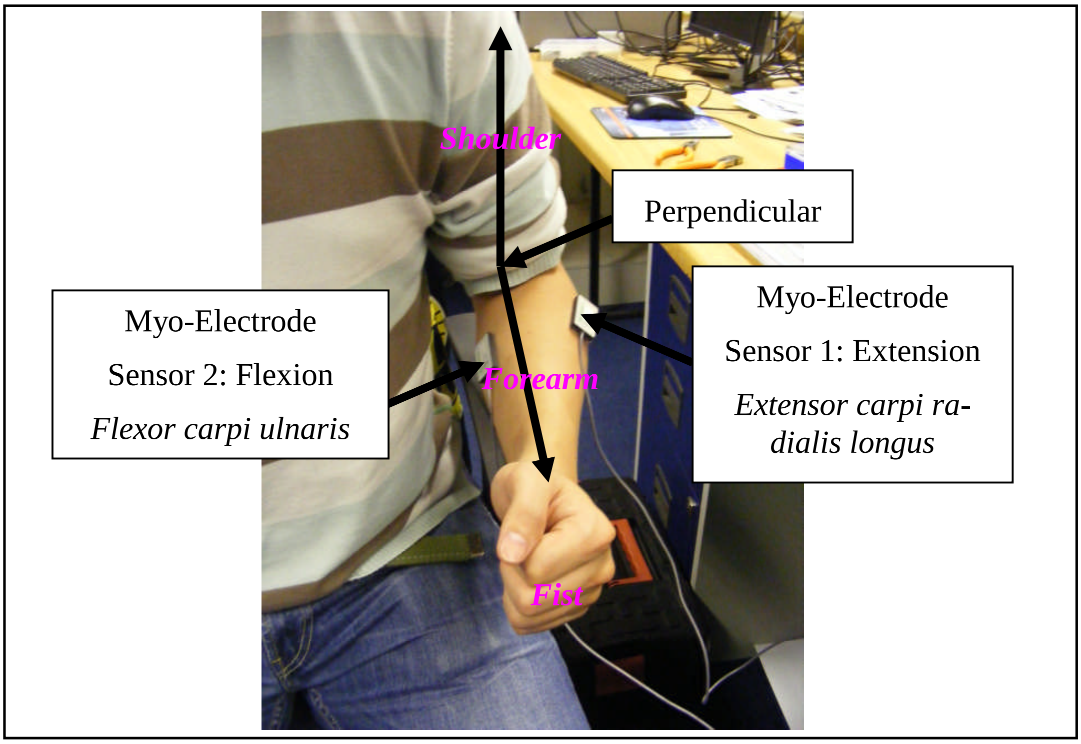


Figure 6.4: Placement of myo-electrode sensors, and subject showing position A – Centre Point.

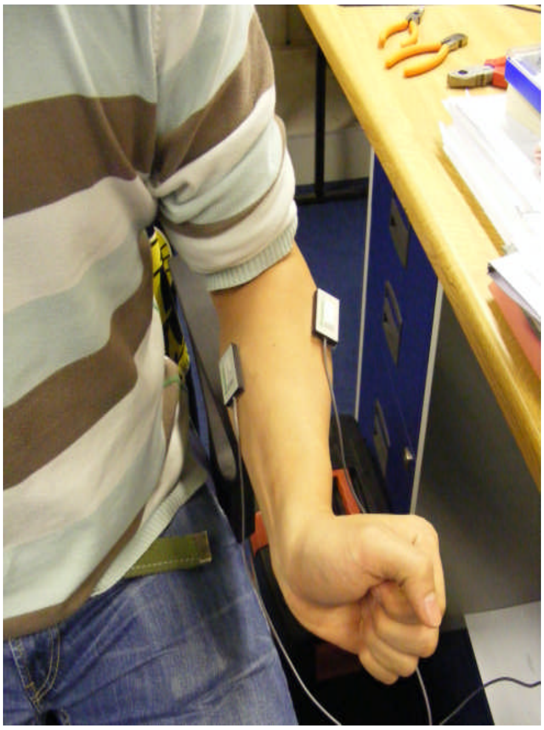


Figure 6.5: Position B, Full extension.

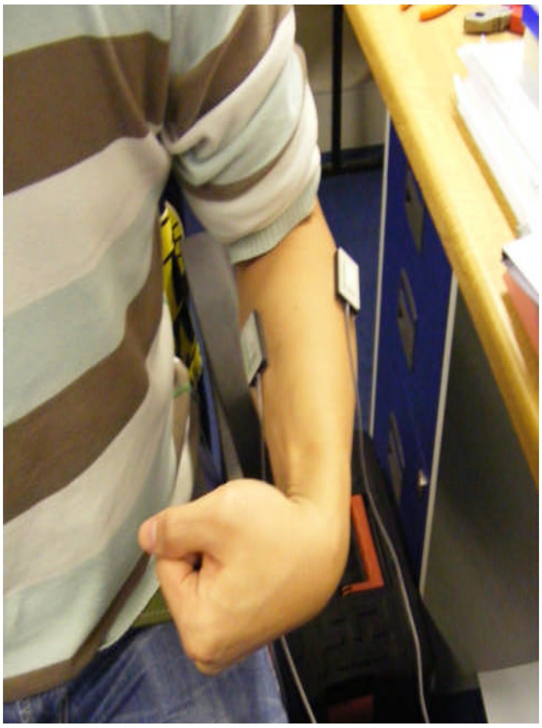


Figure 6.6: Position C, Full flexion.

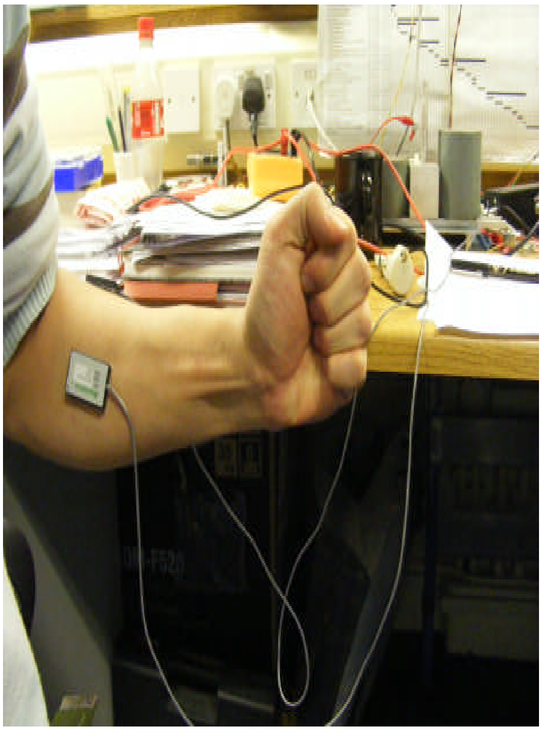


Figure 6.7: Position D, Fully up.

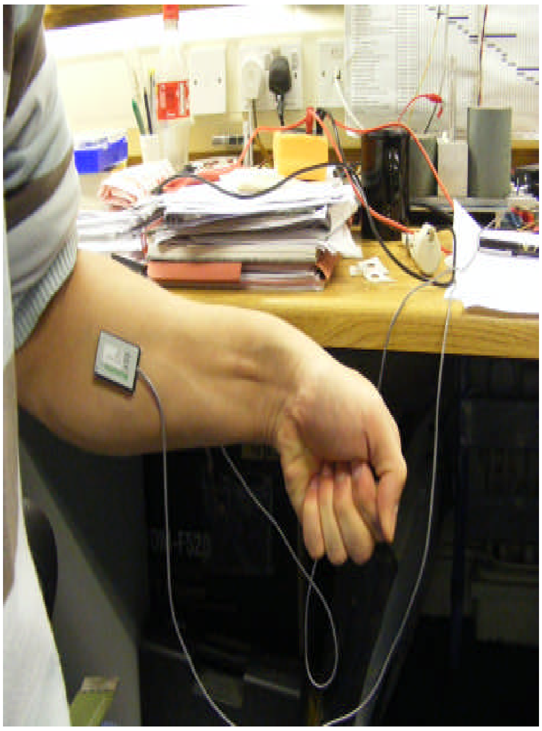


Figure 6.8: Position E, Fully down.

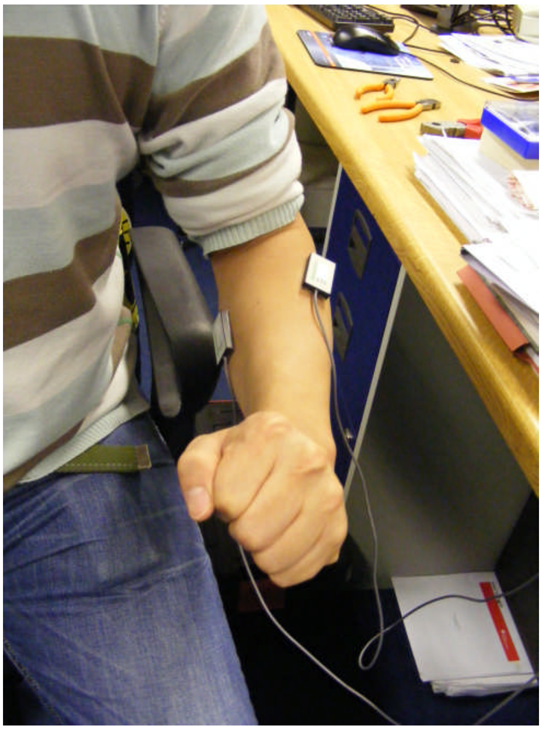


Figure 6.9: Position F, +45^0 fully up.

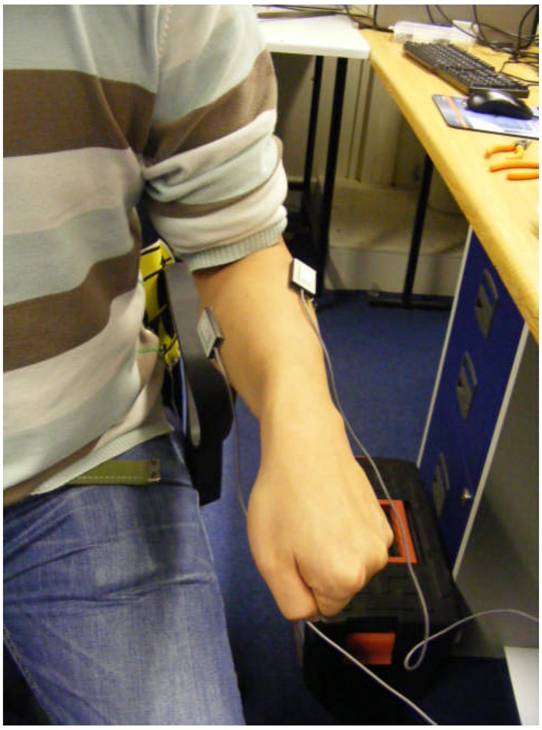


Figure 6.10: Position G, +45^0 fully down.

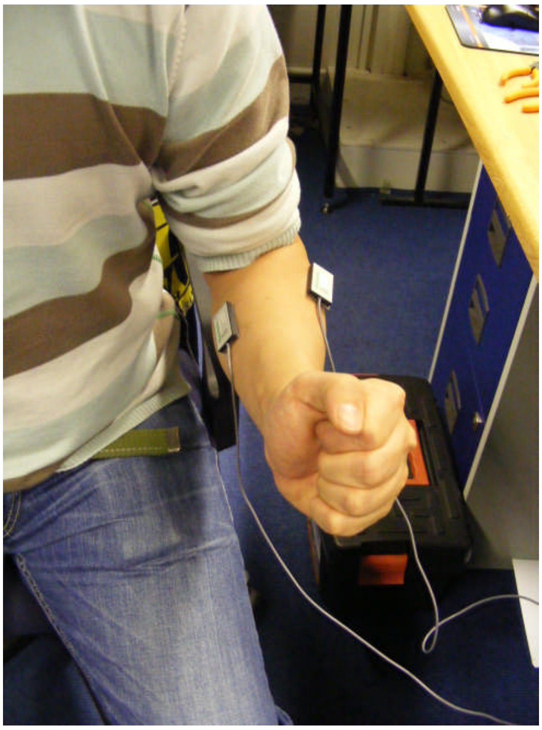


Figure 6.11: Position H, -45^0 fully up.

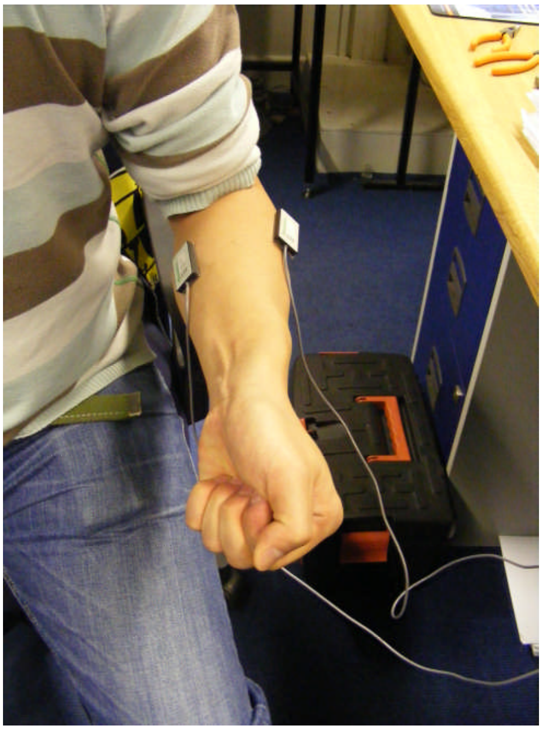


Figure 6.12: Position I, -45^0 fully down.

Figures 6.4 to 6.12 show the position of the subject’s hand corresponding to the nine patterns of wrist joint movements, Centre Point, Full Extension, Full Flexion, Fully Up, Fully Down, +45^0 Right Up and Down, -45^0 Left Up and Down. These nine wrist joint movements will be used to control the nine different finger movement patterns. The corresponding of these nine different types of wrist joint movement and the multiple finger movement patterns is separated into four groups is show in Tables 6.2 to 6.5. The purpose of this is to try to investigate how many pattern can be recognised based on use of only two myoelectrodes.

Table 6.2: Relationship between three movements of wrist joint and prosthetic fingers.

Patterns	Wrist Joint Movements	Prosthetic Hand
A	Wrist Joint Centre Point	No Fingers Movement
B	Wrist Joint Fully Extended	Four Fingers extended (Thumb is not included)
C	Wrist Joint Fully Flexed	Four Fingers flexed (Thumb is not included)

Table 6.3: Relationship between five movements of wrist joint and prosthetic fingers.

Patterns	Wrist Joint Movements	Prosthetic Hand
A	Wrist Joint Centre Point	Original Position (All Fingers extended)
B	Wrist Joint Fully Extended	Index Finger flexed
C	Wrist Joint Fully Flexed	Index and Middle Fingers flexed
D	Wrist Joint Fully Up	Index, Middle and Ring Fingers flexed
E	Wrist Joint Fully Down	Index, Middle, Ring and Little Fingers flexed

Table 6.4: Relationship between seven movements of wrist joint and prosthetic fingers. (Consider ring and little fingers as one digit).

Patterns	Wrist Joint Movements	Prosthetic Hand
A	Wrist Joint Centre Point	Original Position (All Fingers extended)
B	Wrist Joint Fully Extended	Index Finger flexed
C	Wrist Joint Fully Flexed	Index and Middle Fingers flexed
D	Wrist Joint Fully Up	Index, Middle, Ring and Little Fingers flexed
E	Wrist Joint Fully Down	Middle Finger flexed
F	Wrist Joint +45^0 Right Up	Middle, Ring and Little Fingers flexed
G	Wrist Joint +45^0 Right Down	Ring and Little Fingers flexed

Table 6.5: Relationship between nine movements of wrist joint and prosthetic fingers.

Patterns	Wrist Joint Movements	Prosthetic Hand
A	Wrist Joint Centre Point	Original Position (All Fingers extended)
B	Wrist Joint Fully Extended	Index Finger flexed
C	Wrist Joint Fully Flexed	Index and Middle Fingers flexed
D	Wrist Joint Fully Up	Index, Middle and Ring Fingers flexed
E	Wrist Joint Fully Down	Index, Middle, Ring and Little Fingers flexed
F	Wrist Joint +45^0 Right Up	Middle Finger flexed
G	Wrist Joint +45^0 Right Down	Middle and Ring Fingers flexed
H	Wrist Joint -45^0 Left Up	Middle, Ring and Little Fingers flexed
I	Wrist Joint -45^0 Left Down	Ring and Little Fingers flexed

6.4.3 Initial sEMG Data Collection

Eighteen sets of data were collected from a subject (author) in the initial sEMG data collection stage. Each wrist joint movement was performed ten trials for every set and the time for data collection was set to 3 sec in each trial. After the 3 sec data recording, the subject was asked to relax around 3 to 4 sec in order to avoid muscle fatigue (see Figure 6.13). Consequently, for each set of data, a maximum of 180 blocks of data were recorded for nine types of wrist joint movement (nine wrist joint movements x 10 trials x 2 channel of sEMG sensors). Hence, the total number of samples in four different groups is shown in Table 6-6.

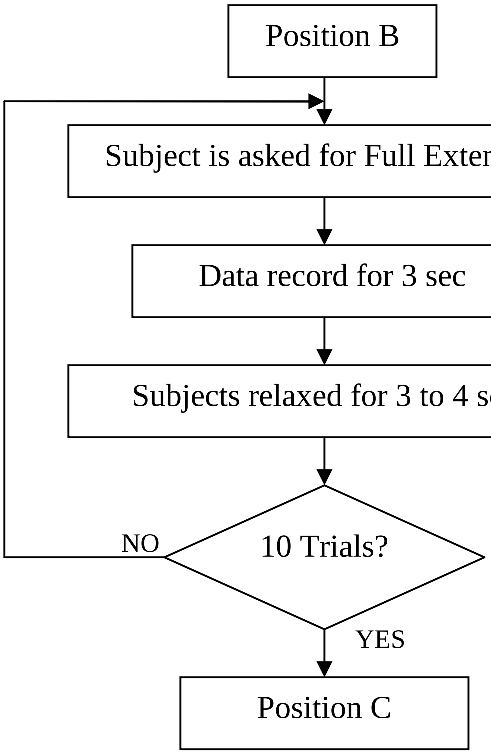

Position B
Subject is asked for Full Extended
Data record for 3 sec
Subjects relaxed for 3 to 4 sec
10 Trials?
NO
YES
Position C

Figure 6.13: Flowchart for data acquisition from wrist joint movement B to wrist joint movement C.

Table 6.6: Total number of samples and data blocks in four different groups of patterns.

Patterns	No. of Wrist movements	No. of Myoelectrodes	No. of Trials	No. of samples	Total No. of data blocks
Three	3	2	10	18	1080
Five	5	2	10	18	1800
Seven	7	2	10	18	2520
Nine	9	2	10	18	3240

6.5 Feature Extraction and Selection

The second and third steps of the sEMG signal processing is feature extraction and features selection.

During feature extraction, the measured sEMG signal was processed in order to emphasize the relevant structures in the data, while rejecting noise and irrelevant data, producing the so-called ‘original feature’ set. Sometimes a reduction of the dimensionality is needed to simplify the task of the classifier.

The role of the feature selection is to retain information important for application such as signal classification. Their work on feature selection involved methods attempting to determine the best subset of the original feature set, and feature projection to determine the best combination of the original features. Their overall aim for dimensionality reduction is to have classifier a with fewer inputs and fewer adaptive parameters which should lead to a classifier with better generalisation properties.

The signal features were extracted using Continuous Wavelet Transform (CWT). Five different scales indeces of CWT will be used to extract the features. These five different scales indeces are 8, 16, 32, 64, and 128. Low scale number contains compressed wavelet, rapidly changing details and high frequency; High scale number contains stretched wavelet, slowly changing details and low frequency. Figure 6.16 shows the signal differences with respect to different scales.

The features selection was aimed to determine the RMS values of the signals, mean and median frequencies from the average power spectrum for each signal at difference scales of CWT. These features are selected as they are commonly used in most EMG research using the conventional Fourier analysis method [21, 41, 69, 70]. The RMS value is to provide useful measurement of the signal power or amplitude and is obtained by Equation 6.1. The power spectrum of the total signal reveals the component of the individual motor unit properties[21]. The area under the power spectral curve equals the signal power. The mean and median frequencies are two reliable measures of the power spectrum. The mean frequency is the average of all frequencies from the power spectrum and is obtained by Equation 6.2. The median frequency is the frequency with 50% or half of the frequency distribution on each side. By conclusion, the mean and median frequencies are used to assess muscle fatigue. The results of [70] illustrate the significant increase in sEMG amplitude (contraction force) and mean power frequency was higher, which meant that during muscle fatigue, the power spectrum of sEMG shows a shift to lower frequencies. Median frequency is used to quantify this shift.

After values of features are extracted, these will be stored in a table, such as Table 6.7, for Artificial Neural Network (ANN) pattern classification.

$$RMS = \sqrt{\frac{\sum_{i=0}^{n} A_i^2}{n}} \quad \text{(Equation 6.1)}$$

<u>where</u>:

RMS : Root Mean Square,

A_i : Amplitude in *i*th sample,

n : Total number of samples.

$$\overline{\mathrm{F}} = \frac{\sum_{\mathrm{i=0}}^{\mathrm{n}} \mathrm{f_i A_i^2}}{\sum_{\mathrm{i=0}}^{\mathrm{n}} \mathrm{A_i^2}} \quad \text{(Equation 6.2)}$$

<u>where</u>:

$\overline{F}$: Mean Frequency,

f_i : Frequency in *i*th sample,

A_i : Amplitude in *i*th sample,

n : Total number of samples

The flowchart of the MatLab program developed to carry out the required calculations is shown in Figure 6.14. The full program is enclosed in Appendix D. The detailed specification of settings used to process on this stage 6.3 is listed in Table 6.8.

Table 6.7: Feature extraction from raw sEMG signal for sensor 1 or sensor 2.

No	**Features**	**Data Set 1**								
		1st Trial			**… nth Trial …**			**10th Trial**		
		Position A	**….**	**Position I**	**Position A**	**….**	**Position I**	**Position A**	**….**	**Position I**
	Median Frequency at Scale:									
1	8									
2	16									
3	32									
4	64									
5	128									
	Mean Frequency at Scale:									
6	8									
…	**…..**									
10	128									
	RMS at Scale:									
11	8									
…	**…..**									
15	128									

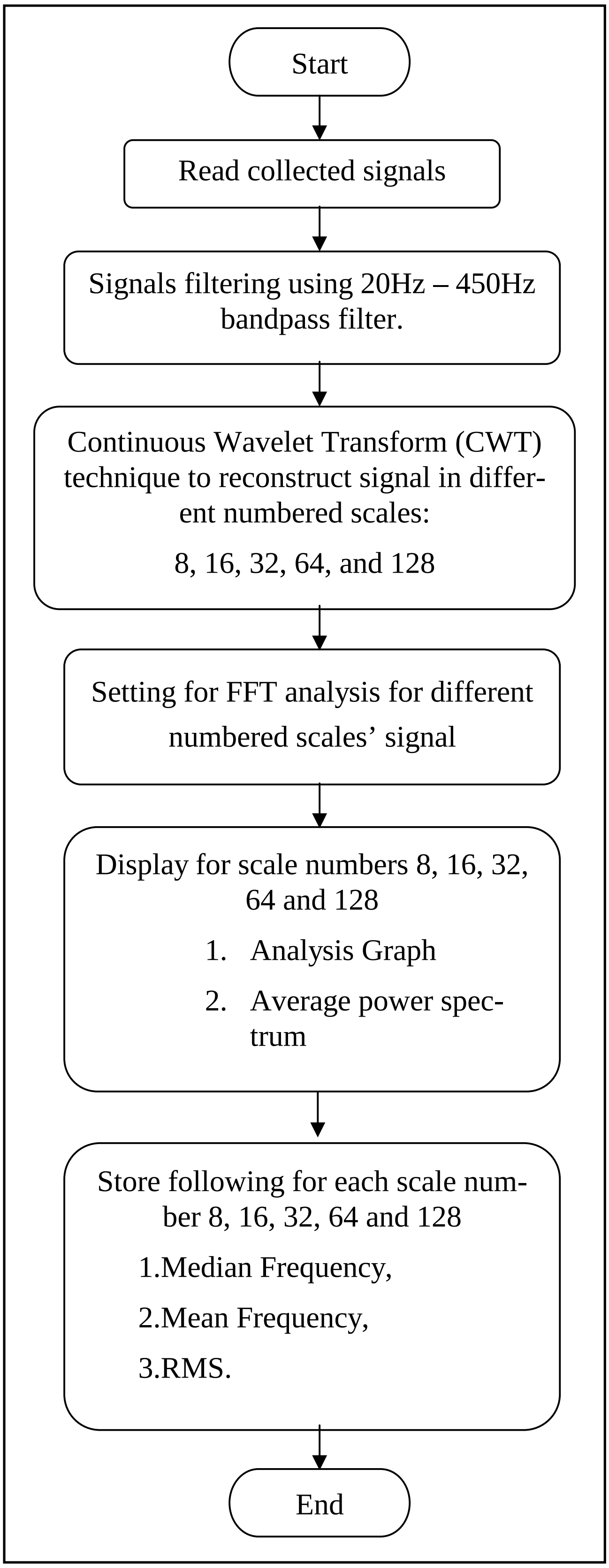

Start
Read collected signals
Signals filtering using 20Hz – 450Hz bandpass filter.
Continuous Wavelet Transform (CWT) technique to reconstruct signal in different numbered scales:
8, 16, 32, 64, and 128
Setting for FFT analysis for different numbered scales' signal
Display for scale numbers 8, 16, 32, 64 and 128
1. Analysis Graph
2. Average power spectrum
Store following for each scale number 8, 16, 32, 64 and 128
1.Median Frequency,
2.Mean Frequency,
3.RMS.
End

Figure 6.14: Flowchart of MatLab Program developed for final analysis, displaying and storing result of sEMG signal using CWT.

Table 6.8: List of specifications for signal processing using CWT.

Sampling Frequency	1024Hz
Filter	5th order Butterworth 20Hz – 450Hz Bandpass filter
Total Time of raw Signal	3 seconds
Total no. of Samples	3072
Features Extraction Technique	Continuous Wavelet Transform (CWT)
Scales	8, 16, 32, 64, 128
Mother Wavelet	Db05
Features Selection	RMS, Mean and Median Frequency in Difference Scales

6.6 Result of One Typical Signal

This section shows the result of one typical data obtained from one of the eighteen sets of data. This includes the presentation of the original filtered signal, the scalogram by the CWT and the signal reconstruction in different scale selection. The result contains determined values of the RMS and the mean and median frequencies for each of the selected scales.

The CWT scalogram shows the scales from 1 to 257, and the scales selected are 8, 16, 32, 64 and 128. For every one of these scales, the values of the RMS, the mean and median frequencies are obtained, corresponding to one of the nine different types of wrist joint movement, i.e., either Central, Fully Extension, Fully Flexion, Fully Vertical Up, Fully Vertical Down, +45^0 Up and Down or -45^0 Up and Down. These nine different types of wrist joint movement samples are obtained from maximum voluntary contraction (MVC) on the left hand by placing two myoelectrodes on the skin over the muscles of *extensor carpi radialis longus* and *flexor carpi ulnaris*.

Table 6.9 shows the values of the RMS, the mean and median frequencies in five different scales for one of the nine types of wrist joint movement. Figures 6.15 and 6.16 show the CWT scalogram of the original filtered signal and the signal reconstruction in different selected scales, respectively.

Table 6.9: Features value of the left forearm *extensor carpi radialis longus* muscles from one data set and trail at fully extension (Pattern B) scheme.

Scales	Median Frequency (Hz)	Mean Frequency (Hz)	RMS (mV)
8	95	90.1049	0.1829
16	57	56.3854	0.1479
32	26	24.1891	0.0929
64	16	15.9622	0.0577
128	11	12.0299	0.0074

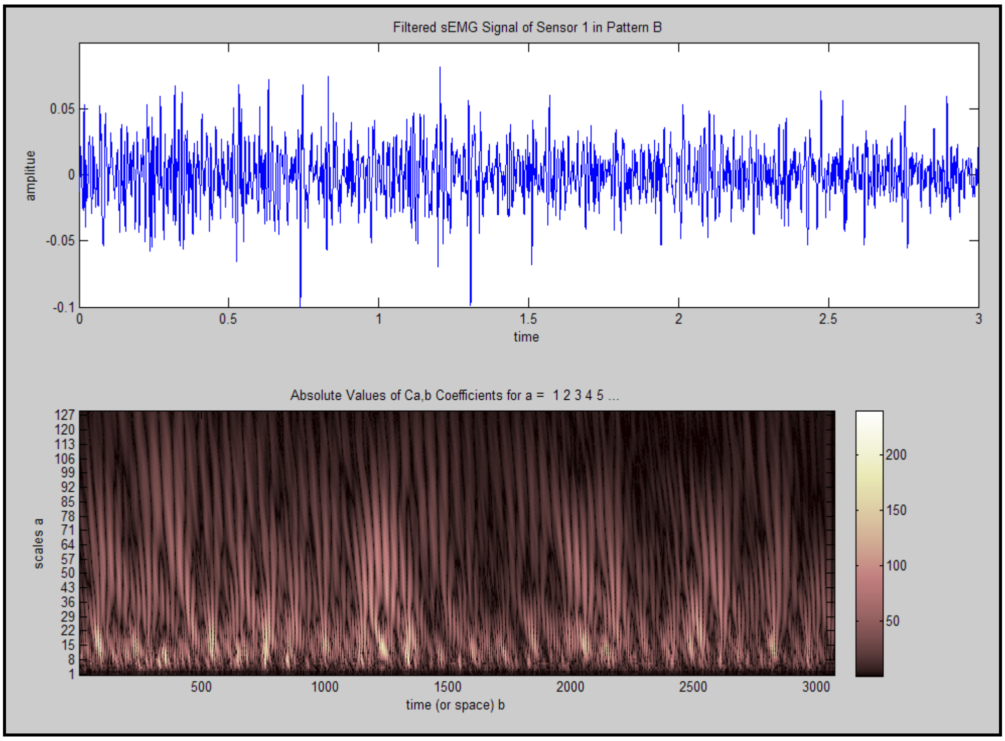


Figure 6.15: Filtered signal and CWT colour scalogram plot from scales 1 to 129 (colour bar on right of scalogram plot is number of CWT coefficients).

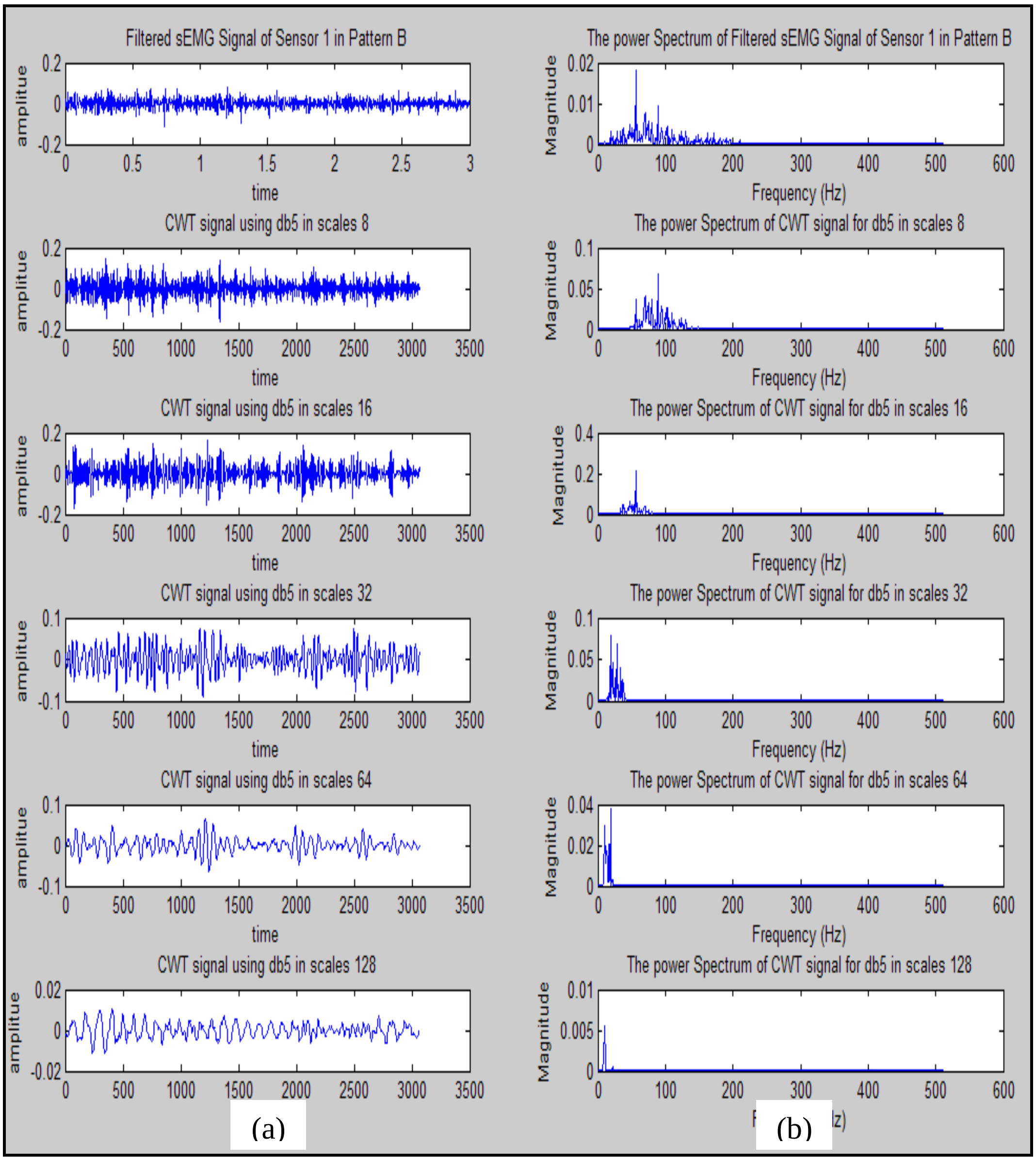


Figure 6.16: (a) Original filtered signal and signal reconstruction in scales of 8, 16, 32, 64, 128, (b) The power spectrum corresponding to each scale.

6.7 Summary

This chapter presented the technology of the Electromyography (EMG) used in this research. A simple understanding of EMG signal and its effect have been described.

The processing of sEMG signals can be divided into a three stage process.

1. The first step is data acquisition, including amplification, analogue to digital conversion and signal conditioning. A factor of 1,000 is used in amplification

and 20Hz – 450Hz band pass filter was used to filter out the noise of the sEMG signal.

   a. Only two myoelectrode sensors were placed on the skin of the left forearm muscles, which are *extensor carpi radialis longus* and *flexor carpi ulnaris*. These two muscles are the thickest muscles of the human forearm.

   b. One subject with healthy left forearm muscle participated in this experiment and eighteen sets of data were collected according to the nine different types of wrist joint movement.

   c. The relationship between the nine different types of wrist joint movement and the multiple finger movement patterns is separated into four groups. The purpose of this task was to investigate the number of patterns that can be recognised based on use of only two myoelectrodes.

2. Secondly, to extract desired features from the sEMG signal, Continuous Wavelet Transform (CWT) technique was used in five different scales, i.e. scales number 8, 16, 32, 64 and 128.

3. Thirdly, a features selection stage is important for Artificial Neural Network (ANN) pattern classification. The features selected are RMS values, mean and median frequencies of power spectrum in scales number 8, 16, 32, 64 and 128.

After the signal pre-processing and features selection stage, a total of 15 columns of features and 180 rows of every set data for nine patterns were used as the input data of Neural Network pattern classification, 15 columns of features and 140 rows of every set data for seven patterns, 15 columns of features and 100 rows of every set data for five patterns, and 15 columns of features and 60 rows of every set data for three patterns.

The result of sEMG signal processing in full extension (Pattern B) for one particular set of data and one particular trial for one myo-electrode sensor was also presented in this chapter.

CHAPTER 7
Pattern Recognition

7.1 Introduction

Study in the past has used Surface Electromyography (sEMG) pattern recognition to control prosthetic devices in amputees [50]. The main prosthesis control functions of interest were flexion and extension of the forearm to control hand opening and grasping actions. Artificial Neural Network (ANN) is the most popular classification tool which is commercially available and it may be defined as structures comprised of densely interconnected adaptive simple processing elements (called artificial neurons or nodes) capable of performing massively parallel computations for data processing and knowledge representation [71]. Figure 7.1 shows the flow of the sEMG signal.

The objective of this chapter is to execute the last stage of the sEMG signal pattern classification by using ANN. Section 7.2 shows the fundamental principles of neural network and gives a picture of the functionality of neural network and of the back-propagation algorithm. Subsequently, Section 7.3 will show the data assembling and array management for the input and output-target data of the neural network in order to train three, five, seven and nine patterns. The methodology for designing the network is shown in the Section 7.4, with parameters' setting of the neural network and the number separation for training, validating and testing, etc. Lastly, Section 7.5 shows the results, such as the correct percentage and the analysis of the pattern recognition.

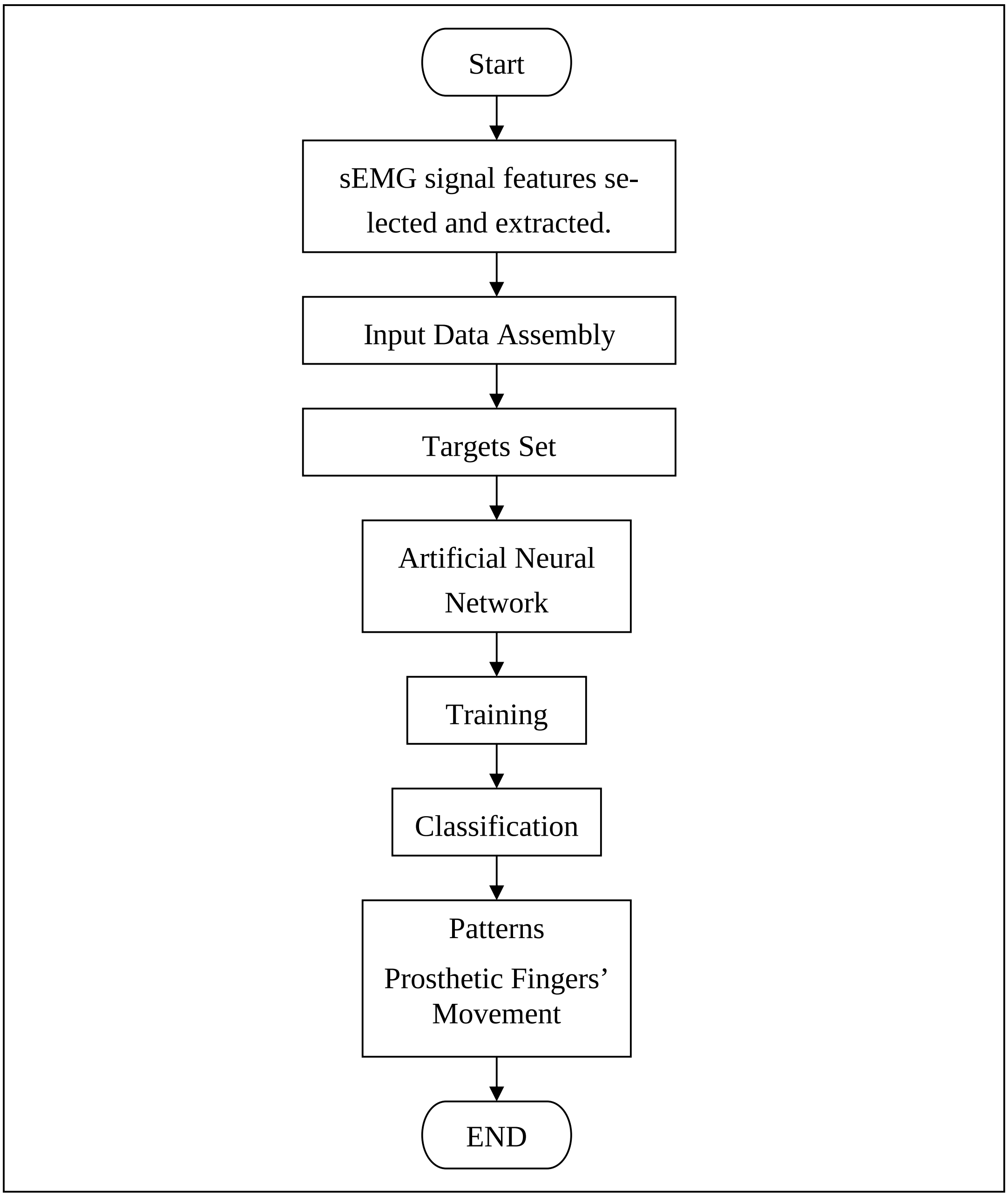

Start
sEMG signal features selected and extracted.
Input Data Assembly
Targets Set
Artificial Neural Network
Training
Classification
Patterns
Prosthetic Fingers' Movement
END

Figure 7.1: Flow chart of sEMG signal for pattern classification.

7.2 Fundamentals of Neural Network

This section aims to explain the fundamental principles of the neural network so that a better understanding of its functions can be established. Artificial Neural Network (ANN) is a computational modelling tool that has recently emerged and found extensive acceptance in many disciplines for modelling complex real-world problems [71, 72]. ANN can be defined as a structure consisting of compactly interconnected adaptive simple processing elements, called artificial neurons or nodes, capable of performing extremely parallel computations for data processing and knowledge representation [71]. The knowledge is stored in the interneuron connection strengths known as synaptic weights.

During the training process, the weights are tuned to the correct values, where a set of examples of input-output pairs flow through the model as well as being adjusted to minimise the error between the given network and the desired outputs. Once the weights have been set, the model is able to produce answers for input values which are not included in the training data [72].

ANN is a powerful tool that appeals to many researchers because it is able to solve complex problems such as non-linearity, high parallelism, robustness, fault and failure tolerance. In addition that it can handle imprecise and fuzzy information [71].

In this research, the features extracted from the sEMG muscle signals were trained or put through neural network as inputs (Figure 7.1). These inputs created an environment which had its own typical features or patterns in the layer called the hidden layer of the network. Every time a new input was trained, the hidden layer readjusted the previous features to become the output layer of the network. This process is iterated until all the data are trained and the output target is reached [55, 71]. The iteration process was shown in 'epochs' in Matlab's Artificial Neural Network toolbox. This type of iterate process is referred to as backpropagation (BP) [71] [55]. Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors or classify input vectors in an appropriate way as defined by users. Feedforward backpropagation is the most common neural network. The term 'feedforward' describes the process in which neurons in each layer are flowing forward through to the next connection layer and there is no backward connection. Figure 7.2 describes a simple three layer (one input layer with 6 neurons, one hidden layer with 4 neurons, and one output layer with 2 neurons) of 'feedforward' network.

A layer of 'input' unit is connected to a layer of 'hidden' unit, which is connected to a layer of 'output' unit, as shown in Figure 7.2. The activity of the input units represents the raw information that is fed into the network. The activity of each hidden unit is determined by the activities of the input unit and the weights on the connections between the input and hidden units. The behaviour of the output units depends on the activity of the hidden unit and the weights between the hidden and output units [73].

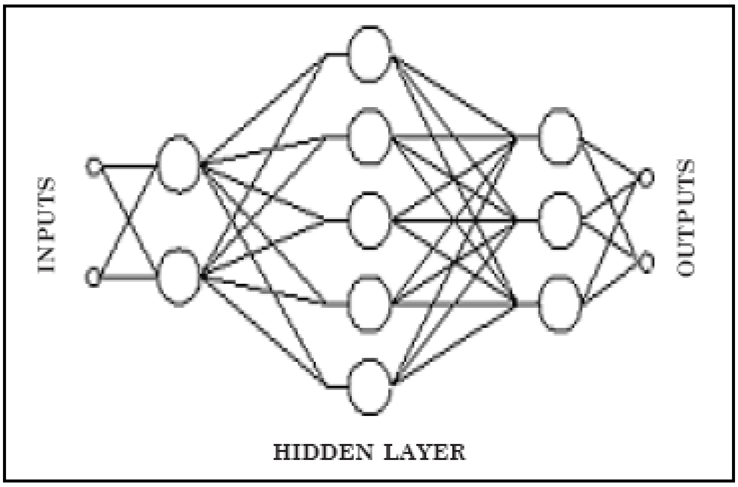


Figure 7.2: Example of simple 'FeedForward' network [73].

There are five general steps in the training process:

1. Assembling training data
   a. Already gathered in Chapter 6, Table 6.7.
   b. Data pre-processing, such as data normalisation in between-1 to 1.
2. Creating network object
   a. Designing output target.
   b. Designing and Initialising neural network.
3. Training network
   a. Also called batch training.
   b. Data splitting for training, such as 80%.
4. Simulating network
   a. Simulating output of trained network.

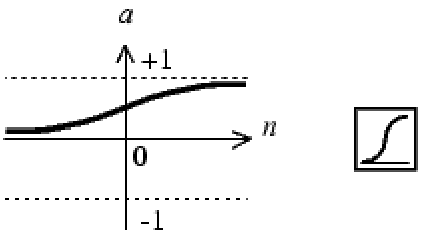
5. Validating and testing

   a. To compare results of the computed output of the network with the 20% new data or which have never been seen or used in training.

   b. Purpose of validating is to avoids data over-fitting problems.

   c. Purpose of testing is to simulate trained network.

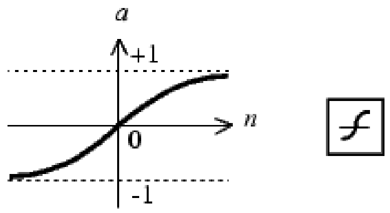
After understanding the training process of the neural network, the transfer functions used in the hidden layer and output layer are also of paramount importance as they are widely used in multilayer network. The transfer function of the backpropagation has log-sigmoid (*logsig*), tan-sigmoid *(tansig*) and linear transfer function (*pureline*) [55]. The function *logsig* generates outputs between 0 and 1 as the neuron's net input goes from negative to positive infinity (Figure 7.3). Alternatively, multilayer networks can use the function *tansig*, generating output between -1 and 1, as shown in Figure 7.4. Occasionally, the linear transfer function *purelin* is used in backpropagation networks, generating output between negative infinity value to positive infinity value (Figure 7.5).



MATLAB notation *a = logsig(n)*

$$\text{Mathematical expression } a = \frac{1}{1+e^{-n}}$$

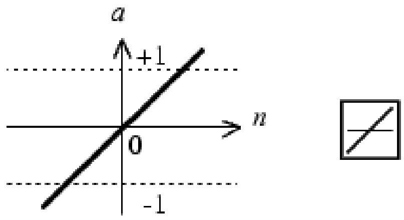
Figure 7.3: Log-sigmoid (*logsig*) transfer function (where *n* is input to transfer function and *a* is output data) [55].



MATLAB notation *a = tansig(n)*

Mathematical expression $a = \dfrac{1-e^{-n}}{1+e^{-n}}$

Figure 7.4: Tan-sigmoid (*tansig*) transfer function (where *n* is input to transfer function and *a* is output data) [55].



MATLAB notation *a = purelin(n)*

Mathematical expression $a = f(n) = n$

Figure 7.5: Linear transfer function (*purelin*) (where *n* is input to transfer function and *a* is output data) [55].

7.3 Data Assembling and Array Management

Before importing the number of samples to the neural network, data assembling and array management are essential. For example, the array of the input data must have some matrix form with the array of the output-target. The data in Table 6.7 will need to be organised according to the nature or architectural structure of the neural network program. The ANN requires the structure of the target to be set in the form of output vector based on Boolean notation. The input data of the ANN and the target are set depending on the data gathered and how they can be classified.

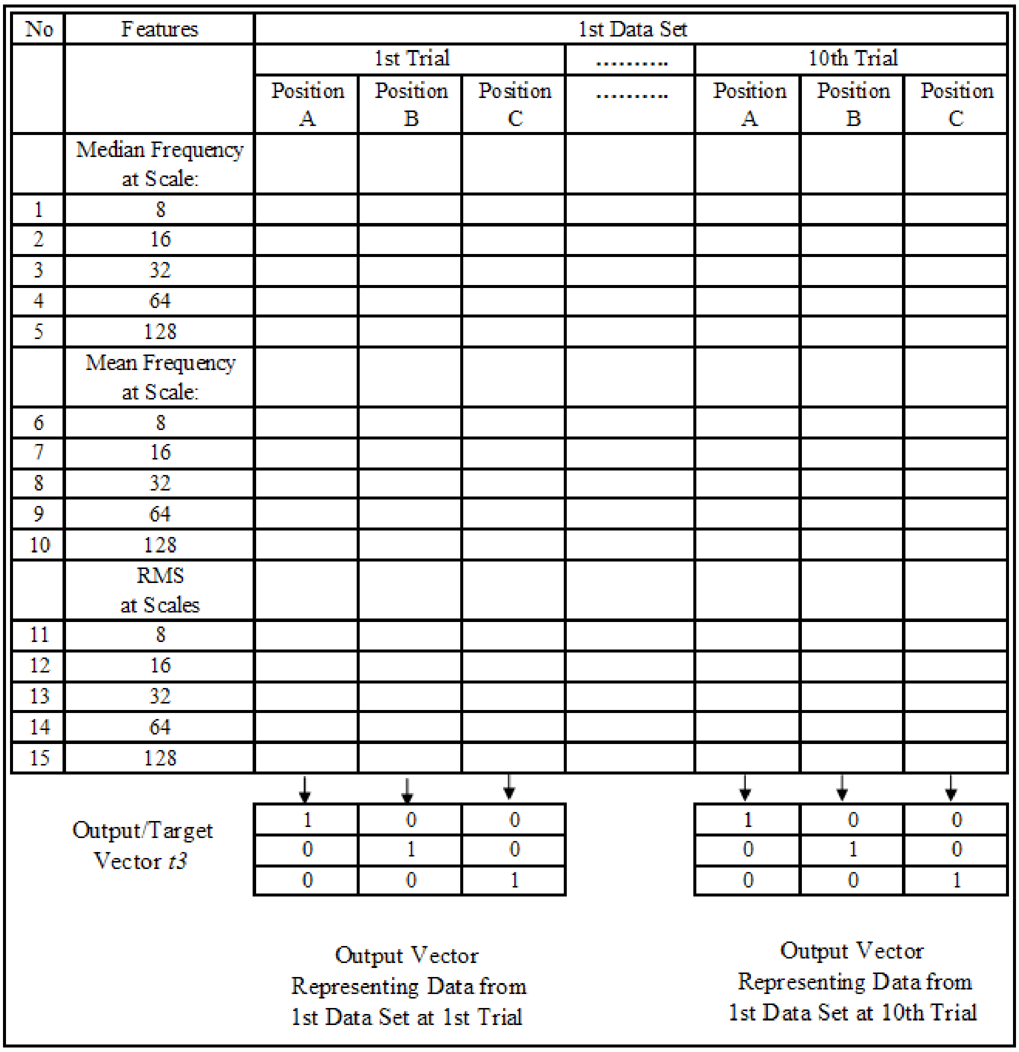
The data assembled for the training of the neural network were collected from one subject making three, five, seven, and nine different types of wrist joint movements from the left forearm muscles (see Chapter 6). The extracted features from these three, five, seven and nine different types of wrist joint movement became the target vector or the main output vector formed as Equations 7.1, 7.2, 7.3 and 7.4, respectively.

7.3.1 Three Patterns of Movement

Each target column consisted of 15 features from each set of data, which included the mean and median frequencies from the power spectrum and the RMS values of the wavelet coefficients at different scales of 8, 16, 32, 64 and 128. The structure of the network input array corresponding to the network output/target array is shown in Figure 7.6. The output target t_3 in Equation 7.1 will iterate 360 times to represent overall eighteen sets of samples, as two channels of sEMG sensors were used and each pattern in each set of data had to do ten trials (18 sets of samples x two channel of myoelectrodes x number of trials).

$$t_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ (Equation 7.1)}$$

where the first column is the target for the central point of wrist joint movement and the second and third for the full extension and full flexion, respectively, of the wrist joint. The wrist joint movement is illustrated in Figures 6.3 to 6.6.



No	Features	1st Data Set						
		1st Trial			....	10th Trial		
		Position A	Position B	Position C	....	Position A	Position B	Position C
	Median Frequency at Scale:							
1	8							
2	16							
3	32							
4	64							
5	128							
	Mean Frequency at Scale:							
6	8							
7	16							
8	32							
9	64							
10	128							
	RMS at Scales							
11	8							
12	16							
13	32							
14	64							
15	128							

Output/Target Vector *t3*

1	0	0
0	1	0
0	0	1

Output Vector Representing Data from 1st Data Set at 1st Trial

1	0	0
0	1	0
0	0	1

Output Vector Representing Data from 1st Data Set at 10th Trial

Figure 7.6: Array of input data corresponding to three output targets, t_3, vector.

7.3.2 Five Patterns of Movement

The array arrangement for five targets, t_5, vector is:

$$t_5 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(Equation 7.2)}$$

where:

1st Column: ***Central point***. Figures 6.3 & 6.4.

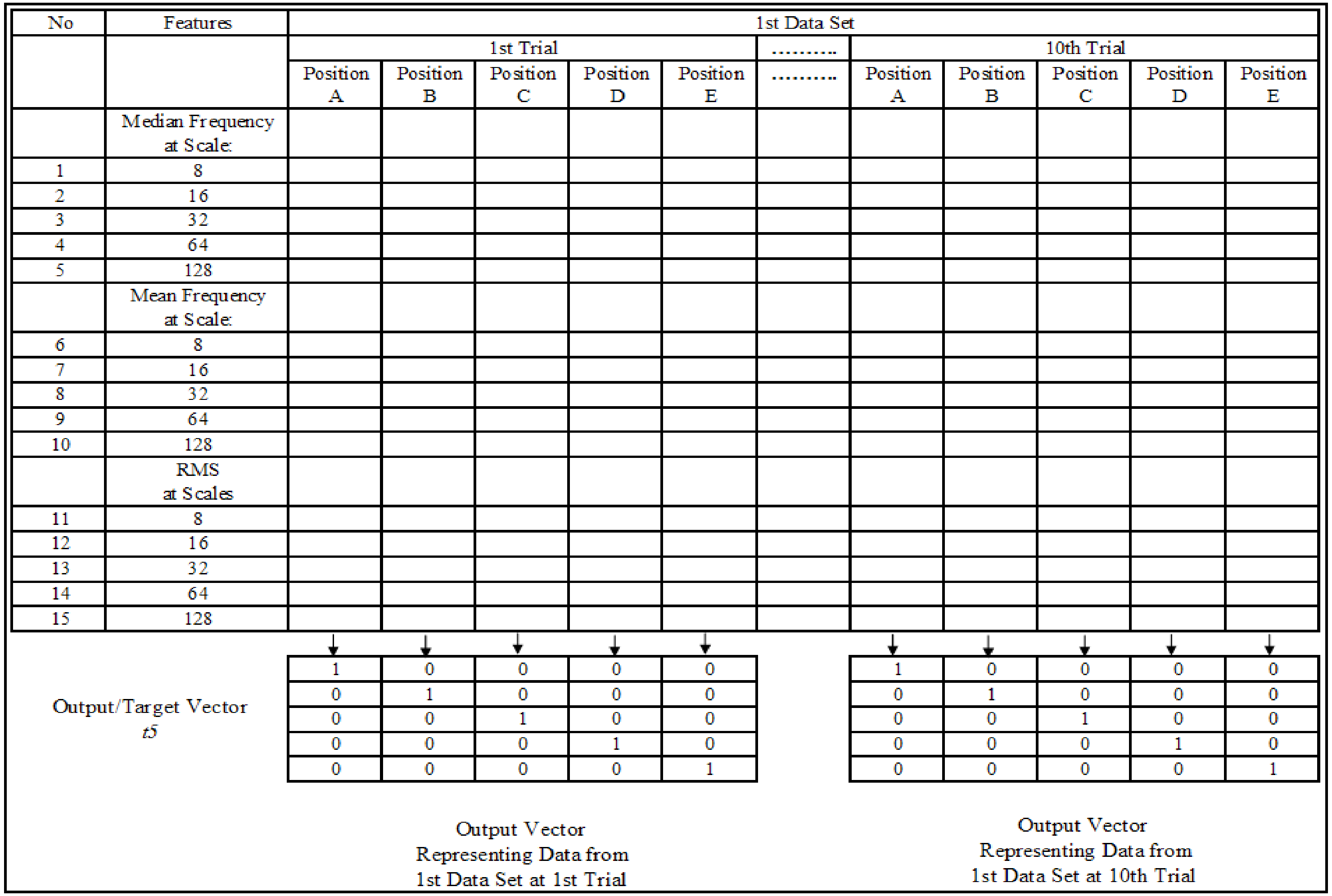
2nd Column: ***Full extension***. Figures 6.3 & 6.5.

3rd Column: ***Full flexion***. Figures 6.3 & 6.6.

4th Column: ***Fully up***. Figures 6.3 & 6.7.

5th Column: ***Fully down***. Figures 6.3 & 6.8.

Figures 6.4 to 6.8 illustrate these five patterns. Each target column consisted of 15 features from each data set, which included the mean and median frequencies from the power spectrum and the RMS values of the wavelet coefficients at different scales of 8, 16, 32, 64, and 128. The structure of the network input array corresponding to the network output target array is shown in Figure 7.7. The output target t_5 in Equation 7.2 will iterate 360 times to represent overall eighteen sets of data, due to use of two channels of sEMG sensors and each pattern in each set of data having to do ten trials.



No	Features	1st Data Set										
		1st Trial					……….	10th Trial				
		Position A	Position B	Position C	Position D	Position E	……….	Position A	Position B	Position C	Position D	Position E
	Median Frequency at Scale:											
1	8											
2	16											
3	32											
4	64											
5	128											
	Mean Frequency at Scale:											
6	8											
7	16											
8	32											
9	64											
10	128											
	RMS at Scales											
11	8											
12	16											
13	32											
14	64											
15	128											

Output/Target Vector *t5*

1	0	0	0	0	1	0	0	0	0
0	1	0	0	0	0	1	0	0	0
0	0	1	0	0	0	0	1	0	0
0	0	0	1	0	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1

Output Vector Representing Data from 1st Data Set at 1st Trial

Output Vector Representing Data from 1st Data Set at 10th Trial

Figure 7.7: Array of input data corresponding to five output target, t_5 , vector.

7.3.3 Seven Patterns of Movement

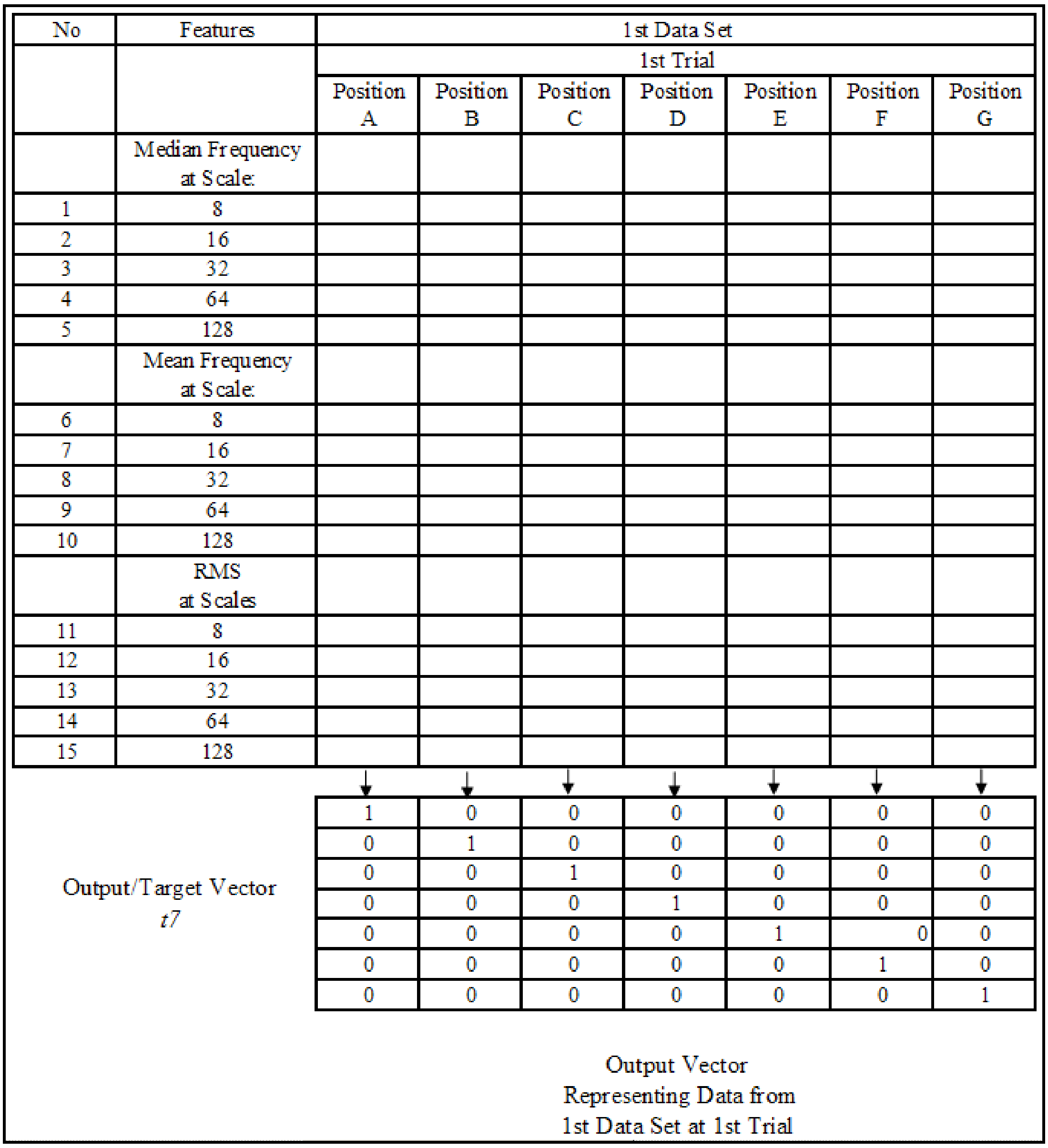
The seven targets vector t_7 is:

$$t_7 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(Equation 7.3)}$$

where:

1st Column:	***Central point***. Figures 6.3 & 6.4.
2nd Column:	***Full extension***. Figures 6.3 & 6.5.
3rd Column:	***Full flexion***. Figures 6.3 & 6.6.
4th Column:	***Fully u*****p**. Figures 6.3 & 6.7.
5th Column:	***Fully down***. Figures 6.3 & 6.8.
6th Column:	***$+45^0$ fully up***. Figures 6.3 & 6.9.
7th Column:	***$+45^0$ fully down***. Figures 6.3 & 6.10.

Figures 6.4 to 6.10 illustrate these seven patterns. Each target column consists of 15 features from each data set, including the mean and median frequencies from the power spectrum and the RMS values of the wavelet coefficients at different scales of 8, 16, 32, 64, and 128. The structure of the network input array corresponding to the network output target array is shown in Figure 7.8. The output target t_7 in Equation 7.3 will iterate 360 times to represent overall eighteen sets of data, due to use of two channels of sEMG sensors and each pattern in each set of data having to do ten trials.



No	Features	1st Data Set						
		1st Trial						
		Position A	Position B	Position C	Position D	Position E	Position F	Position G
	Median Frequency at Scale:							
1	8							
2	16							
3	32							
4	64							
5	128							
	Mean Frequency at Scale:							
6	8							
7	16							
8	32							
9	64							
10	128							
	RMS at Scales							
11	8							
12	16							
13	32							
14	64							
15	128							

Output/Target Vector
t7

1	0	0	0	0	0	0
0	1	0	0	0	0	0
0	0	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	1	0	0
0	0	0	0	0	1	0
0	0	0	0	0	0	1

Output Vector
Representing Data from
1st Data Set at 1st Trial

Figure 7.8: Array of input data corresponding to seven output target t_7 vector.

7.3.4 Nine Patterns of Movement

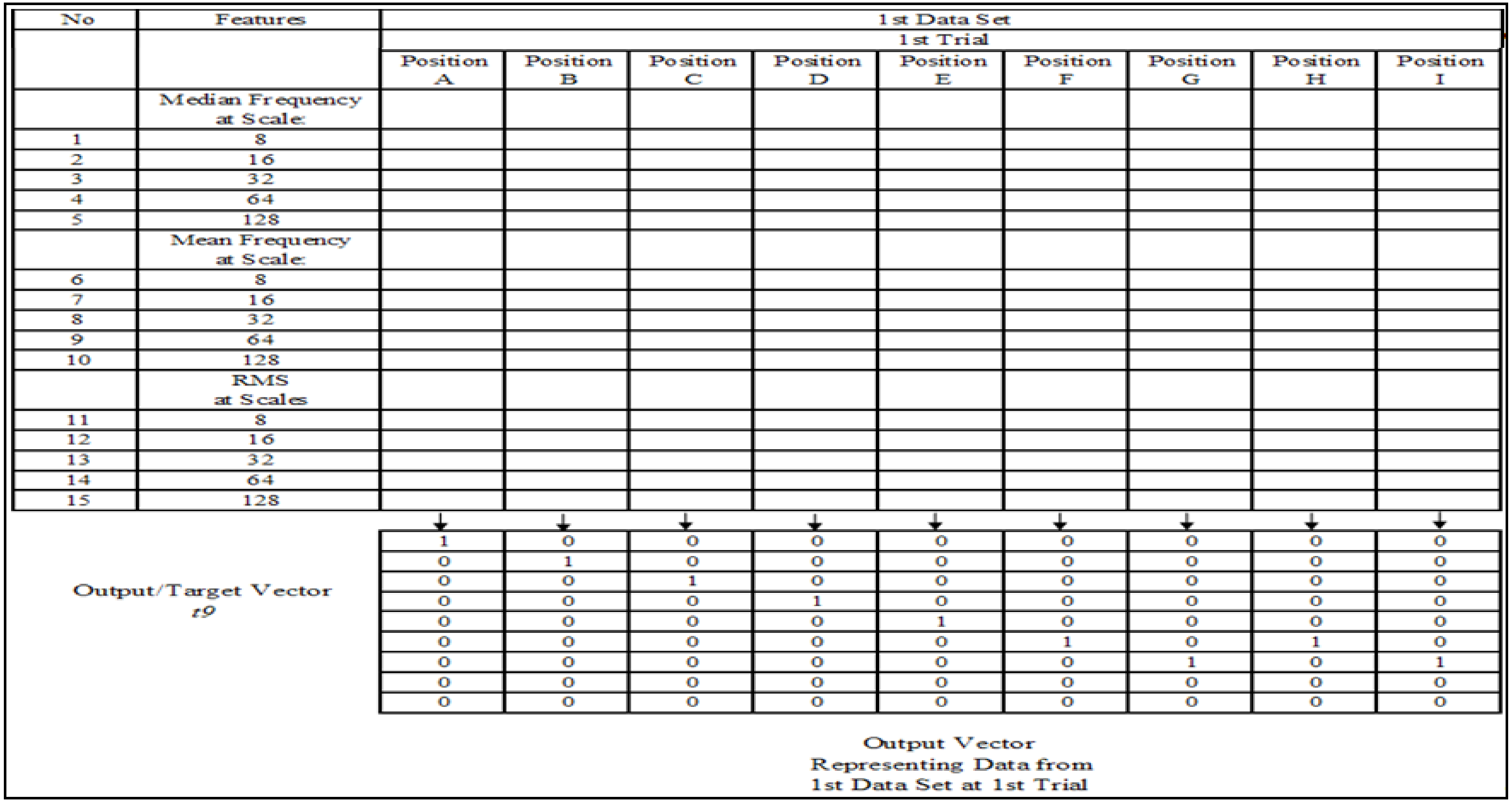
The nine targets t_9 vector is:

$$t_9 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(Equation 7.4)}$$

where:

1st Column:	***Central point***. Figures 6.3 & 6.4.
2nd Column:	***Full extension***. Figures 6.3 & 6.5.
3rd Column:	***Full flexion***. Figures 6.3 & 6.6.
4th Column:	***Fully u*****p**. Figures 6.3 & 6.7.
5th Column:	***Fully down***. Figures 6.3 & 68.
6th Column:	***+45^0 fully up***. Figures 6.3 & 6.9.
7th Column:	***+45^0 fully down***. Figures 6.3 & 6.10.
8th Column:	***-45^0 fully up***. Figures 6.3 & 6.11.
9th Column:	***-45^0 fully down***. Figures 6.3 & 6.12.

Figure 6.4 to 6.12 illustrate these nine patterns. Each target column consists of 15 features from each data set, including the mean and median frequencies from the power spectrum and the RMS values of the wavelet coefficients at different scales of 8, 16, 32, 64 and 128. The structure of the network input array corresponding to the network output target array is shown in Figure 7.9. The output target t_9 in Equation 7.4 will iterate 360 times to represent overall eighteen sets of data, two channels of sEMG sensors being used and each pattern in each set of data having to do ten trials.



No	Features	1st Data Set								
		1st Trial								
		Position A	Position B	Position C	Position D	Position E	Position F	Position G	Position H	Position I
	Median Frequency at Scale									
1	8									
2	16									
3	32									
4	64									
5	128									
	Mean Frequency at Scale									
6	8									
7	16									
8	32									
9	64									
10	128									
	RMS at Scales									
11	8									
12	16									
13	32									
14	64									
15	128									
		↓	↓	↓	↓	↓	↓	↓	↓	↓
	Output/Target Vector *t9*	1	0	0	0	0	0	0	0	0
		0	1	0	0	0	0	0	0	0
		0	0	1	0	0	0	0	0	0
		0	0	0	1	0	0	0	0	0
		0	0	0	0	1	0	0	0	0
		0	0	0	0	0	1	0	1	0
		0	0	0	0	0	0	1	0	1
		0	0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0	0

Output Vector
Representing Data from
1st Data Set at 1st Trial

Figure 7.9: Array of input data corresponding to nine output target t_9 vector.

7.4 Methodology for Designing and Training Neural Network

After the input and target data assembling, a unique methodology for designing a suitable neural network was created, based on understanding the fundamentals of neural network.

The method for designing and using ANN is shown in Figure 7.10. It is includes five general steps in the training process described in section 7.2.

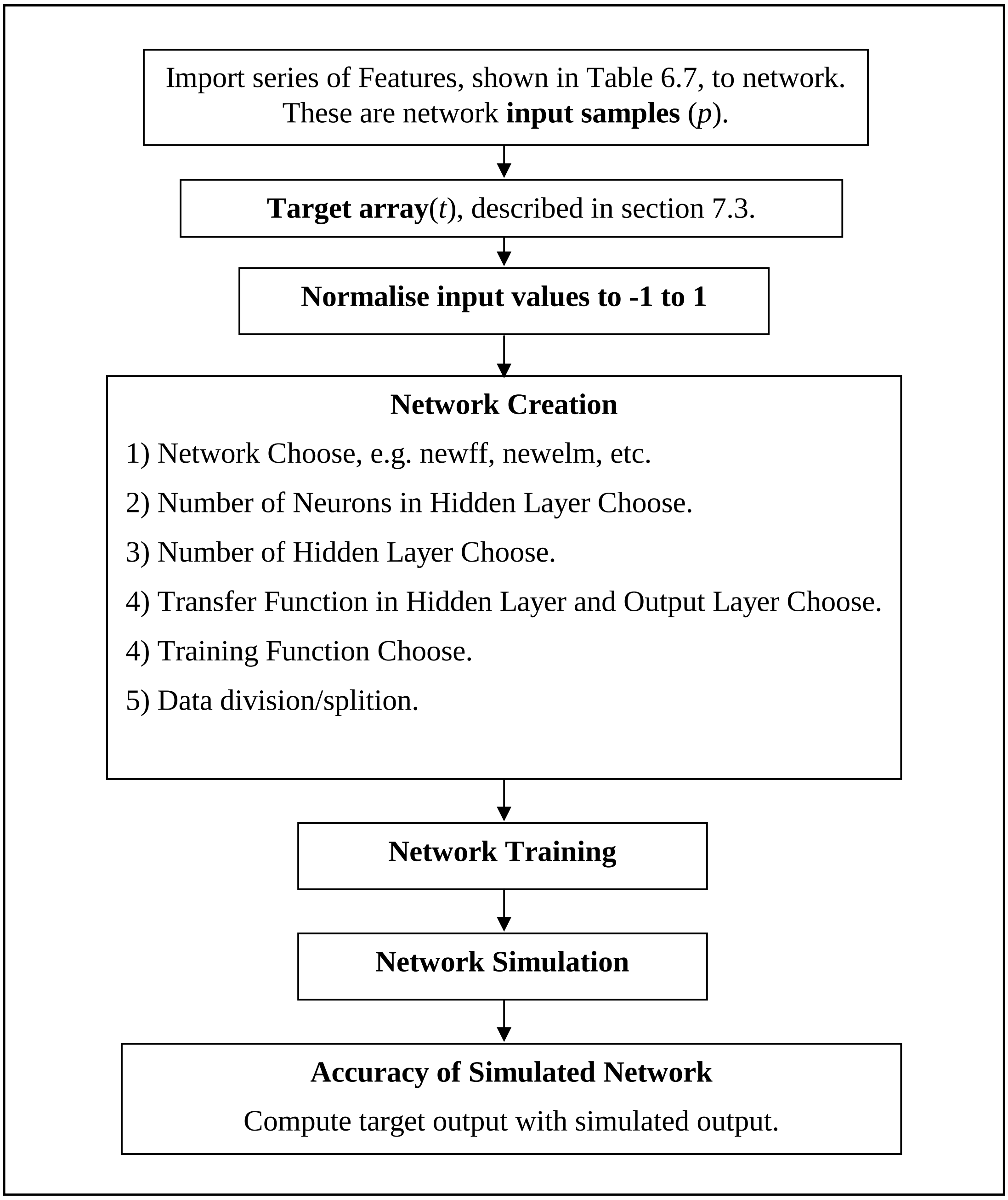
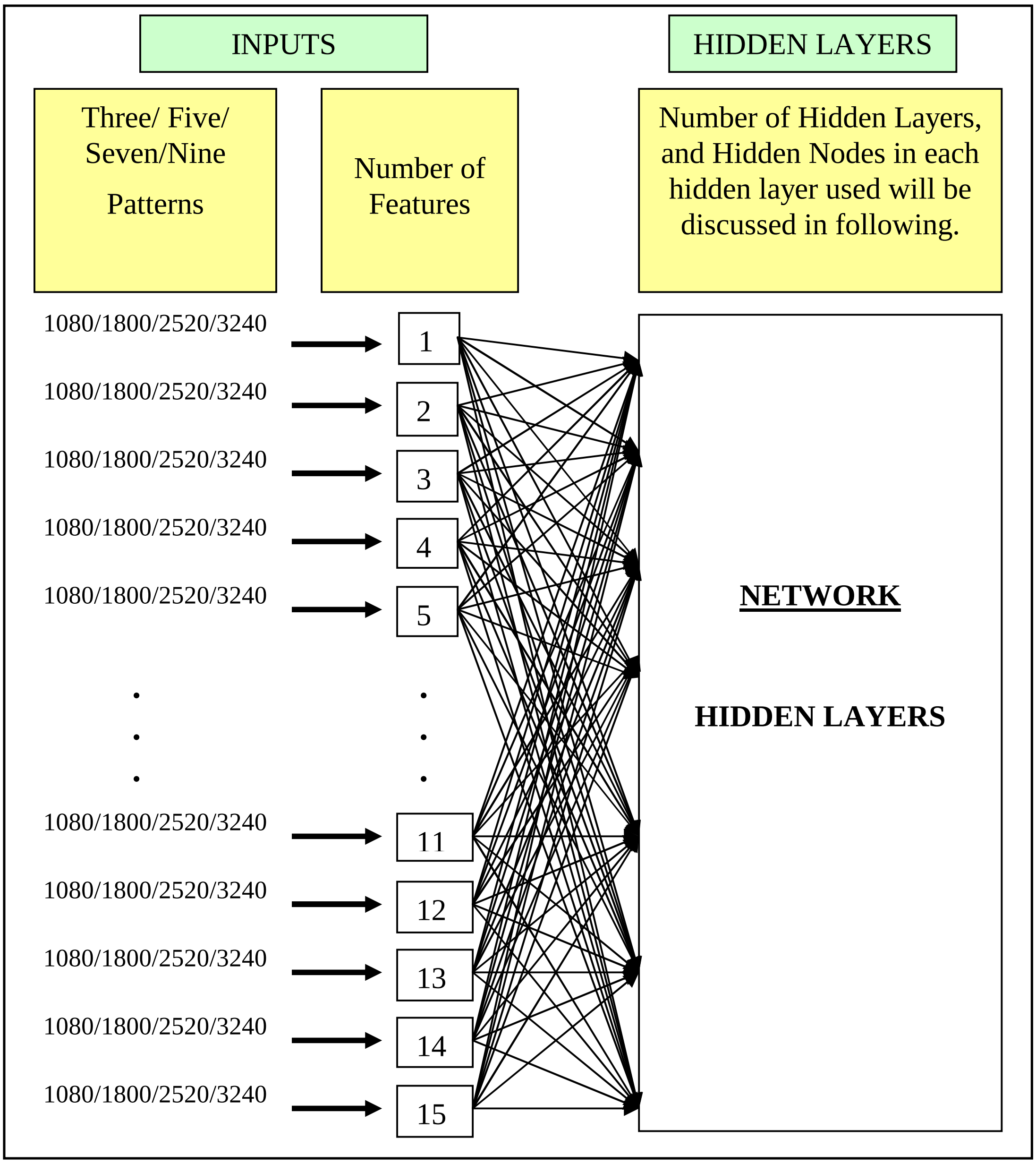

Import series of Features, shown in Table 6.7, to network.
These are network **input samples** (*p*).
Target array(*t*), described in section 7.3.
Normalise input values to -1 to 1
Network Creation
1) Network Choose, e.g. newff, newelm, etc.
2) Number of Neurons in Hidden Layer Choose.
3) Number of Hidden Layer Choose.
4) Transfer Function in Hidden Layer and Output Layer Choose.
4) Training Function Choose.
5) Data division/splition.
Network Training
Network Simulation
Accuracy of Simulated Network
Compute target output with simulated output.

Figure 7.10: Methodology for Designing ANN.

7.4.1 Network's Inputs Samples and Target's Array

The first and second stages of the methodology for designing an ANN were discussed in Section 7.3. For three patterns, each feature contains 1080 data from two sEMG sensors; for five, each feature contains 1800 data from two sEMG sensors; for seven, each feature contains 2520 data from two sEMG sensors; and for the nine patterns, each feature contains 3240 data from two sEMG sensors. Figure 7-11 shows the schematic diagram of ANN inputs data.



INPUTS

HIDDEN LAYERS

Three/ Five/ Seven/Nine Patterns

Number of Features

Number of Hidden Layers, and Hidden Nodes in each hidden layer used will be discussed in following.

Figure 7.11: Methodology for Designing ANN.

7.4.2 Network Creation

In network creation, five factors need to be taken into consideration. The first factor is choosing a suitable network, such as Feedforward Backpropagation (Matlab notation *newff*), Elman Backpropagation (*newelm*), Hopfield Network (*newhop*), etc. Feedforward Backpropagation was chosen in this project due to its popularity.

The second and third factors are the number of hidden layers chosen and number of hidden nodes used in each chosen hidden layer. Nodes (neurons) can exhibit complex global behaviour determined by the connections between the processing elements and element parameters. In the central nervous system, nodes (neurons) contain the most significant information processing elements [71].

These two factors cannot be derived from any mathematical or theoretical method and thus are obtained only by trial and error. As mentioned above [71, 72], the nodes of the hidden layer are very important; a network with *too few* hidden nodes would be incapable of differentiating between complex patterns leading to only a linear estimate of the actual trend. In contrast, if the network has *too many* hidden nodes it will follow the noise in the data due to over-parameterisation leading to poor generalisation for untrained data. The most popular approach to finding the optimal number of hidden nodes is by trial and error. As a result, to find the suitable number of nodes in our network, trial and error technique will use by a loop in the programming, as shown in Appendix E.

According to [72, 74], one hidden layer is sufficient to solve a lot of problems, such as function consisting of a finite collection of points. Function is continuous and is defined on a compact domain. By using trial and error, all three layers, i.e. one input, one hidden and one output layer, are already sufficient for our case.

The third factor is the transfer function on hidden and output layers chosen. Because the input values of the network will be pre-possessed to normalise the inputs values to in between -1 to 1, the transfer function of the hidden layer will use tan-sigmoid (*tansig*). Apart from that, according to the value of outputs targets, the transfer function of the output layer must be set to log-sigmoid (*logsig*).

The fourth factor is training function, which is used to train the network. Once the network weights and biases are initialised, the network is ready for training. During training, the weights and biases of the network are iterately adjusted to minimise the network performance function, such as mean square error (mse). The mse is the average squared error between the network output *y* and the target output *t*. There are several different training algorithms for feedforward networks. All of these algorithms use the gradient of the performance function to determine the way to adjust the weights to minimise performance [55]. The gradient is determined using a technique called backpropagation, which involves performing computations backwards through the network. A proper training function choice is dependent on their training parameters, such as learning rate, number of epochs, mse goal, training time, etc.

The fifth factor of the network creation is the division of the input data for training, validating and testing. In our case, 80% of the input samples are separated for training, 10% for validation, and the last 10% for testing. Table 7.1 shows the number of samples for training, validating and testing.

Table 7.1: Samples for training, validating and testing in four different groups of patterns.

Patterns	Total no. of samples	No. of samples for training	No. of samples for validating	No. of samples for testing
Three	1080	864	108	108
Five	1800	1440	180	180
Seven	2520	2016	252	252
Nine	3240	2592	324	324

7.4.3 Network Simulation and Accuracy of Simulated Network

The trained neural network can now be tested with the testing samples partitioned from the main dataset. The testing data are not used in training in any way and hence provide a dataset on which to test the network. This will give us a sense of how well the network will do when tested with data which are untrained.

After network training, 10% of the testing data, which are untrained, were used to simulate the trained network. After simulation, the target output, which is corresponds to the 10% untrained testing data to find the accuracy of the simulated output respond. The matlab notation for simulation is *out=sim(net, tstS.P).* Because the log-sigmoid transfer function was used in the output layer, the value of *out* is in between 0 to 1.

To determine the accuracy of the simulated output, *[y_out, i_out] = max(out)* in matlab notation was used to find the maximum simulated value and return the indices in vector *i_out* (see Appendix E). In fact, the vector i_out can be classed into the vector 'position' of the maximum value. The output value of simulated, *out*, has 108 columns and 3 rows of values during the classification of the three patterns; 180 and 5 during the classification of the five; 252 and seven during the classification of the seven; and 324 and 9 row of values during the classification of the nine patterns.

Besides, we also need to evaluate the vector 'position' for the target vector of the testing data by using *[y_t, i_t] = max(tstS.T)*. The value of the target vector contains just the value of either 1 or 0. Hence, *y_t* only has a value of 1, and the *i_t* is the vector 'position' of the *y_t*.

If the value of *i_ou*t and *i_t* is the same, then it is stored as ONE. If it is not, then it is stored ZERO. A simple example for counting the percentage correction value for the three target patterns is illustrated in Figure 7.12.

	Column 1	Column 2	Column 3	….	Column 118	Column 119	Column 120
Simulated Output value, *out*:	0.2606	0.1980	0.0434	….	0.1967	0.0805	0.0129
	0.2946	0.2323	0.4140	….	0.2313	0.3142	0.6625
	0.2177	0.4718	0.5821	….	0.4667	0.5792	0.4436

Row Position of the Maximum simulated output value out :

	Column 1	Column 2	Column 3	….	Column 118	Column 119	Column 120
***I_out*:**	2	3	3	….	3	3	2

	Column 1	Column 2	Column 3	….	Column 118	Column 119	Column 120
Target value, t_3, corre-sponding to *out*:	0	0	0	….	0	0	0
	1	1	1	….	1	0	0
	0	0	0	….	0	1	1

Row Position of the Maximum target value $\boldsymbol{t_3}$:

	Column 1	Column 2	Column 3	….	Column 118	Column 119	Column 120
***I_t*:**	2	2	2	….	2	3	3

Compare between *I_out* and *I_t*. If *I_out* - *I_t* =0, then store in value ONE. If *I_out* - *I_t* $\neq$ 0, then store in value ZERO:

Column 1	Column 2	Column 3	….	Column 118	Column 119	Column 120
ONE	ZERO	ZERO	….	ZERO	ONE	ZERO

Percentage Correction for Pattern Classification (%) =

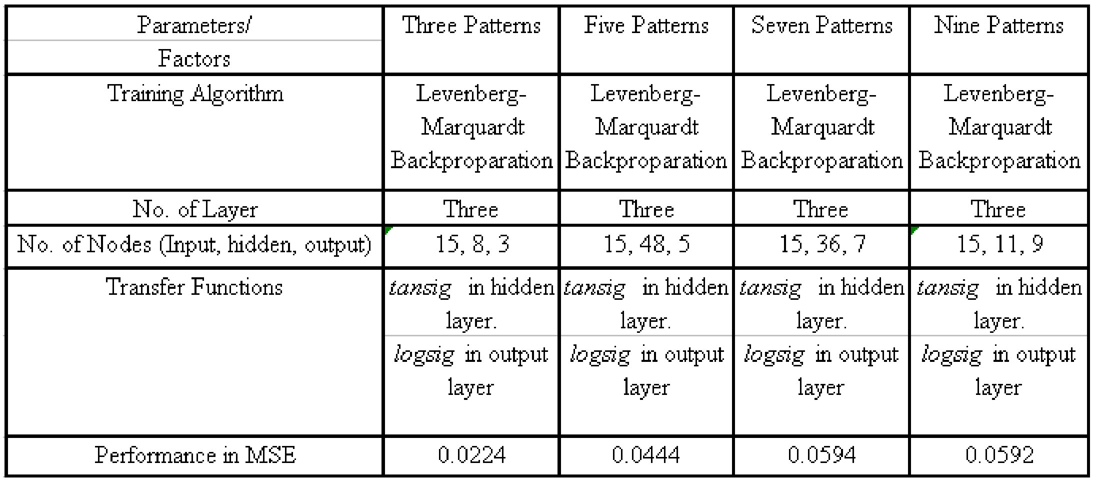
$$\frac{NumberofValue'ONE'}{TotalNumberofTestingColumn}x100$$

Figure 7.12: Example for counting percentage correction value for three patterns.

7.5 Results and Analysis of Pattern Classification

This section will discuss the total number of patterns which can be recognised by using neural network based on just two myoelectrodes. Table 7.2 shows the parameters which can perform the highest correct percentage of pattern recognition in four different groups, such as three, five, seven and nine patterns.

Table 7.2: Parameters/factors in three, five, seven and nine patterns.



Parameters/ Factors	Three Patterns	Five Patterns	Seven Patterns	Nine Patterns
Training Algorithm	Levenberg-Marquardt Backproparation	Levenberg-Marquardt Backproparation	Levenberg-Marquardt Backproparation	Levenberg-Marquardt Backproparation
No. of Layer	Three	Three	Three	Three
No. of Nodes (Input, hidden, output)	15, 8, 3	15, 48, 5	15, 36, 7	15, 11, 9
Transfer Functions	*tansig* in hidden layer. *logsig* in output layer	*tansig* in hidden layer. *logsig* in output layer	*tansig* in hidden layer. *logsig* in output layer	*tansig* in hidden layer. *logsig* in output layer
Performance in MSE	0.0224	0.0444	0.0594	0.0592

The recognition results in Table 7.3 show the number of correct recognitions. The 1st set of samples contains 60 samples for three patterns, 100 for five, 140 for seven and 180 for nine. The 2nd set of samples contains the current and previous groups of samples, hence, the 2nd set of samples, contains 120 samples for three patterns, 200 samples for five patterns, 280 patterns for seven patterns, and 360 samples for nine patterns. The 3rd set of samples contains the current sample and the 2nd set of sample, the 4th set of sample contain the current sample and the 3rd set of samples, and so on. It means that the final total number of samples, which is shown in Table 7.1 are the samples in 17th set.

From the results shown in Table 7.3, the average correct recognition rate is above 95% for three patterns and above 85% for five. However, the correct recognition rates less than 80% are 70% for seven and 60% for nine patterns.

Based on the literature review and application complexities, recognition rates above 70% (usually in the range of 70-90) will be acceptable. The desirable correct recognition rate is to be as high as possible to minimise the errors involved. For example, 80% correct recognition rate means that there is a possible two-trial error in each ten trials. Hence, to set a threshold it was considered that the correct recognition rate of 80% or higher is acceptable for the current application. However, the rate between 70% to 79% could be considered as good, and below 70% as poor.

Table 7.3: Correct pattern recognition percentage in four different groups of patterns by different sets of samples.

Set No.	**3 Patterns (%)**	**5 Patterns (%)**	**7 Patterns (%)**	**9 Patterns (%)**
1	100.00	80.00	71.43	50.00
2	100.00	90.00	60.71	55.56
3	100.00	83.33	66.67	57.41
4	100.00	82.50	67.86	58.33
5	100.00	90.00	61.43	57.78
6	94.44	90.00	67.86	58.33
7	97.62	88.57	71.43	58.73
8	100.00	90.00	73.21	59.03
9	98.15	88.89	71.43	64.20
10	96.67	92.00	72.14	65.00
11	96.97	89.09	72.73	63.13
12	98.61	90.83	74.40	63.43
13	96.15	89.23	73.63	62.82
14	98.81	93.57	75.51	62.70
15	97.78	90.00	76.19	64.07
16	95.83	90.00	72.77	63.89
17	95.10	88.82	73.95	64.05
18	97.22	87.22	72.22	62.65
Average	97.96	88.56	70.86	60.62

The lines in Figure 7.13 are not straight due to different timing of the collection of the 18 sets of data. For example, the 2nd times data collected (include in 2nd set of data) and the 1st times data colleted were taken on different days, and the 3rd times data collected (include in 3rd set of data) and 4th times data collected (include in 4th set of data) were taken on the same day but different time. Owing to the difficulty in placing the same myoelectrodes at the same precise desired position throughout the whole process, variation in the location and variation in the strength of muscle, possibly caused by overuse, leads to variation in the results.

The seven patterns line (yellow) shows that the rate of correct recognition increased from set 1 to set 15 of the samples, due to the increasing number of samples for training. However, it also cannot prove that increasing the number of training samples will increase the rate of pattern recognition, such as for sets 15 to 18. The same problem occurred with the nine patterns (cyan colour line) as well.

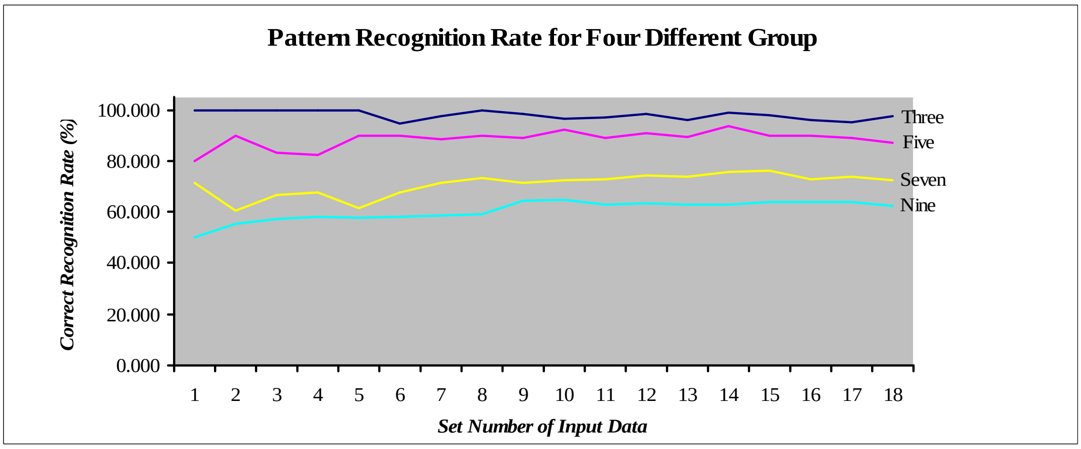


Figure 7.13: Correct pattern recognition rate from Table 7.3.

Patterns F, G, H and I shown in Figures 6.9 to 6.12 are very hard to control in precise 45^0 of supination-pronation rotation and sometimes because their signals are similar to the signals of Patterns D and E (Figures 6.7 and 6.8). All these problems can also cause the seven and nine patterns not to have a high rate of correct recognition. These types of problems also occurred in [75].

7.6 Summary

The objective of this chapter was to execute the last stage of the sEMG’s signal pattern classification by using ANN in order to discuss the total number of patterns, which can be recognised by using just two myoelectrodes.

The fundamentals of the Artificial Neural Networks were presented in order to discuss and understand the factors/parameters used, such as number of hidden layers, number of nodes, activation functions, learning algorithm, etc.. Based on this, the methodology for designing the required neural network was created and presented.

The management of the arrays of the input samples and the output-targets is of crucial importance. The total output-target and input sample arrays of the four groups of patterns are presented.

The first group of wrist joint movement patterns is classified into three: the wrist joint at central point, fully extended and fully flexed. The correspondence between the wrist joint movements and the fingers' movements are also presented in this chapter.

The second group of patterns is classified into five wrist joint movements, which are wrist joint at central point, fully extended, fully flexed, fully up and fully down. The correspondence between the wrist joint movements and the fingers' movements are discussed.

The third group of patterns to classify the seven different types of wrist joint movements, which are wrist joint at central point, fully extended, fully flexed, fully up, fully down, fully $+45^0$ up and fully $+45^0$ down. The correspondence between the wrist joint movement and the fingers movements are presented.

The last group of patterns is classified into nine different types of wrist joint movement. The correspondence between the wrist joint movements and the fingers' movements are covered.

The last section of this chapter discusses the recognition results for the above four groups of pattern. Only the three and five patterns' groups achieved up to 85% accuracy. The seven and nine patterns' groups achieved on average just 70% and 60% accuracy rate, respectively. This could be improved by using more complex ANN.

The factors affecting the rate of correct recognition in seven and nine patterns' groups are the sitting of the myoelectrodes, muscle fatigue, the 45^0 wrist joint turn and the problems of similar signal.

CHAPTER 8
Summary and Conclusions

8.1 Assessment of Research Objectives

In chapter one a set of five research objectives were defined. This section evaluates the extent to which the objectives were achieved.

1. Prosthetic Finger Design

   This objective was addressed in Chapter 3. A preliminary prosthetic adult middle finger was designed and developed and rapid prototyped to allow the hybrid actuation mechanism, comprising DC and SMA actuators, to increase the number of active DOF to provide the functionality of a real hand in order to mimic movements such as coarse and fine grasping. To avoid complexity in hand design and control, a prosthetic finger design was proposed and rigorously studied. It was postulated that the hand design and control would be similar to that of a single prosthetic finger. The dimensions of the finger design were based on the author's middle finger and it was custom-made because of large variations in the size of human hands and fingers.

2. Hybrid Actuation System Design

   The actuation mechanism used was a hybrid system consisting of a DC and a SMA actuator. This hybrid actuation system would not only allow increasing the DOF of the prosthetic finger but also allow the prosthetic finger to have sufficient grasping force of an object. All of this was presented in Chapter 4. However, the adult middle finger PIP joint needed to have up to 0.28Nm torque to produce up to 32N force in the fingertip, which required it to have up to ten layers with one loop/route of SMA wire actuator and it was extremely difficult to find a small dimension material and non-conductivit mechanism strong enough (especially the multi-layer pulley) to hold the force of the SMA wires.

3. Hybrid Actuation Control System Design

A control system is a device to manage, command, direct or regulate the behaviour of other devices or systems. The system is designed to control the hybrid actuation mechanism in order to position the finger, especially fingertip, correctly. A PID algorithm was used to control the movement of the DC actuator to ‘bring’ the MCP joint to the correct position. A simple open loop with the value of the input voltage and the output linear displacement change of the SMA actuator was used to control movement of the PIP joint. This objective was addressed in Chapter 5.

4. Control Input Signal

This objective was covered in Chapter 6. The electrical activity generated by contracting muscle in the forearm was to be used as a control input signal for the prosthetic fingers. The electrical activity of the muscle is called a myoelectric signal and is monitored using surface Electromyography (sEMG) sensors (myoelectrodes). Only two myoelectrodes were to be used to facilitate the amputees’ or users’ training and end-use. These two myoelectrodes sensors were placed on the skin of the left forearm muscles, which are *extensor carpi radialis longus* and *flexor carpi ulnaris*. These two muscles are the thickest muscles of the human forearm.

5. Control Input Signal Processing

The raw surface EMG (sEMG) signal contains valuable information. However, this information is useful only if it can be quantified by a signal processing method to achieve the accurate and actual sEMG signal to be used for control.

The processing of sEMG signals can be divided into a three-stage process. The first step is data acquisition, including amplification, analogue to digital conversion and signal conditioning. One subject with healthy left forearm muscle took part in this experiment and eighteen sets of data were collected according to the nine different types of wrist joint movement. The purpose was to investigate the number of patterns that can be recognised, based on use of only two myoelectrodes.

Secondly, for features extraction of the sEMG signal, a technique called Continuous Wavelet Transform (CWT) with its five different scales was used, the five different scales number were 8, 16, 32, 64 and 128.

Thirdly, a features selection stage is important for Artificial Neural Network (ANN) pattern classification. The features selected were RMS values, mean and median frequencies of the power spectrum in scales number 8, 16, 32, 64 and 128.

6. Control Input Signal Pattern Recognition

The control input signal pattern recognition was considered using ANN pattern classification technique to train and test the varying patterns of human finger movement (thumb not considered), and this was addressed in Chapter 7. From the experimental results, the three and five patterns' groups achieved up to 85% accuracy. The seven and nine patterns' groups achieved on average just 70% and 60% accuracy rate, respectively using the simple ANN configuration.

8.2 Conclusions

The research work reported in this thesis proposes the design and development of an intelligent prosthetic hand using novel actuation and control.

The two areas to which the research has contributed in the field of prosthetic hands are:

1. A novel hybrid actuation system where both DC motors and SMAs could be used to provide a prosthetic hand with more degrees of freedom and dexterity to get closer in functionality to a natural hand. This feature could particularly be more attractive for small prosthetic hands where there is limited space available for the actuators.

2. The second aspect is a research on myoelectric signals to be used for control. It is proposed in this research that two myoelectrodes to be used but the signal processing and ANN to be further explored to improve the number of features that could be successfully extracted from the two myoelectrodes and used in control.

The results have shown promising outcomes for controlling up to nine independent movements.

For the proof of concept, a system with a middle finger prototype was considered. Testing through simulation and physical models revealed a number of limitations. The prosthetic finger had 3 DOF in contrast to 4 DOF of the human finger, since the movement of abduction-adduction in the MCP joint was not necessary for the middle finger to accomplish grasping functionality.

In order to verify the first novel actuation approach, which aimed to increase the number of DOF, a hybrid actuation system consisting of DC and SMA actuators was used and applied into the middle finger prototype. The DC actuator was placed in the palm associated with a suitable worm gear to control the flexion-extension movement of the metacarpophalangeal (MCP) joint. The SMA actuator was also placed in the palm but at a different level from that of the DC actuator and was attached to tendon cable in order to control the PIP joint and DIP joint flexion movement. A spring is connected to the bottom side of the MP to control extension movement. The movement of the DIP joint is coupled with the PIP joint through a pulley.

The control input signal used the electrical activity of the muscle i.e. the myoelectric signal and was monitored using surface Electromyography (sEMG) sensors (myoelectrodes). Only two myoelectrodes were used in order to facilitate to enhance the user-friendliness of the device.

8.3 Future Works

Future work will primarily concern further testing and development of the research. From the conclusions a number of areas have been identified:

1. Integrating and modelling of the developed system

   Since the control of the hybrid actuation was analysed in a single individual in Chapter 5, it was not correlated to the sEMG pattern recognition. Hence, in future work, integration and modelling of the hybrid actuation and sEMG pattern recognition are needed. Besides, in order to connect the hybrid actuation in one power supply, voltage regulators are needed to split the supply voltage into two different voltages for the DC and SMA actuators.

2. SMA actuator design and control

Preliminary experimental results in Chapter 4 are based on single and double layer SMA wire of the actuator. That is because suitable material and small non-conductive mechanisms, such as brass or multi-layers pulley, could not be found to be readily available. Furthermore, since there was no position sensor for the PIP joint, an open-loop control was used. Therefore, in future work, finding a suitable position sensor connected on the PIP joint is needed for close-loop control.

3. Prosthetic finger design

Reconstruction of the finger design is needed to allow the DIP joint to be in fact passively connected to the PIP joint with pulleys without any other external mechanism. Besides, the new finger design must allow positioning of the tendon cable and a new worm gear and position sensor on the PIP joint.

4. Force sensor

A force sensor connecting on the fingertip is needed to define accurately the grasping force based on the hybrid actuation mechanism.

5. Self-organising map

A self-organising map (SOM) is needed; the purpose of which is to define the quality of the features extracted from the myoelectric signal. The meaning of ‘quality’ is defined as whether the features extracted are useful/meaningful, whether they overlap and whether they have good clustering. This can probably improve the pattern recognition of the nine patterns.

REFERENCES

[1] D. G. Smith, J. W. Michael, and J. H. Bowker, *Atlas of Amputation and Limb Deficiencies - Surgical, Prosthetic, and Rehabilitation Principles*: American Academy of Orthopaedic Surgeons, 2004.

[2] A. Muzumdar, *Powered Upper Limb Prostheses- Control, Implementation and Clinical Application*: Springer, 2004.

[3] C. Pfeiffer, K. DeLaurentis, and C. Mavroidis, "Shape Memory Alloy Actuated Robotic Prostheses: Initial Experiments," in *Robotics and Automation,* 1999, pp. 2385-2391.

[4] M. C. Carrozza, B. Massa, P. Dario, M. Zecca, S. Micera, and P. Pastacaldi, "A two DoF finger for a biomechatronic artificial hand," *Technology and Health Care,* vol. 10, pp. 77-89, 2002.

[5] N. A. Andrade, G. A. Borges, F. A. d. O. Nascimento, A. R. S. R. Romariz, and A. F. d. Rocha, "A new biomechanical hand prosthesis controlled by surface electromyographic signals," in *Proceedings of the 29th Annual International Conference of the IEEE EMBS,* 2007, pp. 6141-6144.

[6] V. Bundhoo, E. Haslam, B. Birch, and E. J. Park, "A shape memory alloy-based tendon-drive actuation system for biomimetic artificial fingers Part 1: Design and Evaluation," *Cambridge Journals,* vol. 27, pp. 131-146, Jan 2009 2008.

[7] K. J. DeLaurentis and C. Mavroidis, "Mechanical Design of a Shape Memory Alloy Actuated Prosthetic Hand," *Technology and Health Care,* vol. 10, pp. 91-106, June 2002.

[8] V. Bundhoo and E. J. Park, "Design of an Artificial Muscle Actuated Finger Towards Biomimetic Prosthetic Hands," in *Proceedings of. Advanced Robotics, ICAR '05, 12th International Conference,* 2005, pp. 368-375.

[9] Y. Bar-Cohen, T. Xue, M. Shahinpoor, J. O. Simpson, and J. Smith, "Low-mass muscle actuators using electroactive polymers (EAPs)," in *Proceedings of SPIE's 5th Annual International Symposium on Smart Structures and Materials,* San Diego, 1998, pp. 3324-3332.

[10] A. E. Kritter, "Myoelectric Prostheses," *The Journal of Bone & Joint Surgery,* vol. 67, pp. 654-657, 1985.

[11] C. Connolly, "Prosthetic hands from Touch Bionics," *Industrial Robot-an International Journal,* vol. 35, pp. 290-293, 2008.

[12] J. B. Rosmarin and H. H. Asada, "Synergistic Design of a Humanoid Hand with Hybrid DC Motor-SMA Array Actuators Embedded in the Palm," in *International Conference on Robotics and Automation*, Pasadena, CA, USA, 2008, pp. 773-778.

[13] A. Kargov, T. Wemer, C. Pylatiuk, and S. Schulz, "Development of a miniaturised hydraulic actuation system for artificial hands," *Sensors and Actuators a-Physical,* vol. 141, pp. 548-557, Feb 2008.

[14] G. K. Klute, J. M. Czerniecki, and B. Hannaford, "McKibben Artificial Muscle: Pneumatic Actuators with Biomechanical Intelligence," in *International Conference on Advanced Intelligent Mechatronics (AIM'99),* Atlanta, USA, 1999, pp. 221-226.

[15] G. Roznowski and M. Drzewiecki, "A New Approach to the Prosthetic Finger Design," in *Modern Problems of Radio Engineering, Telecommunications and Computer Science Conference,* 2004, pp. 545-547.

[16] K. T. O'Toole and M. M. McGrath, "Mechanical Design and Theoretical Analysis of a Four Fingered Prosthetic Hand Incorporating Embedded SMA Bundle Actuators," *International Journal of Mathematical, Physical and Engineering Sciences,* vol. 1, pp. 83-90, 2007.

[17] F. Rothling, R. Haschke, J. J. Steil, and H. Ritter, "Platform Portable Anthropomorphic Grasping with the Bielefed 20-DOF Shadow and 9-DOF TUM Hand," in *International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, 2007, pp. 2951-2956.

[18] B. Hannaford, J. M. Winters, C.-P. Chou, and P.-H. Marbot, "The Anthroform Biorobotic Arm: A system for the study of spinal circuits," *Annals of Biomedical Engineering,* vol. 23, pp. 399-408, Jul-Aug, 1995 1995.

[19] K. J. DeLaurentis, A. Fisch, J. Nikiteczuk, and C. Mavroidis, "Optimal Design of Shape Memory Alloy Wire Bundle Actuators," in *Proceedings of. Robotics and Automation. ICRA '02. IEEE International Conference*, 2002, pp. 2363-2368.

[20] M. B. I. Reaz, M. S. Hussain, and F. Mogd-Yasin, "Techniques of EMG Signal Analysis-Detection, Processing, Classification and Applications," *Biological Procedures Online,* vol. 8, pp. 11-35, December 2006.

[21] C. J. D. Luca, "The Use of Surface Electromyography in Biomechanics," *Journal of Applied Biomechanics,* vol. 13, pp. 135-163, 1997.

[22] M. Knaflitz and G. Balestra, "Computer analysis of the myoelectric signal," *Micro, IEEE,* vol. 11, pp. 12-15, 48-58, 1991.

[23] W. C. Shiel Jr. and M. C. Stoppler, "Electromyogram (EMG)," 2009.

[24] R. F. Weir, P. R. Troyk, G. DeMichele, and T. Kuiken, "Implantable myoelectric sensors (IMES) for upper-extremity prosthesis control- preliminary work," in *Proceedings of the 25th Annual International Conference of the IEEE embs,* Mexico, 2003.

[25] P. Kampas, "The Optimal Use of Myoelectrodes," *Translation of: Med. Orth. Tech.,* vol. 121, pp. 21-27, 2001.

[26] C. Castellini and P. van der Smagt, "Surface EMG in advanced hand prosthetics," *Biological Cybernetics,* vol. 100, pp. 35-47, Jan 2009.

[27] M. C. Carrozza, B. Massa, S. Micera, M. Zecca, and P. Dario, "The Development of a Novel Prosthetic Hand-Ongoing Research and Preliminary Result," *Mechatronics,* vol. 7, pp. 108-114, 2002.

[28] F. V. G. Tenore, A. Ramos, A. Fahmy, S. Acharya, R. Etienne-Cummings, and N. V. Thakor, "Decoding of Individuated Finger Movements Using Surface Electromyography," *IEEE Transactions on Biomedical Engineering,* vol. 56, pp. 1427-1434, MAY 2009 2009.

[29] C. Martelloni, J. Carpaneto, and S. Micera, "Classification of upper arm EMG signals during object-specific grasp," in *2008 30th Annual International Conference of the Ieee Engineering in Medicine and Biology Society, Vols 1-8,* 2008, pp. 5061-5064.

[30] P. Shenoy, K. J. Miller, and R. P. N. Rao, "Online Electromyographic Control of a Robotic Prosthesis," *IEEE Transactions of Biomedical Engineering,* vol. 55, pp. 1128-1135, March, 2008 2008.

[31] R. J. Smith, F. Tenore, D. Huberdeau, and R. Etienne-Cummings, "Continuous Decoding of Finger Position from Surface EMG Signals for the Control of Powered Prostheses," in *Engineering in Medicine and Biology Society, EMBS Conference*, Vancouver, British Columbia, Canada, 2008, pp. 197-200.

[32] S. M. M.Zecca, M.C.Carrozza, and P.Dario, "Control of Multifunctional Prosthetic Hand by processing the electromyographic signal," *Critical Review in Biomedical Engineering,* vol. 30, pp. 459-485, 2002.

[33] K. E. P.Parker, and B.Hudgins, "Myoelectric Signal Processing for Control of Powered Limb Prostheses," *Journal of Electromyography and Kinesiology,* vol. 16, pp. 541-548, 2006.

[34] M. Khezri and M. Jahed, "A Novel Approach to Recognize Hand Movements via sEMG Patterns," *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE,* pp. 4907-4910, 22-26th Aug. 2007 2007.

[35] L. Biometrics, "EMG Sensor Operating Manual Type no. SX230," 2006.

[36] M. Asghari Oskoei and H. Hu, "Myoelectric Control System - A survey," *Biomedical Signal Processing and Control,* vol. 2, pp. 275-294, 2007.

[37] M. Zardorshti-Kermani, B. C. Wheeler, and R. M. Hashemi, "EMG Feature Evaluation for Movement Control of Upper Extremity Prostheses," *IEEE Transactions on Rehabilitation Engineering,* vol. 3, pp. 324-333, December 1995 1995.

[38] M. J. M.Khezri, N.Sadati, "Neuro-Fuzzy Surface EMG Pattern Recognition for Multifunctional Hand Prosthesis Control," *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium* pp. 269-274, 4-7 June 2007 2007.

[39] K. Englehart, B. Hudgins, and P. A. Parker, "A Wavelet-Based Continuous Classification Scheme for Multifunction Myoelectric Control," *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING,* vol. 48, pp. 302-311, March 2001 2001.

[40] a. M. H. M. Reza Boostani, "Evaluation of the Forearm EMG Signal Features for the Control of a Prosthetic Hand," *Physiological Measurement,* vol. 24, pp. 309-319, 2003.

[41] L. Bueno, F. Brunetti, A. Frizera, and J. L. Pons, "Human - Robot Cognitive Interaction," in *Wearable Robotics: Biomechatronics Exoskeletons*, J. L.Pons, Ed.: WileyBlackwell, 2008, pp. 87-125.

[42] K. Englehart, B. Hudgins, P. A. Parker, and M. Stevenson, "Classification of the Myoelectric Signal using Time-Frequency Based Representations," *Medical Engineering and Physics on Intelligent Data Analysis in Electromyography and Electroneurography,* vol. 21, pp. 431-438, 1999.

[43] W. Cochran, J. Cooley, D. Favin, H. Helms, R. Kaenel, W. Lang, G. Maling, Jr. , D. Nelson, C. Rader, and P. Welch, "What is the Fast Fourier Transform?," *IEEE Transactions on Audio and Electroacoustics,* vol. 15, pp. 45-55, 1967.

[44] M. Misiti, Y. Misiti, G. Oppenheim, and H.-M. Poggi, "Wavelet Toolbox 4, User's Guide," The MathWorks, 2007.

[45] J. Kilby and H. G. Hosseini, "Wavelet Analysis of Surface Electromyography Signals," *Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE,* vol. 1, pp. 384-387, 2004.

[46] H. J. Hermens and T. A. M. van Bruggen, "Simulation of fatigue related changes in spectral parameters of surface EMG," in *Engineering in Medicine and Biology Society*, 1988, pp. 1706-1707.

[47] B. Hudgins, P. Parker, and R. N. Scott, "A new strategy for multifunction myoelectric control," *IEEE Transactions on Biomedical Engineering* vol. 40, pp. 82-94, Jan, 1993 1993.

[48] S. Ferguson and G. R. Dunlop, "Grasp Recognition from Myoelectric Signals," *Australasian Conference on Robotics and Automation,* pp. 83-87, 27-29 Nov 2002.

[49] C. M. Bishop, *Pattern Recognition and Machine Learning*: Springer Science + Business Media, LLC, 2006.

[50] F. H. Y. Chan, Y.-S. Yang, F. K. Lam, Y.-T. Zhang, and P. A. Parker, "Fuzzy EMG classification for prosthesis control," *IEEE Transaction on Rehabilitation Engineering,* vol. 8, pp. 305-311, 2000.

[51] W. Karwowski, A. Gaweda, W. S. Marras, K. Davis, J. M. Zurada, and D. Rodrick, "A fuzzy relational rule network modeling of electromyographical activity of trunk muscles in manual lifting based on trunk angles, moments, pel-

vic tilt and rotation angles," *International Journal of Industrial Ergonomics,* vol. 36, pp. 847-859, October 2006.

[52] D. R. Camargo, "Fuzzy logic towards pattern recognition of myoelectric signals for a hand prosthesis," in *Proceedings of the International Special Topic Conference on Information Technology in Biomedicine*, Ioannina, Greece, 2006.

[53] A. B. Ajiboye and R. F. Weir, "A heuristic fuzzy logic approach to EMG pattern recognition for multifunctional prosthesis control," *Neural Systems and Rehabilitation Engineering,* vol. 13, pp. 280-291, September 2005 2005.

[54] A. B. Ajiboye and R. F. Weir, "Fuzzy c-means clustering analysis of the EMG patterns of six major hand grasps," in *9th International Conference on Rehabilitation Robotics, ICORR,* 2005, pp. 49-52.

[55] H. Demuth, M. Beale, and M. Hagan, "Neural Network Toolbox™ 6 User's Guide," *The MathWorks, Accelerating the pace of engineering and science,* 2009.

[56] MathWorks, "Fuzzy LogicToolbox™ 2 User's Guide," *The MathWorks, Accelerating the pace of engineering and science,* 2009.

[57] Y. A. Arslan and Y. Hacioglu, "Modeling and Control of a Humanoid Robot Hand Finger," in *Proceedings of 5th International Symposium on Intelligent Manufacturing System,* 2006, pp. 1141-1149.

[58] M. Shinohara, M. L. Latash, and V. M. Zatsiorsky, "Age Effects on force produced by intrinsic and extrinsic hand muscles and finger interaction during MVC tasks," *Journal of Applied Physiology,* vol. 95, pp. 1361-1369, October 2003.

[59] F. o. E. Keyworth Institude, University of Leeds, "DuraForm PA plastic," 2008.

[60] F. o. E. Keyworth Institude, University of Leeds, "DuraForm GF plastic," 2008.

[61] K. S. Fu, *Robotics: Control, sensing, vision, and intelligence*: New York; London: McGraw-Hill, 1987.

[62] A. B. Soares, "Shape Memory Alloy Actuators for Upper Limb Prostheses." vol. Degree of Doctor of Philosophy: University of Edinburgh, 1997, pp. 1-194.

[63] H. M. B. Alcimar B.Soares, and David Gow, "The Application of SMA in the Design of Prosthetic Devices," *Proceedings of the Second International Conference on Shape Memory and Superelastic Technologies,* March 1997 1997.

[64] National_Instrument, "LabVIEW System Identification Toolkit User Manual," National Instrument, 2006.

[65] L. Lennart, "System Identification Toolbox 7-Getting Started Guide," The MathWorks, 2009.

[66] National_Instrument, "LabVIEW PID Control Toolkit User Manual," National Instrument, 2006.

[67] Michigan_College_of_Eingineering, "Control Tutorials for Matlab - PID Tutorial," 2007.

[68] Y. Huang, K. B. Englehart, B. Hudgins, and A. D. C. Chan, "A Gaussian Mixture Model Based Classification Scheme for Myoelectric Control of Powered Upper Limb Prostheses," *Biomedical Engineering, IEEE Transactions* vol. 52, pp. 1801-1811, Nov 2005.

[69] J. Kilby and H. G. Hosseini, "Extracting Effective Features of SEMG Using Continuous Wavelet Transform," *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE,* pp. 1704-1707, 30th Aug - 3rd Sept 2006 2006.

[70] M. S. H. M.B.I.Reaz, and F.Mogd-Yasin, "EMG Analysis Using Wavelet Functions to Detemine Muscle Contraction," *e-Health Networking, Applications and Services, 2006. HEALTHCOM 2006. 8th International Conference,* pp. 132-134, 17-19 Aug 2006.

[71] I. A. Basheer and M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application," *Journal of Microbiological Methods,* vol. 43, pp. 3-31, 2000.

[72] K. Swingler, *Applying Neural Networks: A Practical Guide*. Scotland, UK: Academic Press Inc., 1996.

[73] N. Guru, A. Dahiya, and N. Rajpal, "Decision Support System for Heart Disease Diagnosis using Neural Network," *Delhi Business Review,* vol. 8, pp. 99-101, January - June 2007.

[74] T. Masters, *Practical Neural Network Recipes in C++*. London: Academic Press, Inc., 1993.

[75] C. Xiang, Z. Xu, Z. Zhang-Yan, Y. Ji-Hai, V. Lantz, and W. Kong-Qiao, "Multiple Hand Gesture Recognition Based on Surface EMG Signal," in *Bioinformatics and Biomedical Engineering, ICBBE 2007,* Wuhan, 2007, pp. 506-509.

APPENDICES

APPENDIX A
DC Actuator System Identification Modelling and PID Control Design

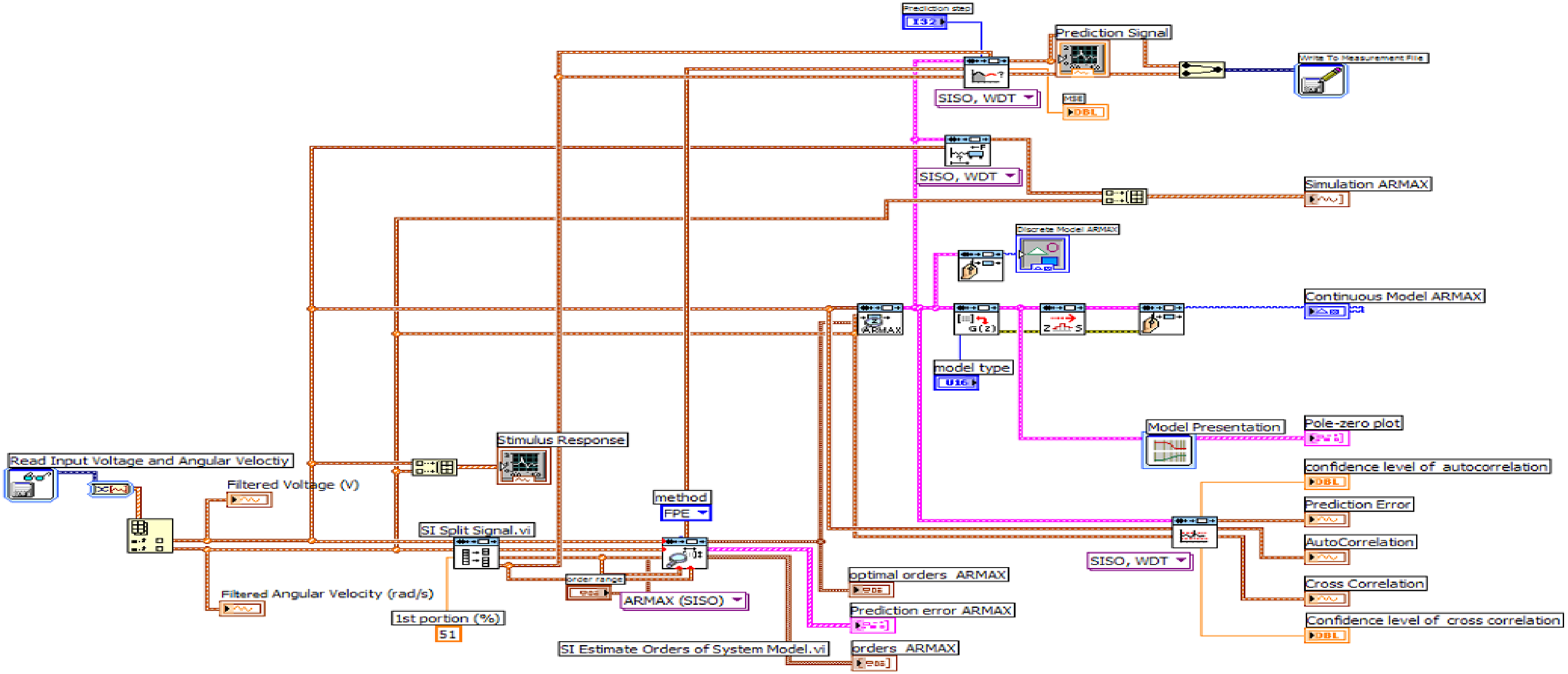
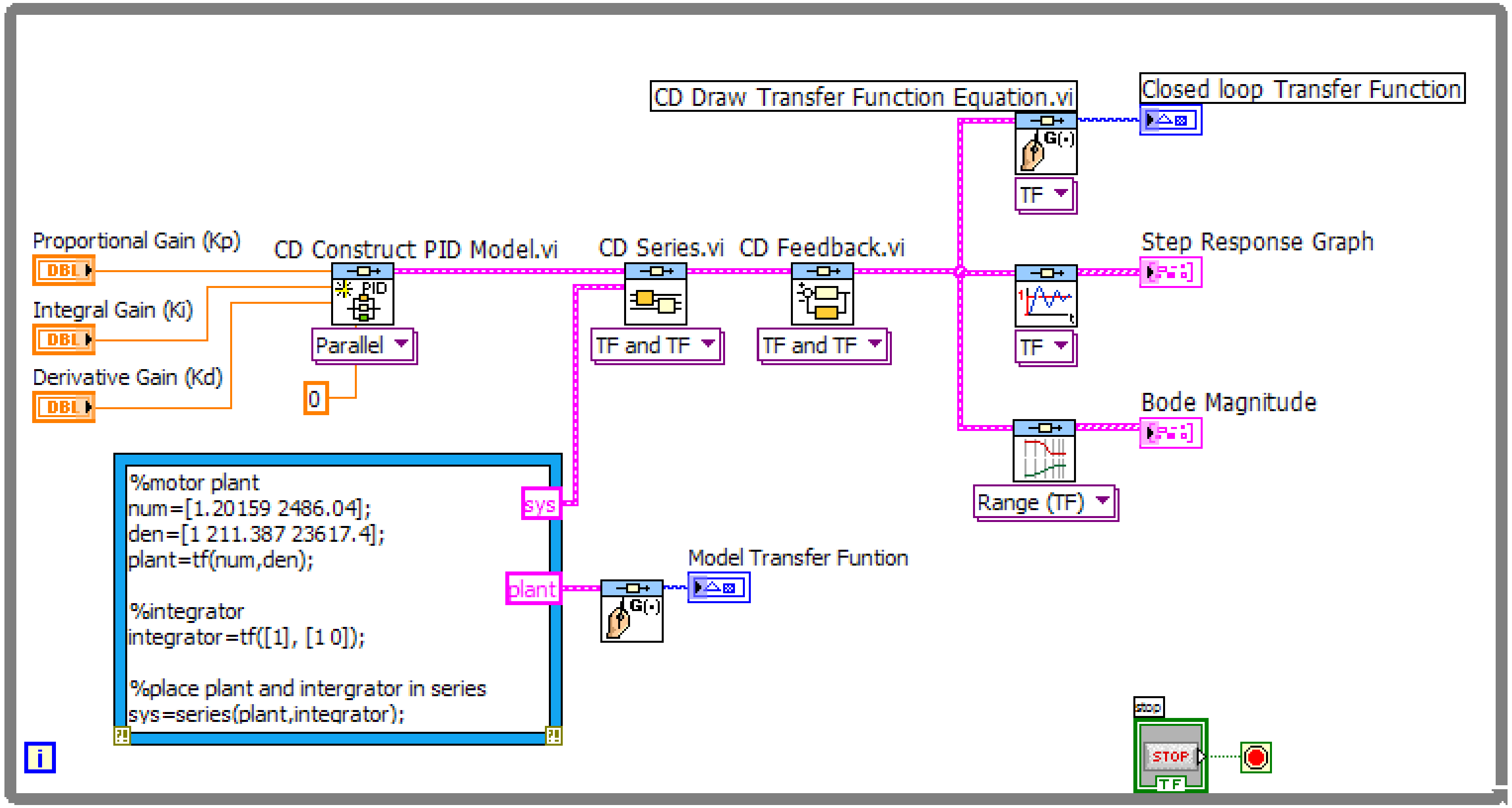


Figure A.1: Block diagram of modelling-system identification for DC actuator

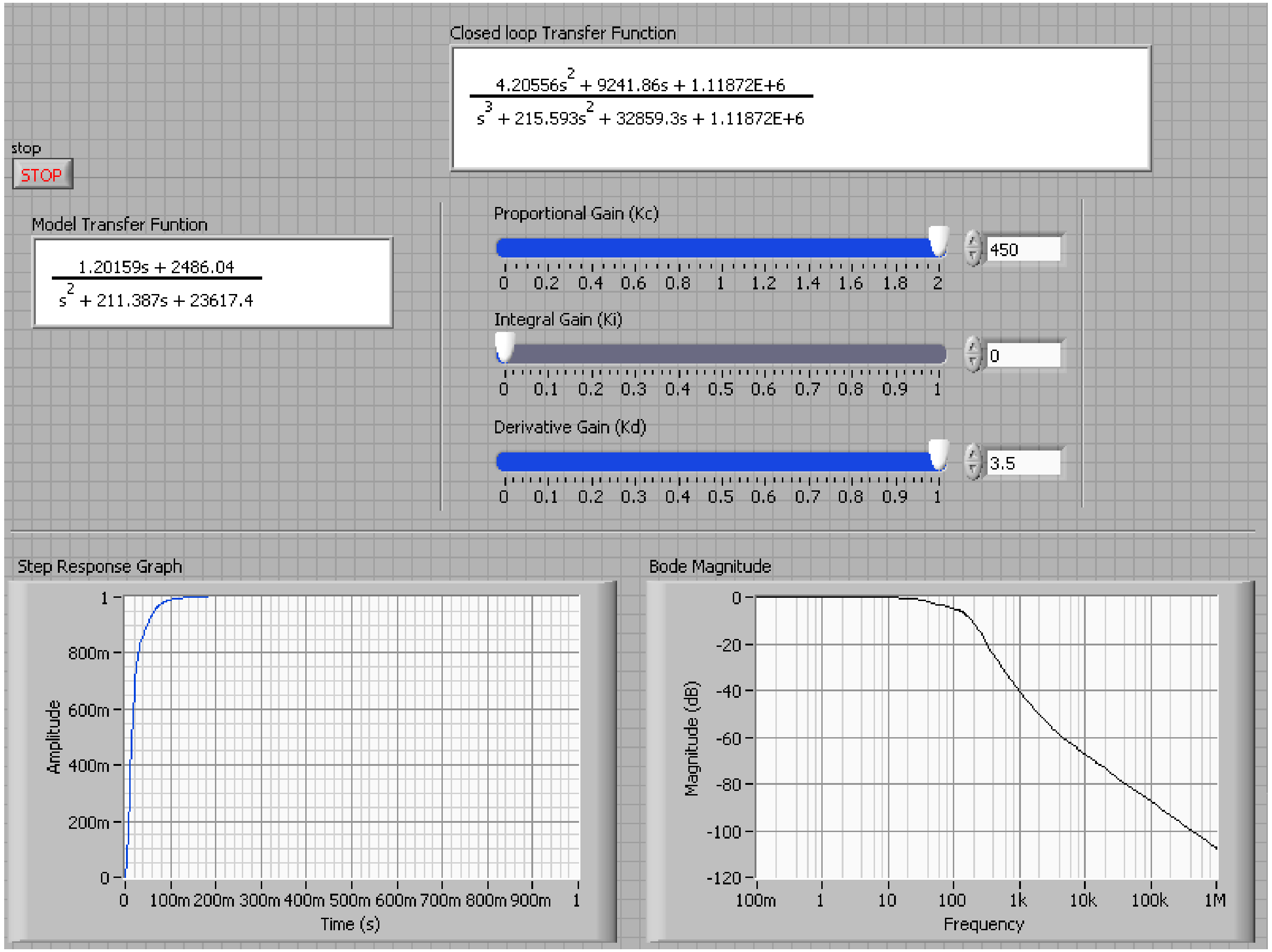


```
%motor plant
num=[1.20159 2486.04];
den=[1 211.387 23617.4];
plant=tf(num,den);

%integrator
integrator=tf([1], [1 0]);

%place plant and intergrator in series
sys=series(plant,integrator);
```

Figure A.2: Block diagram of PID control design for DC actuator



Closed loop Transfer Function

$$\frac{4.20556s^{2} + 9241.86s + 1.11872E+6}{s^{3} + 215.593s^{2} + 32859.3s + 1.11872E+6}$$

stop

STOP

Model Transfer Funtion

$$\frac{1.20159s + 2486.04}{s^{2} + 211.387s + 23617.4}$$

Proportional Gain (Kc) 450

Integral Gain (Ki) 0

Derivative Gain (Kd) 3.5

Figure A.3: Front panel of PID control design for DC actuator

APPENDIX B
Deployment Programming of DC Actuator Control System

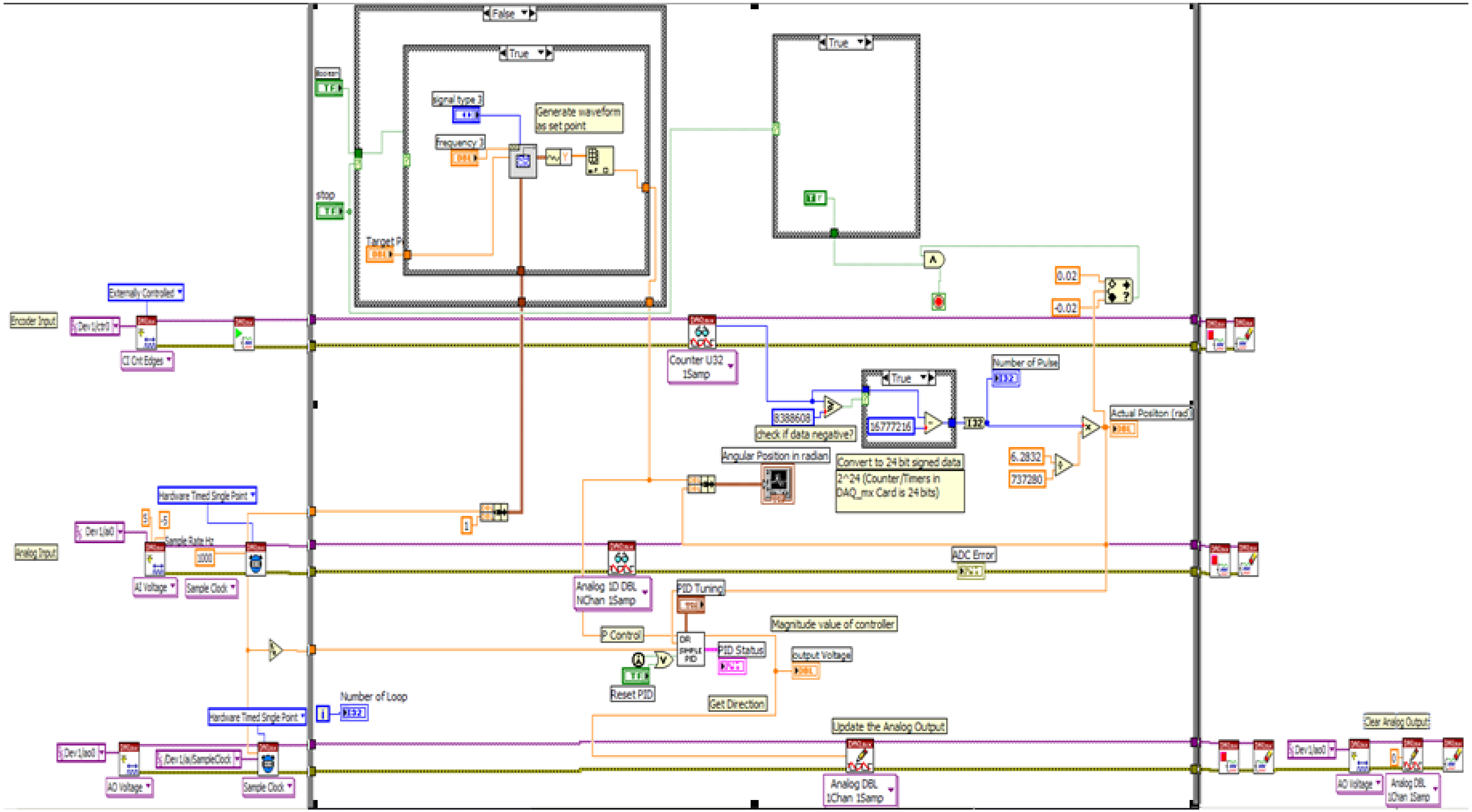
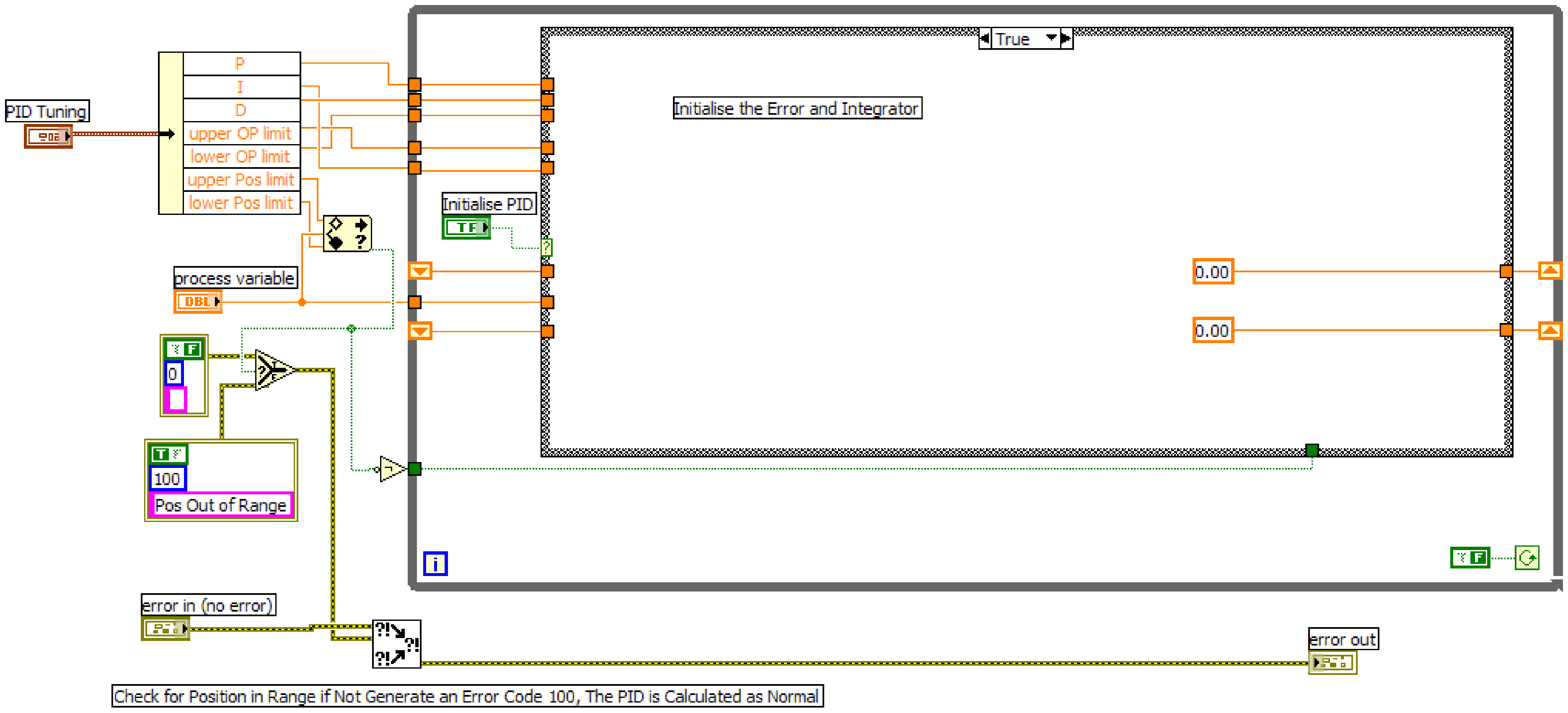


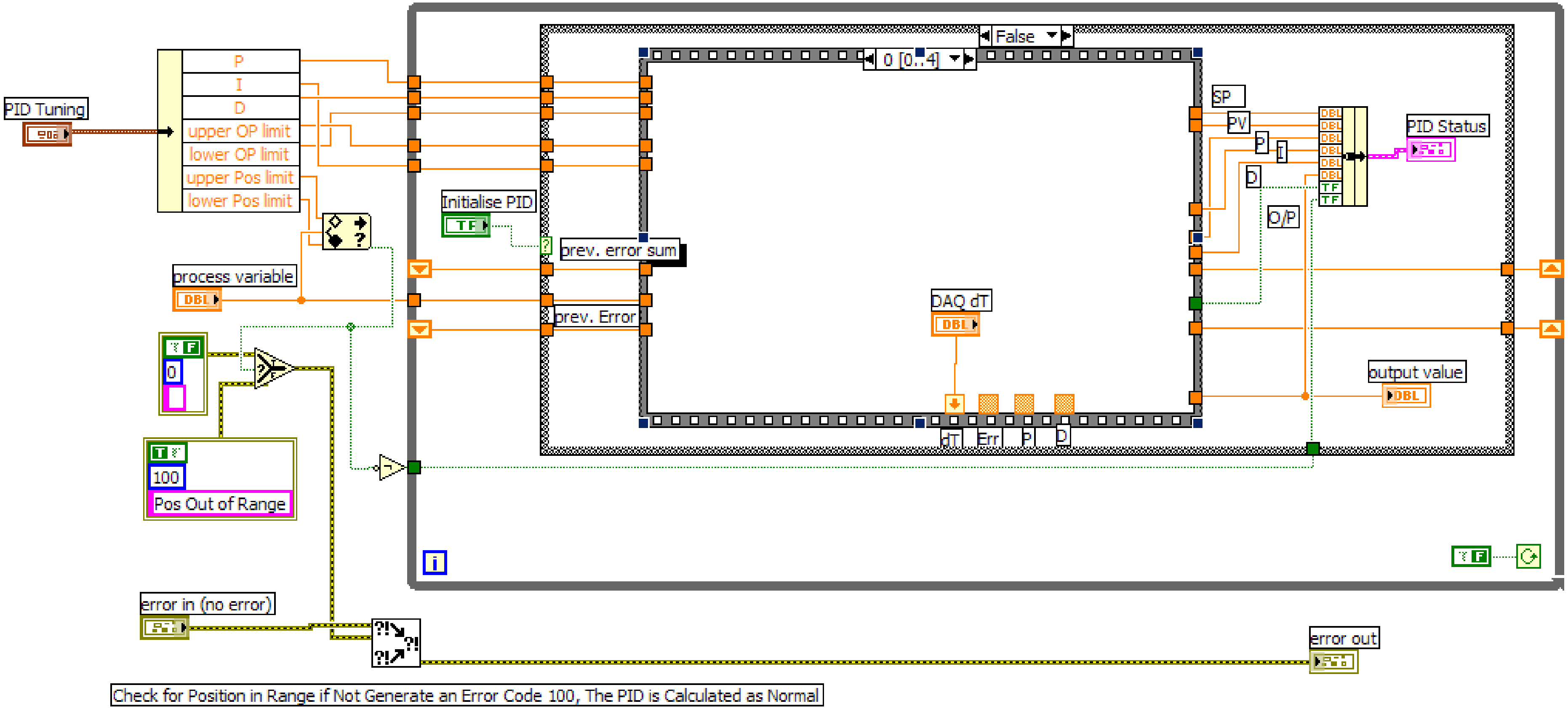
Figure B.1: Programming of DC actuator control system deployment



Initialise the Error and Integrator

Check for Position in Range if Not Generate an Error Code 100, The PID is Calculated as Normal

Figure B.2: PID Programming for DC actuator (Initialise the error and integrator)



Check for Position in Range if Not Generate an Error Code 100, The PID is Calculated as Normal

Figure B.3: PID Programming for DC (evaluate the value of dt)

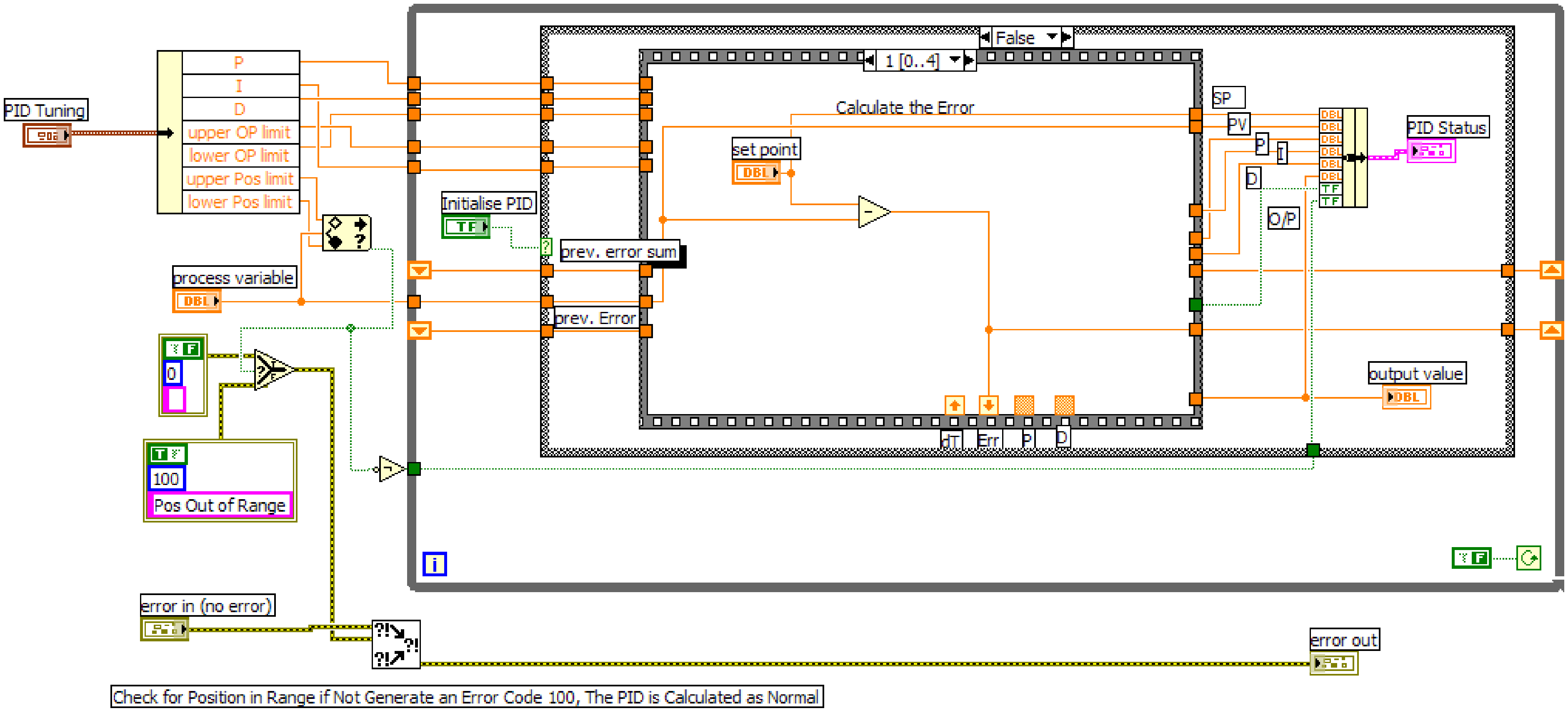


Figure B.4: PID Programming for DC (evaluate the value of error e)

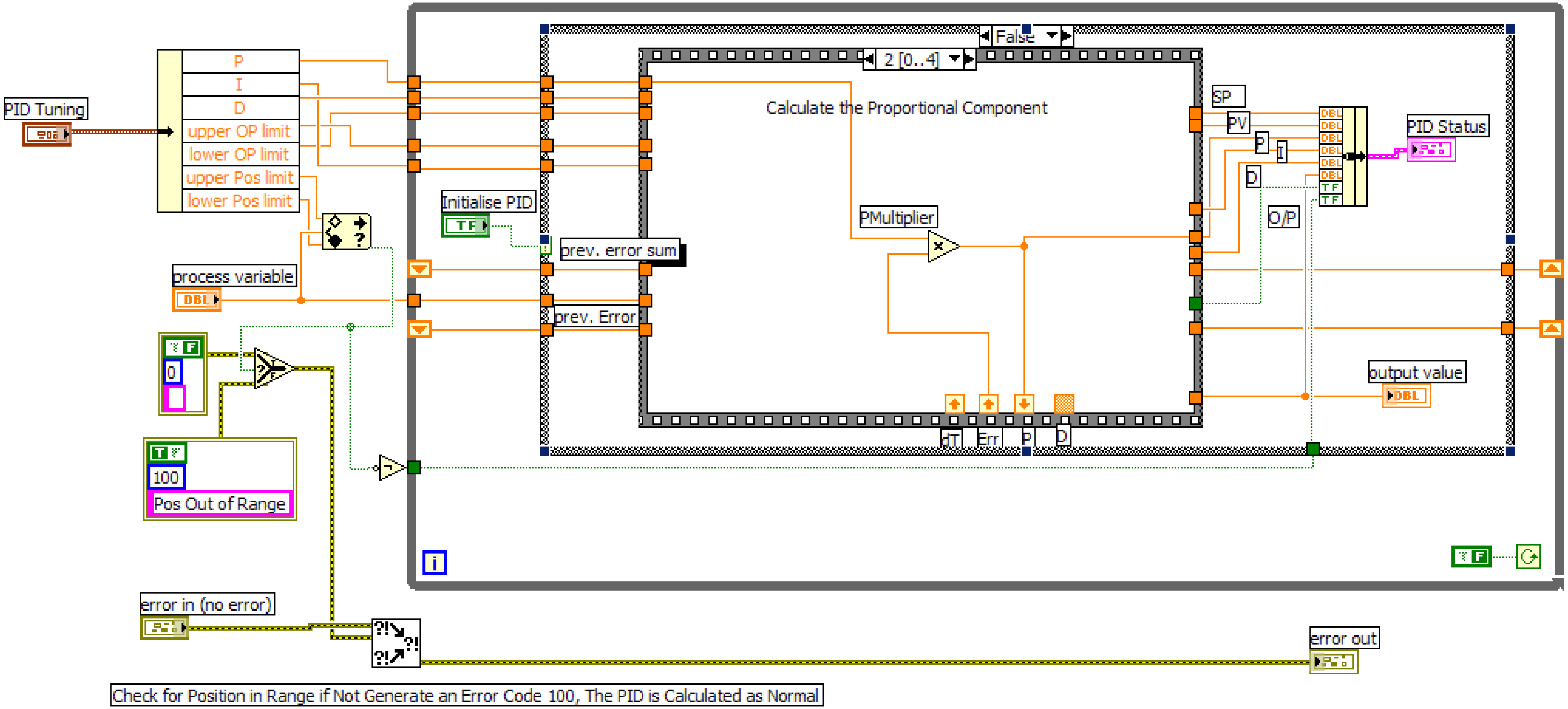


Figure B.5: PID Programming for DC (evaluate the value of proportional gain Kp)

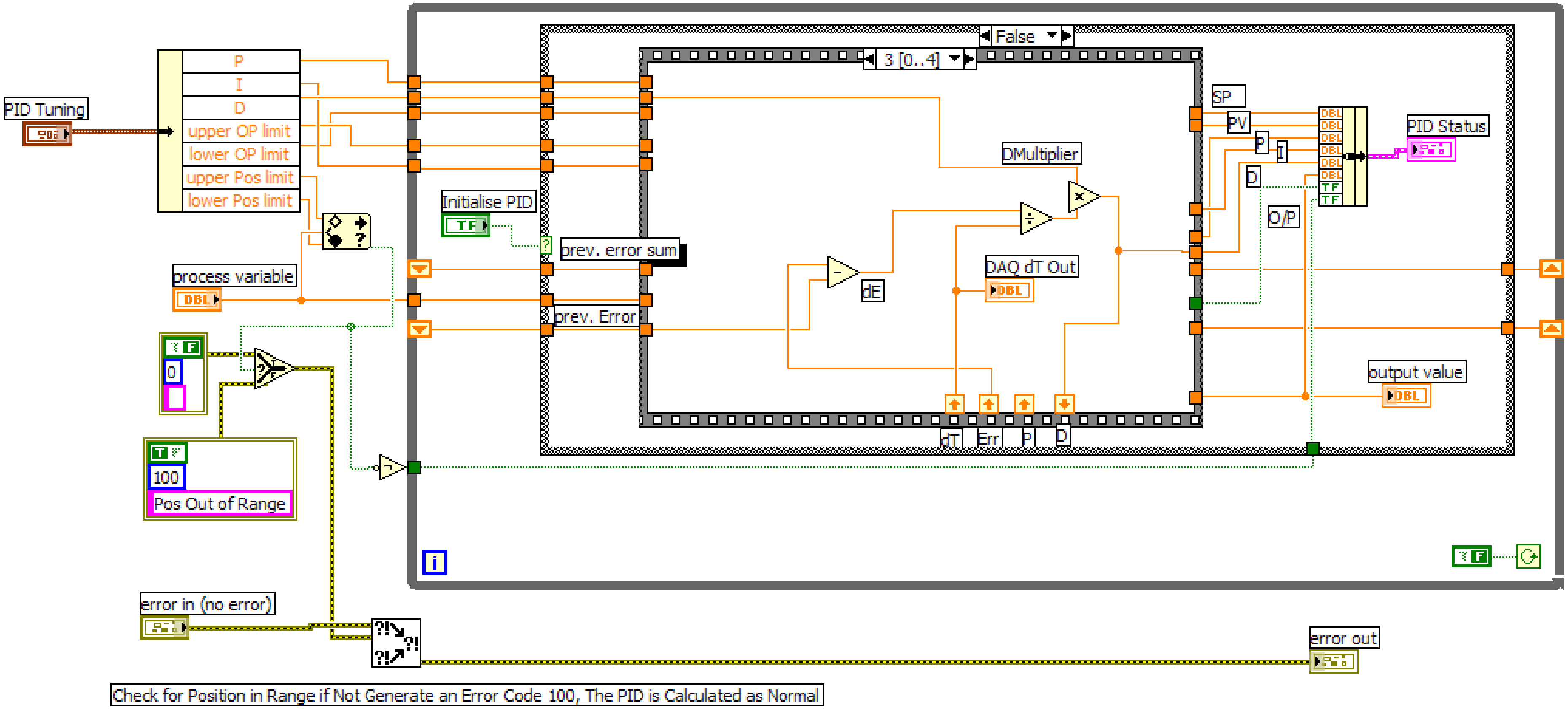
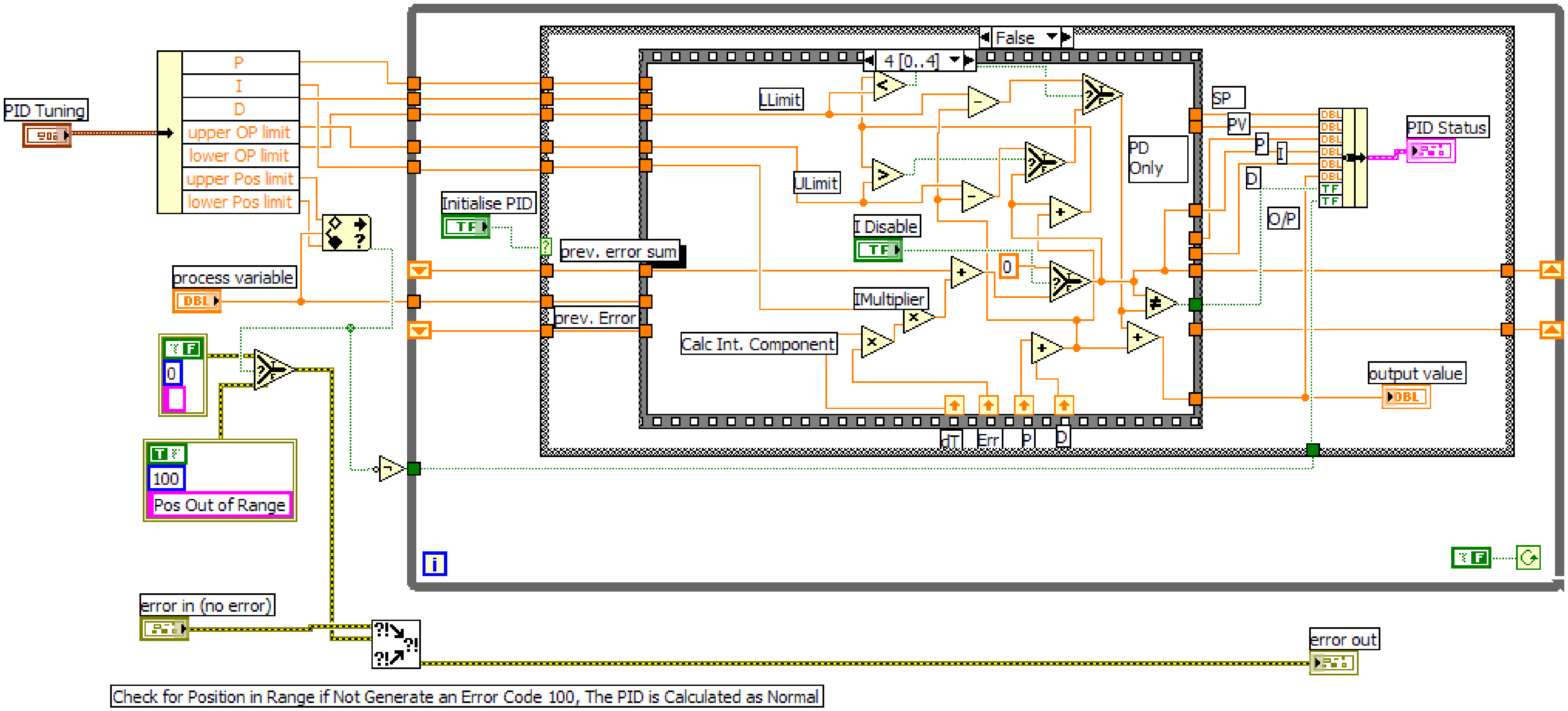


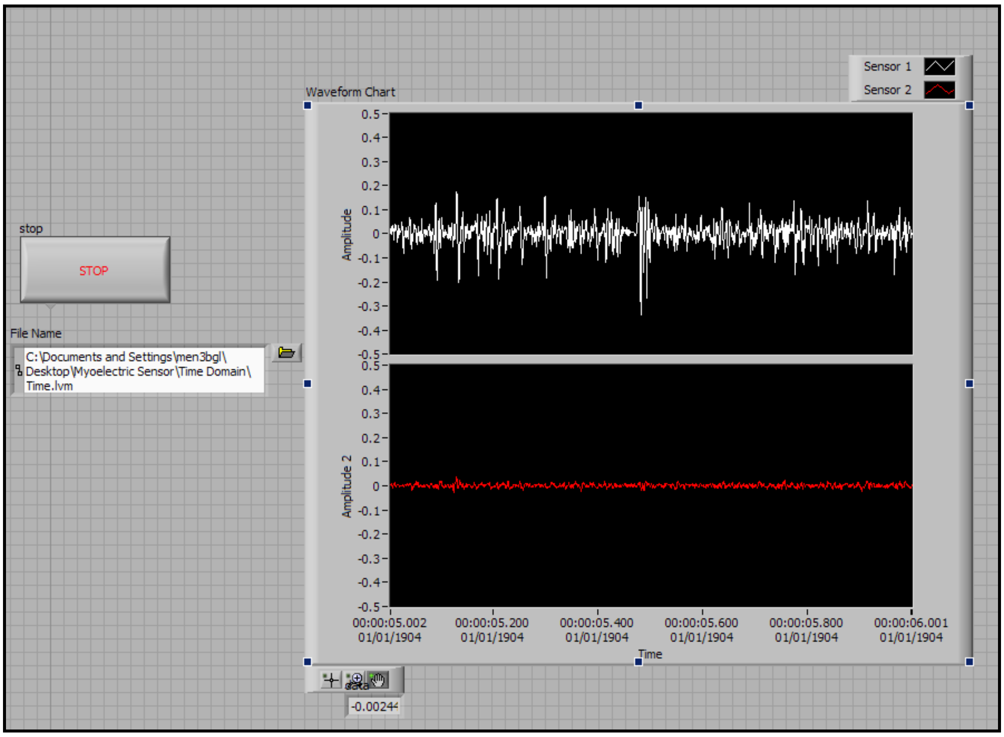
Figure B.6: PID Programming for DC (evaluate the value of derivative gain Kd)



Check for Position in Range if Not Generate an Error Code 100, The PID is Calculated as Normal

Figure B.7: PID Programming for DC (evaluate the value of integration gain Ki and PID output)

APPENDIX C
LabView Environment for sEMG Data Acquisition in Time Domain



File Name

C:\Documents and Settings\men3bgl\Desktop\Myoelectric Sensor\Time Domain\Time.lvm

Figure C.1: Front Panel

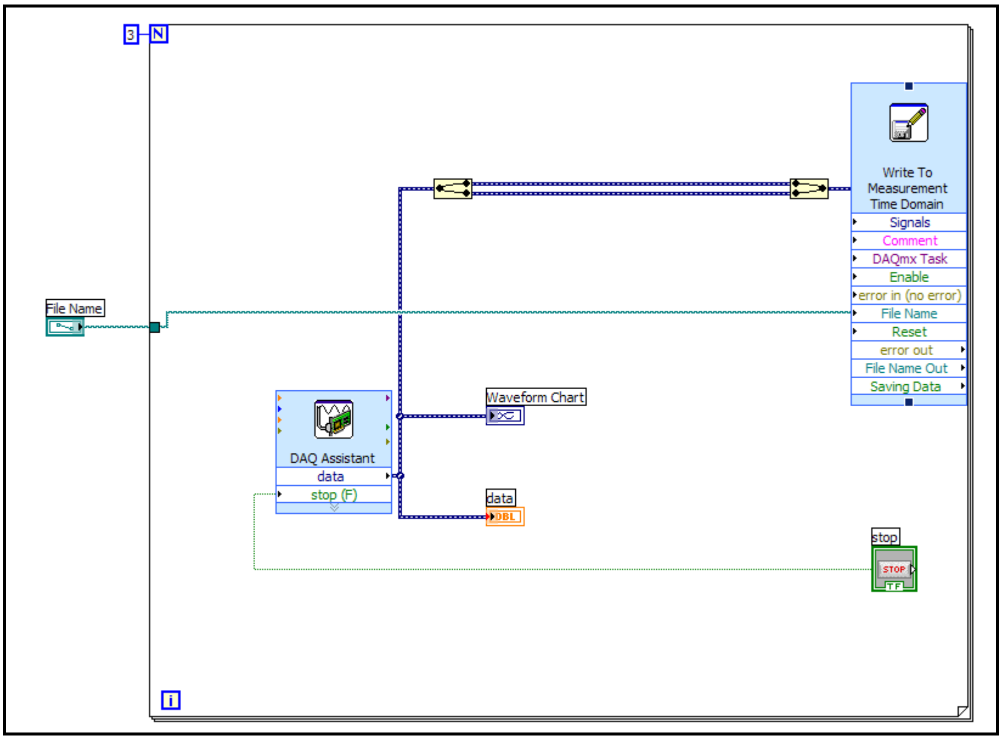


Figure C.2: Block Diagram

APPENDIX D
MatLab Programming for Features Extraction

This programming just show Pattern I, which is -45 supination turn down.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%first part of file name (Type of movement, should be got 9 types)
str11='D:\My Documents\MATLAB\Beng\Time_Domain18\Beng-45down';
%increment number of file name (The maximum value is 10 as 10 times per one movement)
str12 ='1';
str13 ='.xls'; %extension of file name

;%file name to save to
savename ='D:\My Documents\MATLAB\Beng\Features\10Trial18\S1-45down'
fileS1 = strcat(str11,str12,str13);%combining 3 parts of file name to a string
[ndata1, headertext,raw] = xlsread(fileS1);
t1=ndata1(:,1); %Time
s1=ndata1(:,2); %Data get on Sensor 1

%Construct a bandpass filter of a cut off frequency 20Hz-450Hz
%Sampling frequency is 1024Hz, and use the 5th order butterworth filter.
[b,a]=butter(5, [20 450]/1024);
emgfl=filter(b,a,s1);

%%%%% Continuous Wavelet Transform Scalogram (using db5,No of Scales 1-128)  %%%%%%

y1=cwt(emgfl,8,'db5');%Continuous Wavelet Transform using db5 in scales 8
y2=cwt(emgfl,16,'db5');%Continuous Wavelet Transform using db5 in scales 16
y3=cwt(emgfl,32,'db5');%Continuous Wavelet Transform using db5 in scales 32
y4=cwt(emgfl,64,'db5');%Continuous Wavelet Transform using db5 in scales 64
y5=cwt(emgfl,128,'db5');%Continuous Wavelet Transform using db5 in scales 128


%%%%%%%%%%%%%%%% Root Mean Square (RMS) %%%%%%%%%%%%%%%%%%%

RMS1=sqrt(sum(y1.*y1)/length(y1)); %Root Mean Square Value in scales 8
s1=['RMS at Scale=', num2str(RMS1)];
RMS2=sqrt(sum(y2.*y2)/length(y2)); %Root Mean Square Value in scales 16
s2=['RMS at Scale=', num2str(RMS2)];
RMS3=sqrt(sum(y3.*y3)/length(y3)); %Root Mean Square Value in scales 32
s3=['RMS at Scale=', num2str(RMS3)];
RMS4=sqrt(sum(y4.*y4)/length(y4)); %Root Mean Square Value in scales 64
s4=['RMS at Scale=', num2str(RMS4)];
RMS5=sqrt(sum(y5.*y5)/length(y5)); %Root Mean Square Value in scales 128
s5=['RMS at Scale=', num2str(RMS5)];

a=1024;%Sampling Rate

%%%%%%%%%%%%%%  Mean and Median Frequency  %%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%  Scales 8 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

FS1 = fft(y1,a);
PS1 = FS1.*conj(FS1)/a; %Power Spectrum
f1 = a*(0:(a/2))/a; %Frequency
```

```
m1=PS1(1:((a/2)+1));

m1Sq=m1.*m1;
m1Sum=sum(m1Sq);
topsum=0;
for i=1:((a/2)+1)
    top(i)=f1(i)*m1Sq(i);
end
for k=1:513
    topsum=top(k)+topsum;
end
mn1=topsum/m1Sum; %Mean Frequency
r=0.5*sum(m1);
for n=1:((a/2)+1)
    if sum(m1(1:n))>=r
        i=n;
        break
    end
end
med1=f1(i); %Median Frequency

%%%%%%%%%%%%%%%%%%%%  Scales 16  %%%%%%%%%%%%%%%%%%%%%%%%

FS2 = fft(y2,a);
PS2 = FS2.*conj(FS2)/a; %Power Spectrum
f2 = a*(0:(a/2))/a; %Frequency
m2=PS2(1:((a/2)+1));

m2Sq=m2.*m2;
m2Sum=sum(m2Sq);
topsum=0;
for i=1:((a/2)+1)
    top(i)=f2(i)*m2Sq(i);
end
for k=1:513
    topsum=top(k)+topsum;
end
mn2=topsum/m2Sum; %Mean Frequency
r=0.5*sum(m2);
for n=1:((a/2)+1)
    if sum(m2(1:n))>=r
        i=n;
        break
    end
end
med2=f2(i); %Median Frequency

%%%%%%%%%%%%%%%%%%%%  Scales 32  %%%%%%%%%%%%%%%%%%%%%%%%

FS3 = fft(y3,a);
PS3 = FS3.*conj(FS3)/a; %Power Spectrum
f3 = a*(0:(a/2))/a; %Frequency
m3=PS3(1:((a/2)+1));

m3Sq=m3.*m3;
m3Sum=sum(m3Sq);
topsum=0;
for i=1:((a/2)+1)
    top(i)=f3(i)*m3Sq(i);
end
```

```
for k=1:513
    topsum=top(k)+topsum;
end
mn3=topsum/m3Sum; %Mean Frequency
r=0.5*sum(m3);
for n=1:((a/2)+1)
    if sum(m3(1:n))>=r
        i=n;
        break
    end
end
med3=f3(i); %Median Frequency

%%%%%%%%%%%%%%%%%%%  Scales 64  %%%%%%%%%%%%%%%%%%%%%%%

FS4 = fft(y4,a);
PS4 = FS4.*conj(FS4)/a; %Power Spectrum
f4 = a*(0:(a/2))/a; %Frequency
m4=PS4(1:((a/2)+1));

m4Sq=m4.*m4;
m4Sum=sum(m4Sq);
topsum=0;
for i=1:((a/2)+1)
    top(i)=f4(i)*m4Sq(i);
end
for k=1:513
    topsum=top(k)+topsum;
end
mn4=topsum/m4Sum; %Mean Frequency
r=0.5*sum(m4);
for n=1:((a/2)+1)
    if sum(m4(1:n))>=r
        i=n;
        break
    end
end
med4=f4(i); %Median Frequency

%%%%%%%%%%%%%%%%%%%  Scales 128  %%%%%%%%%%%%%%%%%%%%%%%

FS5 = fft(y5,a);
PS5 = FS5.*conj(FS5)/a; %Power Spectrum
f5 = a*(0:(a/2))/a; %Frequency
m5=PS5(1:((a/2)+1));
m5Sq=m5.*m5;
m5Sum=sum(m5Sq);
topsum=0;
for i=1:((a/2)+1)
    top(i)=f5(i)*m5Sq(i);
end
for k=1:513
    topsum=top(k)+topsum;
end
mn5=topsum/m5Sum; %Mean Frequency
r=0.5*sum(m5);
for n=1:((a/2)+1)
    if sum(m5(1:n))>=r
        i=n;
        break
    end
end
med5=f5(i); %Median Frequency

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

b={med1,med2,med3,med4,med5,mn1,mn2,mn3,mn4,mn5,RMS1,RMS2,RMS3,RMS4,RMS5};
s=xlswrite(savename,b,'Signal Feature','A1');%The features store in A1 column of Excel

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

REPEAT THE WHOLE PROGRAMING FOR TEN TIMES for 10 trial of every movement, hence, change the str12='2…..10'

And change the s=xlswrite (savename, b, 'Signal Feature', 'A2…..A10').

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX E
MatLab Programming for Neural Network Pattern Recognition

This programming just show the three patterns, which are from patterns A to C, and just three data sets are in used.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 80% Trainning
% 10% Validating
% 10% Testing
% Transfer Function 'tansig' for Hidden Layer, 'logsig' for output Layer
% 'trainlm' used for Backpropagation network training function

%%%%%%%%%%%%%%%%%  Import data  %%%%%%%%%%%%%%%%%%%%%%

p = fopen('D:\My Documents\MATLAB\Beng\CSV_File\3S12Input123.csv');
M = textscan(p,'%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%s','delimiter',',');
fclose(p);

features = [M{1} M{2} M{3} M{4} M{5} M{6} M{7} M{8} M{9} M{10} M{11} M{12} M{13}
M{14} M{15}]; % inputs to neural network

A = strncmpi(M{16}, 'A', 1);
B = strncmpi(M{16}, 'B', 1);
C = strncmpi(M{16}, 'C', 1);

position = double([A B C]);% targets for neural network

% The neural network object in the toolbox expects the samples
% along columns and its features along rows.
% Our dataset has its samples along rows and its features along
% columns.Hence the matrices have to be transposed.

features = features';
position = position';

% The input data, |features|, is normalized to the range -1 and +1.
% The target data, |position|, is not normalized since it already
% has only |1| or |0| values.

[features,ps] = mapminmax(features);

% The variable |ps| contains the information neccessary to
% normalize new data in the same way the inputs have been
% normalized and also to un-normalize the normalized data if necessary.

%%%%%%%%%%%%% Building the neural network classifier %%%%%%%%%%%%%%
% The next step is to create a neural network that will learn to
% identify the movement position of left forearm wrist.

k=1;

% Number of Hidden Node increasable  from 1 to 70
```

```
for i=1:1:70

% Since the neural network starts with random initial weights, the results of this demo will differ
% slightly every time it is run. The random seed is set to avoid this randomness.

   rand('seed', 491218382)

% A one-hidden layer feed forward network is created with increasable hidden neurons in the
% hidden layer.

nout = size(position,1); %Searching for number of outputs

net=newff(minmax(features),[i 3],{'tansig' 'logsig'},'trainlm');
net.trainParam.show=10; %Each time run show 100 epochs in figure
net.trainParam.epochs=1000; %Maximum epochs show in figure
net.trainParam.goal=0.001; %train till the data hit the goal

% The dataset is now divided into training, validation and testing datasets. The training dataset is
% presented to the network for learning. Validation dataset is used to measure the training
% performance during training and stop training if necessary. The testing dataset is not used in any
% way during training and  hence provides an independent measure of training performance.

[trS, cvS, tstS] = dividevec(features, position, 0.1, 0.1);

% |trS| represents the training samples, |cvS| represents the
% validation samples and |tstS| represents the testing samples.
% Now the network is ready to be trained. It is trained with the
% training,validation and testing samples

net = train(net, trS.P, trS.T, [], [], cvS, tstS);

%%%%%%%%%%%%%%%%%% Testing the classifier  %%%%%%%%%%%%%%%%%%%
% The trained neural network can now be tested with the testing samples we partitioned from the
% main dataset using |dividevec|. The testing data is not used in training in any way and hence
% provides an "out-of-sample" dataset to test the network on. This will give us a sense of how well
% the network will do when tested with data from the real world.

out = sim(net, tstS.P); %Get response from trained network

% The network response can now be compared against the desired target response to build the
% classification matrix which will provides a comprehensive picture of a classifiers performance.

MeanSquareError(k)=mse(tstS.T-out); % Define the mean square error of testing samples

   [y_out,I_out] = max(out);
   [y_t,I_t] = max(tstS.T);

   diff = [I_t - 3*I_out];

   A_A = length(find(diff==-2));
   A_B = length(find(diff==-1));
   A_C = length(find(diff==0));
   B_A = length(find(diff==-5));
   B_B = length(find(diff==-4));
   B_C = length(find(diff==-3));
   C_A = length(find(diff==-8));
   C_B = length(find(diff==-7));
   C_C = length(find(diff==-6));

   N = size(tstS.P,2);% Number of testing samples
```

```
    fprintf('Total testing samples: %d\n', N);

cm = [A_A A_B A_C; B_A B_B B_C; C_A C_B C_C] %classification matrix

% The classification matrix provides a comprehensive picture of the classification performance of
% the classifier. The ideal classification matrix is the one in which the sum of the diagonal is equal
% to the % number of samples. It can also be understood in terms of percentages. The following
% matrix provides the same information as above but in terms of percentages.

cm_p = (cm ./ N) .* 100; %classification matrix in percentages

correct(k)=100*(cm(1,1)+cm(2,2)+cm(3,3))/N;
icorect(k)=100*(cm(1,2)+cm(1,3)+cm(2,1)+cm(2,3)+cm(3,1)+cm(3,2)) / N;

fprintf('Number of Node: %f%%\n', i);
fprintf('Percentage Correct classification: %f%%\n', 100*(cm(1,1)+cm(2,2)+cm(3,3))/N);
fprintf('Percentage Incorrect classification: %f%%\n',
100*(cm(1,2)+cm(1,3)+cm(2,1)+cm(2,3)+cm(3,1)+cm(3,2))/N);
    k=k+1;
end

correct=correct(:);
MeanSquareError=MeanSquareError(:); %define mean square error for testing in every hidden nodes
b=[correct MeanSquareError];
s=xlswrite('D:\My Documents \ MATLAB \ Beng \ Result \
3PatternsResultforBeng123.xls',b,'Sheet1','A1')

[Maxvalue Pos]=max(correct) %define maximum percentage in which hidden nodes

 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX F
Neural Network Pattern Recognition for Different Subjects

The results shown in this appendix is just for three patterns, which are patterns A to C. The data are contain seven different subjects, and it is taken on the year 2006.

The subject 1 is author, and the rest (subjects 2 to 6) of six are author's friend and colleagues. Every subject contain 60 rows (2 sensors x 10 trial for each movement position) and 15 column (features extracted).

The '1234567' in the Table F.1 is meant the all seven subjects' data are put together.

Table F.1: Correct pattern recognition rate in seven different subject for three patterns (pattern A to C).

Subject No.	**3 Patterns** **(%)**
1	100.00 at 17 number of hidden nodes
2	83.33 at 7 number of hidden nodes
3	83.33 at 18 number of hidden nodes
4	83.33 at 10 number of hidden nodes
5	83.33 at 7 number of hidden nodes
6	83.33 at 22 number of hidden nodes
7	83.33 at 47 number of hidden nodes
1234567	66.67 at 12 number of hidden nodes

APPENDIX G
Published Work

Design of a Prosthetic Finger with Hybrid Actuation

BengGuey Lau[1], Abbas Dehghani[2], Robert. C. Richardson[3] and David Gow[4]

School of Mechanical Engineering, University of Leeds, UK, LS2 9JT[1,2,3]
Smart Centre, NHS Lothian[4]
men3bgl@leeds.ac.uk[1]
A.A.Dehghani-Sanij@leeds.ac.uk[2]
R.C.Richardson@leeds.ac.uk[3]
David.gow@nhslothian.scot.nhs.uk[4]

ABSTRACT

The purpose of a prosthetic hand/finger is to imitate the grasping capabilities of the human hand/finger, with obvious improvements in the quality of life of amputees. This paper presents the mechanical design of a 3-DOF prosthetic middle finger, which closely resembles in physical characteristic and functionality that of a male adult. The actuation mechanism used is a hybrid system consisting of a DC actuator and a Shape Memory Alloy (SMA) actuator. The DC actuator is placed in the palm and it controls the flexion-extension movement of the Metacarpophalangeal (MCP) joint. The SMA actuator is also placed in the palm but at a different level from that of the DC actuator, and it is attached to tendon wires, which are similar to the natural tendons and muscles. The tendon wire in turn, is attached to the upper side (grasping side) of the Middle Phalanx in order to control the Proximal Interphalangeal (PIP) joint and Distal Interphalangeal (DIP) joint flexion movement. A spring is connected to the bottom side of the Middle Phalanx to control extension movement. The movement of the DIP joint is coupled with the PIP joint through a pulley. The static, and kinematic analysis of the system, presented in this paper, shows that our design of the prosthetic finger closely resembles the functionality of a human middle finger.

Keywords: Prosthetic Finger, Hybrid Actuation System, Shape Memory Alloy, Mechanical Design, Forward Kinematics

1. Introduction

The main aim of an artificial limb is to re-establish the lost functionality of the amputated organic limb. A prosthesis as a tool makes no pretence of trying to replace the lost limb physiologically but it works as an aid to help provide some of the functions that were lost. The prosthesis is an interchangeable device that is worn and used as needed. Much effort in the field of upper-extremity prosthesis research is directed towards the creation of prostheses as true limb replacements; however, in current practice we are mostly limited to prostheses as tools.

In the past, electric motors were the primary actuators used in prosthetic devices as they have a high torques and are easy to control. However, their heavy weight, sizes, and shapes have been restrictive and uncomfortable to the users [1, 2]. Typically electric motors are not small enough and can only be placed remotely, like ahead of the wrist joint of the hand [2]. This positioning increases the complexity of the mechanical designs, since all the tendons driving the fingers have to pass through the wrist joint. Due to these constraints the design and development of most of the current commercial prosthetic hands have failed to provide enough grasping functionality and have few active degrees of freedom restricting their effectiveness [3]. Several robotic and anthropomorphic hands may have sufficient active degrees of freedoms, such as the Shadow Hand with up to 20 DOF [4], allowing them to have dexterity comparable to that of the human hand, unfortunately though these type of hands cannot be used as prostheses due to their physical characteristic (pneumatic actuator) that pose several serious limitations on human-hand interaction [5].

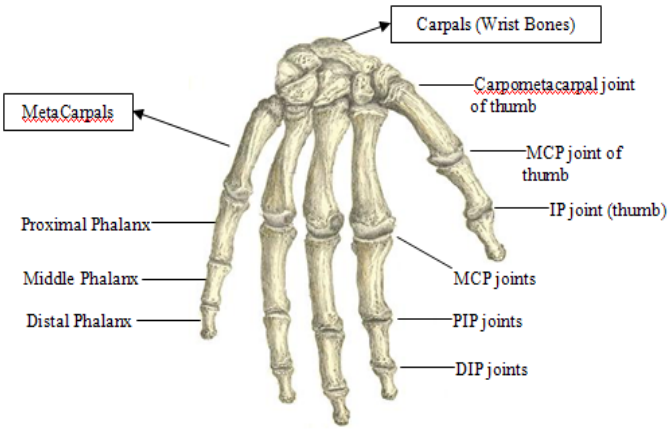
Due to these limitations of the previous electric motors and of the pneumatic actuators, some researchers have used Shape Memory Alloys (SMAs), or Electroactive Polymers (EAPs) to replace the electric motor and the pneumatic actuator as they are operationally similar to the human muscles [6]. Both of these are lightweight and direct-driven actuators, that can provide ways to increase the functionality of artificial hands without adding mechanical complexity [6]. EAP materials have higher response speeds and lower densities when compared to SMAs [7]. However, they have low actuation forces, mechanical energy density and lack of robustness compared to the SMAs [7]. Besides, the big advantages of shape memory alloys are their incredible small size and volume, their high force to weight ratio and their low cost [1, 8]. One drawback with SMAs is that in prosthetic devices large lengths of SMAs are needed, something that limits their life cycle [1].

The main goal of this paper is to report the design of a novel biomimetic middle finger for a male adult, which is to be actuated using an electric DC motor and an SMA actuator.

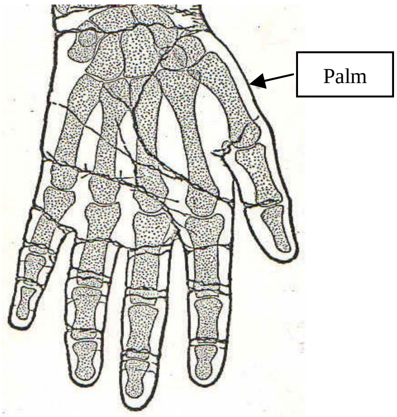
2. Hand and Finger Anatomy

The human hand consists of a broad palm (metacarpal and Carpals) with total five digits attached to the forearm by a wrist joint. These five digits are thumb and four fingers (index finger, middle finger, ring finger and little finger). The thumb is located on one of the sides of the palm and connected to the carpals bone. Each finger consist of three bones, which are the Proximal phalanx, the Middle phalanx, and the Distal Phalanx, and three joints**,** which are the Metacarpophalangeal (MCP) Joint, the Proximal Interphalangeal (PIP) Joint and the Distal Interphalangeal (DIP) Joint. The bones and the joints of the hand are illustrated in Figure 1.

The MCP joint have 2 DOF, these being abduction-adduction and flexion-extension. The PIP and DIP joints have 1 DOF, that is flexion-extension. The abduction-adduction angles of MCP joint is around 20^0, and flexion-extension is 90^0. The flexion-extension movement of the PIP and DIP joints are $100\sim110^0$ and $80\sim90^0$ respectively [3, 5, 6, 10].



(a)



(b)

Figure 1: (a) Anatomy of Human Hand [10], (b) Bones and Joint with Human Hand.

3. Finger Design and its Rapid Prototyping

3.1 Finger Dimensions

The dimensions of the design in Figure 2 and Table 1 are based on the size of the main author’s middle finger and it is custom made due to the result of large variations in the sizes of human hands and fingers.

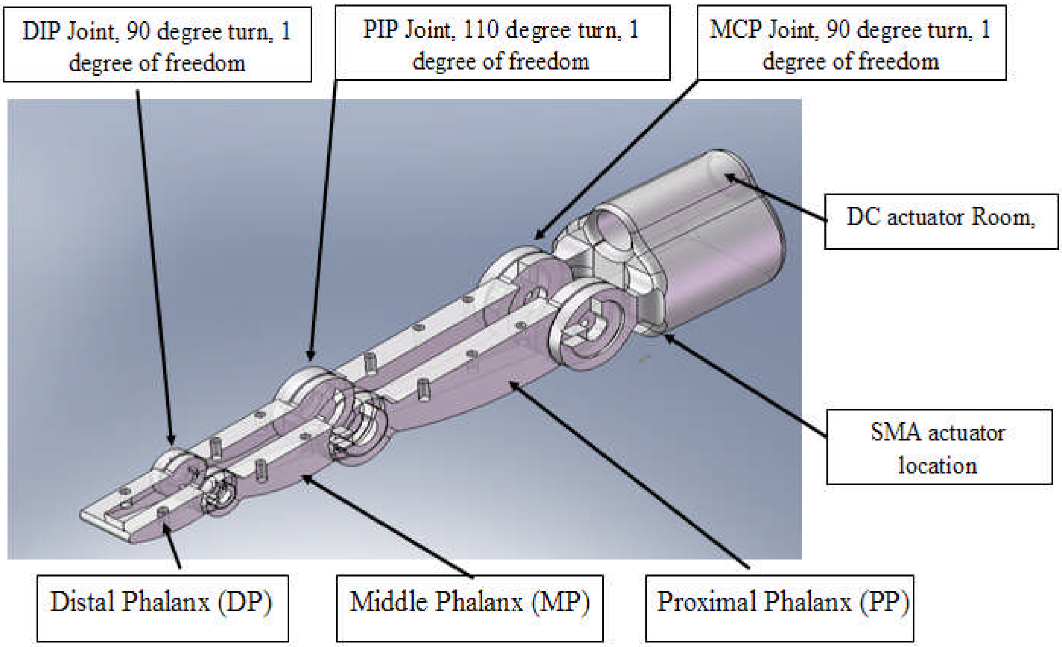

DIP Joint, 90 degree turn, 1 degree of freedom
PIP Joint, 110 degree turn, 1 degree of freedom
MCP Joint, 90 degree turn, 1 degree of freedom
DC actuator Room,
SMA actuator location
Distal Phalanx (DP)
Middle Phalanx (MP)
Proximal Phalanx (PP)

Figure 2: Prosthetic middle finger design.

Table 1: Dimensions of Finger Phalanges.

Lengths (mm)			**Widths (mm)**			**Depths$_{max}$ (mm)**		
PP	MP	DP	PP	MP	DP	PP	MP	DP
58.59	36.03	23.93	17.00	15.00	13.00	15.35	9.43	6.04

The abduction-adduction movement of MCP joint is just 20^0, and does not contribute significantly to its functionality. For this reason, it is neglected leaving the design of our finger with 3 DOF, these being 1 DOF at the MCP joint, 1 DOF at the PIP joint, and 1 DOF at the DIP joint.

Total three resolute joints are used connected in each joint or each DOF, as shown in Figure 3. Each phalanx consists of one head and one tail. The fixed axes (with a stopper) of rotations were called the head. The head of palm is connected to the tail of the proximal phalanx, for the MCP joint, which is corresponding to the movement of flexion-extension: 90^0; the head of the proximal phalanx is connected to the tail of the middle phalanx, for the PIP joint, which is corresponding to 110^0 of flexion-extension; and the head of the middle phalanx is connected to the tail of the distal phalanx, for the DIP joint, which corresponds to 80^0 of flexion-extension.

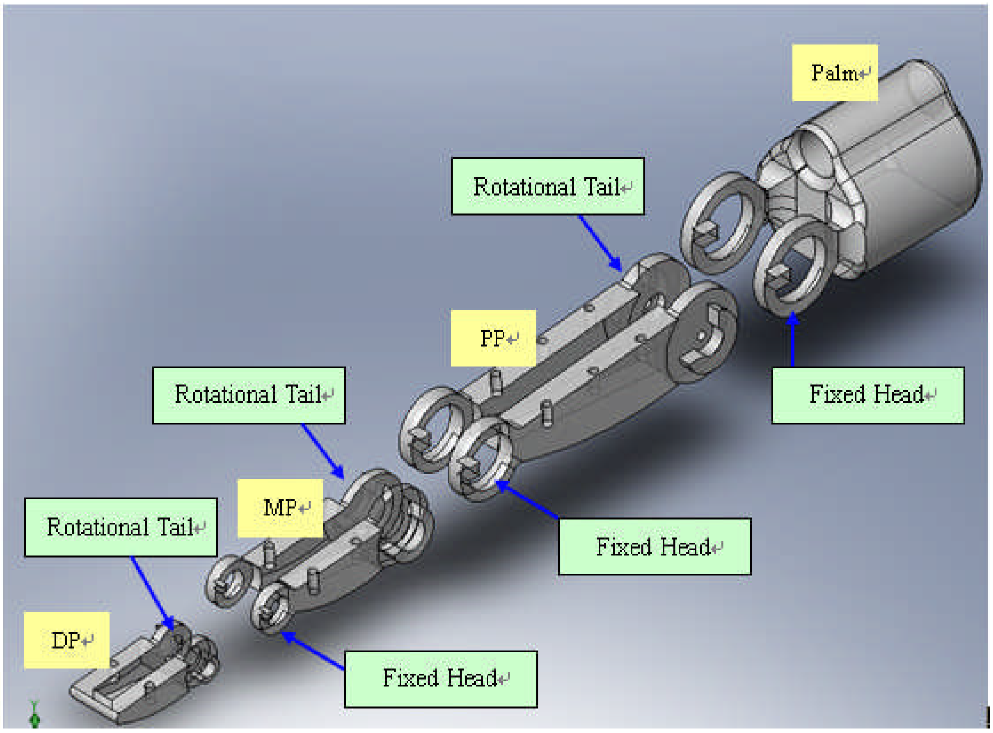


Figure 3: Kinematics architecture of Finger.

3.2 Material Selection for Rapid Prototyping

The finger, shown in Figure 2 was manufactured using a Vanguard HS HiQ SLS (Selective Laser Sintering) rapid prototype machine. DuraForm GF plastic (Glass filled Nylon based on Nylon 12) was used as it has a high enough tensile module (4068MPa), and it is light (1.49g/cm^3). Using this material and the relevant calculations (Equation 1), the object length change (ΔL) is 8.23 x 10^{-5} m (0.0823mm), which suggests that the material is strong enough. The total mass without any mechanism is 53g.

$$E = \frac{Stress}{Strain} = \frac{\sigma}{\varepsilon} = \frac{\frac{F}{A_0}}{\frac{\Delta L}{L_0}} = \frac{FL_0}{A_0 \Delta L} \text{ ------- (Equation 1)}$$

Where: E = Young’s (Tensile) Module,

F = Force applied to the object,

A_0 = Original cross-section area through which the force applied,

ΔL = Amount by which the length of the object change,

L_0 = The original length of the object

Nylon was preferred than aluminium because it has a lower friction coefficient, 0.15-0.25 and 1.05-1.34 respectively. An additional implication would be the use of bearings needed on all aluminium based design, as it would have increased the overall complexity of the system without any benefits.

4. Hybrid Actuator Design for Finger

For a prosthetic finger to be effective , it should allow holding and grasping of objects, as these two are important functions of a human hand. The joints and phalanx torques needed to maintain grasp are presented in this section.

4.1 DC Actuator Design for MCP Joint

The movement of the flexion-extension in the MCP joint is controlled by a DC actuator because the proximal phalanx of the middle finger in one finger tasks can create a higher force than the distal phalanx which is about 40N in proximal phalanx [9].

Maxon DC motor RE10 was used in our design. This motor is extremely light weight and small in size, just around 20g (including gear head), 10mm diameter and 57mm total length (including gear and encoder). It is placed in the palm to control the flexion-extension of MCP joint, as this joint needs more force/power [9] than the rest.

$$\tau_{MCPjo\,\mathrm{int}} = \tau_{DCactuator} \times GearRatio \text{ ------- (Equation 2)}$$

Where: GearRatio= External Gear Ratio,

$\tau_{DCactuator}$ = Torque of DC actuator,

$\tau_{MCPjo\,\mathrm{int}}$ = Torque of MCP joint.

A proper gear to convert the rotation motion of the DC actuator to flexion-extension motion of the finger and to increase the torque of the joint is needed. A higher gear ratio of worm gear is used instead of a bevel gear. This is because the gear ratio is directly proportional to the torque of the MCP joint, as illustrate by Equation 2. The maximum transmission torque in the MCP joint is 4.26Nm by using a 45:1 worm gear.

The design shown in Figure 2 was computed based on using a worm gear. The 'room' for the placement of the DC is located on the Metacarpal bone (palm) (see Figure 1).

4.2 SMAs Actuator Design for PIP and DIP Joint – Conceptual design

The SMAs actuator design is the most challenging part in order to produce the smallest size possible so that to achieve large linear stroke or displacement for large angular motion. We propose a mechanism associated with SMAs wires in order to create large linear displacement through small change of SMAs wires.

An SMA actuator and a spring are used to control the flexion-extension of PIP joint. The SMA actuator is constructed together with the miga motor. It can have a 10mm linear stroke and 6N output force when 200 μ m diameter SMA wire is used. The dimension of the actuator assembly is small enough (56mm x 17mm x 3mm) to be located in the palm. The output force can be increasable by changing the location of the spring connected on the output lever (Figure 4) and changing the size of SMA wires used. The length of the output lever is 30mm (Figure 4). The flexion-extension of DIP joint is coupled to PIP joint. Hence, the proposed finger has two actives DOFs, and one passive DOF. The abduction-adduction of this middle finger is not considered, it is because compared to the human hand, the movement range is very small and just 20^0 of middle finger. Hence, it is not important or not necessary for the middle finger.

When the SMA wires heat up, they contract and pull the output lever to the opposite direction to that of the SMA wire contraction, due to the pivot used, as shown in Figure 4. This achievement pulls the tendon wire which is connected on one side to the 'Place to connect tendon cable' (Figure 4) and on the other side is connected to the middle phalange of finger to allow the PIP joint to move in an angular motion.

There are two reasons for the use of the spring attached on the SMA actuator (see Figure 4). Firstly, it can reduce the recovery time in order the finger can be released faster, and secondly, it increases the output lever force in order to increase the grasping force. The output lever force is based on where the spring is connected to the output lever, as shown in the Figure 4.

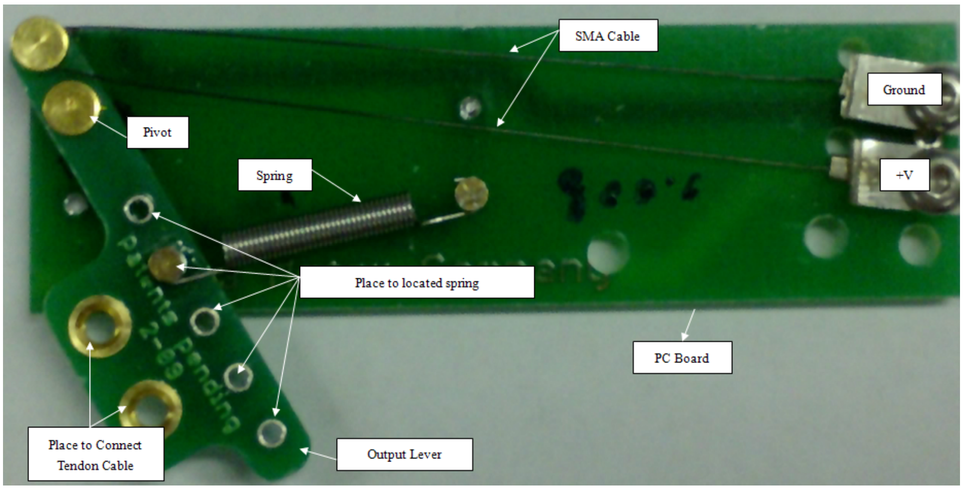


Figure 4: SMA Actuator .

The recovery (resting) position of the SMA actuator's output lever is shown in Figure 4. When the SMA cable fully contracts, the output lever will move to the left and get aligned with the edge of the PC Board of the SMA actuator (see Figure 4). The stroke of this motion is around 10mm.

In static analysis, the torque at PIP and DIP joints are given by Equation 3 and 4 respectively:

$$\tau_{PIPjo\,\mathrm{int}} = F_{SMAactuator} \times a - F_{Oppo\,\mathrm{sin}\,gSpring} \times a_{Spring} \text{ ------- (Equation 3)}$$

Where: $F_{SMAactuator}$ = SMA actuator force.

a = Middle Phalange cross-sectional radius at tendon location

$F_{Oppo\,\mathrm{sin}\,gSpring}$ = Opposite Spring Force.

a_{Spring} = Middle Phalange cross-sectional radius at spring location.

$\tau_{PIPjo\,\mathrm{int}}$ = Torque of PIP joint.

$$\tau_{DIPjo\,\mathrm{int}} = \tau_{PIPjo\,\mathrm{int}} \times \frac{d_{PIPpulley}}{d_{DIPpulley}} \text{ ------- (Equation 4)}$$

Where: $\tau_{PIPjo\,\mathrm{int}}$ = Torque of PIP joint

$d_{PIPpulley}$ = Diameter of PIP joint's pulley

$d_{DIPpulley}$ = Diameter of DIP joint's pulley

When the PIP joint moves, the DIP joint will also move, because the DIP joint is coupled to the PIP joint through a set of 1.375 ($\frac{d_{PIPpulley}}{d_{DIPpulley}} = \frac{11mm}{8mm}$) ratio pulley. The main advantage of this arrangement is the reduction of the number of actuators used. Moreover, the torque of the DIP joint is higher. From Equations 3 and 4, the torque at the PIP joint is 0.022Nm, and at the DIP joint is 0.03Nm. These two torques are adjustable depending on where the spring connects to the output lever (see Figure 4) and on the different sizes of the SMA wires.

5. Finger Kinematics

The purpose of the finger kinematics analysis is to determine the relationship between the angular position of each joint. Our finger design was modelled as a robot hand fixed to the palm, as shown in Figure 5.

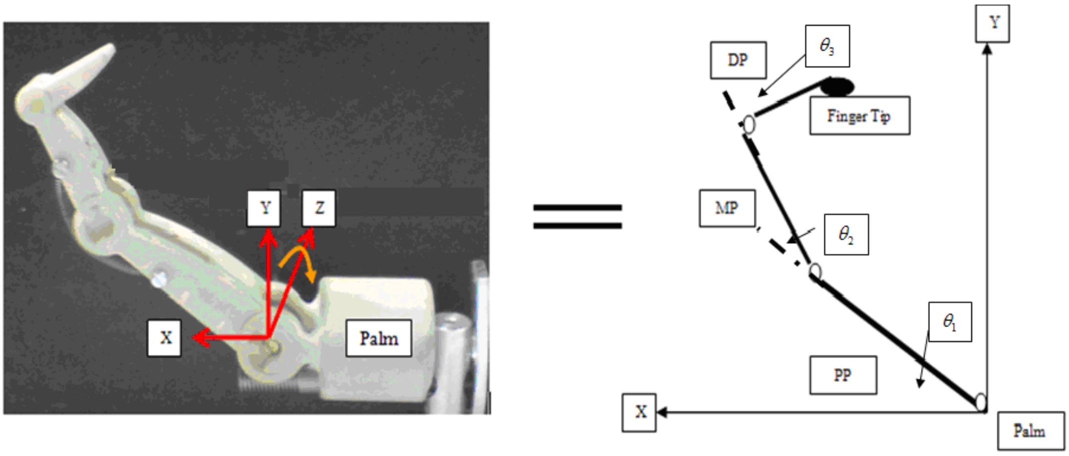


Figure 5: Schematic of a robot hand lying in the X-Y plane based on a prosthetic finger.

Denavit-Hartenberg's (D-H) is the most commonly used method to solve the forward kinematics problem where the transformation of links connected in series are used to produce a single transformation that shows the position or direction of the fingertip with respect to the palm or to the reference base [6, 11]. The finger kinematics can be considered in 2D kinematics and lying in the X-Y plane, as shown in Figure 5, because the abduction-adduction movement of the MCP joint was neglected as previously explained.

The Denavit-Hartenberg (D-H) theorem calculates the position or direction of the fingertip and consists of four variables, these being the joint angle θ_i , the link/phalanx offset d_i , the link/phalanx length l_i, and the link/phalanx twist α_i [11]. Because the three separate finger joints are using resolute/hint joints and they are connected at the same level along the x-axis, the offset, d_i, and the twist, α_i, are equal to zero. The finger link/phalanx coordinate parameters are presented in Table 2.

Table 2: Finger link/phalanx coordinate parameters.

Joint *i*	α_i	θ_i	l_i	d_i	Joint Range
1	0	θ_1	58.59 mm	0	0-90^0
2	0	θ_2	36.03 mm	0	0-110^0
3	0	θ_3	23.93 mm	0	0-80^0

Since the values of the link/phalanx offset (d_i) and of the twist (α_i) are zero, the position of the fingertip is given by:

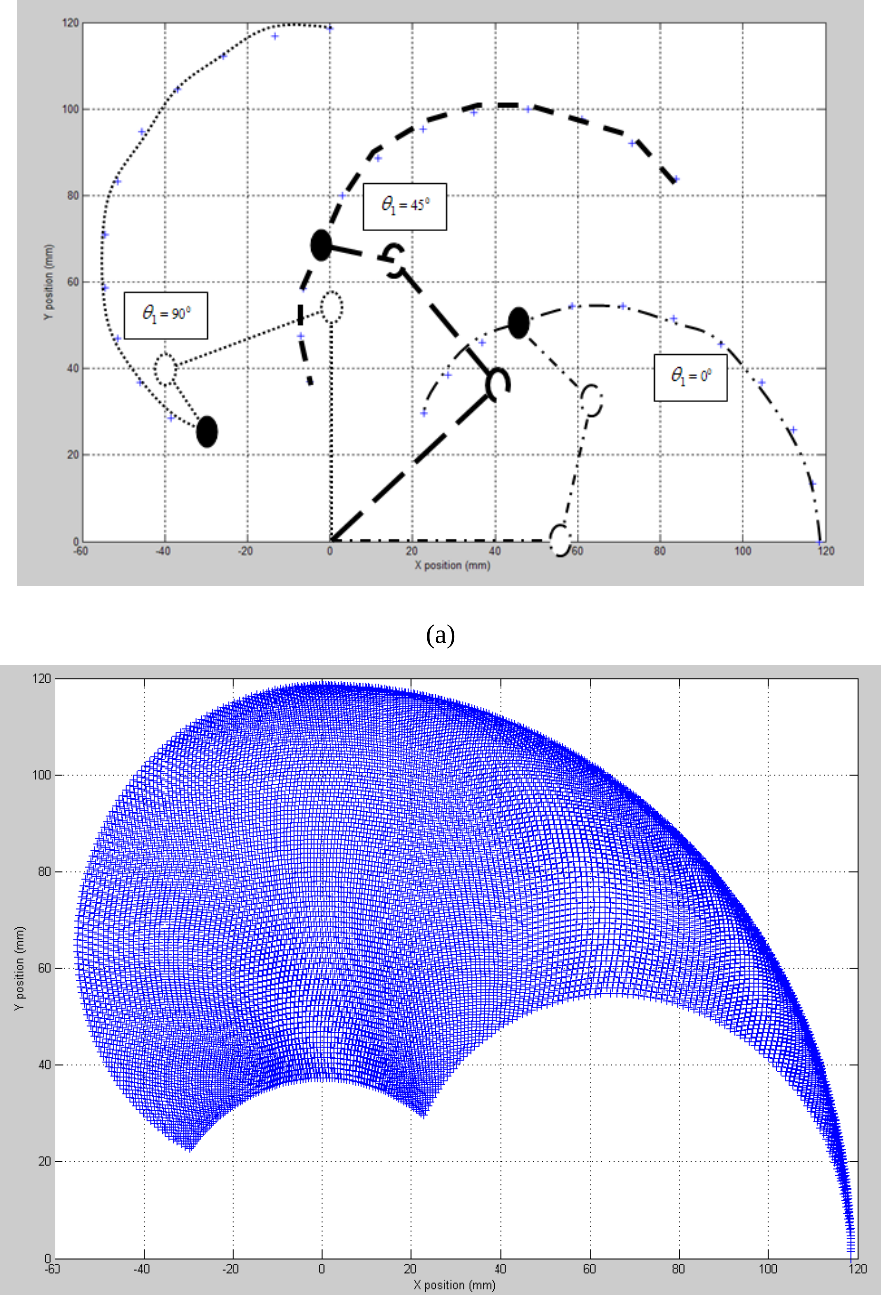
$$\begin{aligned} X_{FingerTip} &= l_1 \cos\theta_1 + l_2 \cos(\theta_1+\theta_2) + l_3 \cos(\theta_1+\theta_2+\theta_3) \\ Y_{FingerTip} &= l_1 \sin\theta_1 + l_2 \sin(\theta_1+\theta_2) + l_3 \sin(\theta_1+\theta_2+\theta_3) \\ Z_{FingerTip} &= 0 \\ \theta_{FingerTip} &= \theta_1+\theta_2+\theta_3 \end{aligned} \quad \text{------- (Equations 5)}$$

Where: l_i = length of the *i*th phalanx

θ_i = Angle of the *i*th joint

1, 2, 3= represent MCP joint, PIP joint, and DIP joint respectively

The fingertip working envelope shown in Figure 5 is based on the Equations 5 and the values of Table 2.



(a)

(b)

Figure 5: (a) Fingertip working Envelope of MCP joint (θ_1) in 45^0 interval with 3 finger links, (b) Finger Tip working Envelope of MCP joint (θ_1) in 1^0 interval.

Figure 5, illustrates that the position Y ranges from $0mm \le Y_{FingerTip} \le 118.55\text{mm}$, while the position X ranges from $-54.80mm \le X_{FingerTip} \le 118.55\text{mm}$. Table 4 shows the corresponding joint angles with respect to minimum and maximum of (X,Y). An important conclusion is that the prosthetic fingertip working envelope has the same range of movement as the main author's middle finger, which the finger design was based upon. Therefore, it can be claimed that it provides a level of functionality close to that of an actual human middle finger.

Table 3: Min. and Max. X, Y values corresponding to Min and Max joint angles.

	$X_{FingerTip}$	θ_1	θ_2	θ_3	$Y_{FingerTip}$	θ_1	θ_2	θ_3
minimum	-54.8mm	90^0	65^0	47^0	0	0	0	0
maximum	118.55mm	0	0	0	118.55mm	90^0	0	0

6. Conclusions

This paper presented a design of a prosthetic middle finger based on a hybrid actuator with the aim to resemble the kinematics architecture of a human middle finger. The hybrid actuator is a combination of a DC actuator and of a shape memory alloy actuator. The prosthetic finger has 3 DOF in contrast to 4 DOF of the human finger, due to the fact that the movement of abduction-adduction in the MCP joint is not necessary for the middle finger to accomplish grasping functionality. The forward kinematics working envelope showed that this novel design can realise a prosthetic middle finger capable of achieving the range of movement of a human one, and hence closely resemble its functionality. Future work involves the implementation of a control unit for object grasping and handling. The design of the control unit will be based on surface Electromyography (sEMG) signal of the forearm muscles. Together with the measurements of the feedback sensors, a close-loop control system will be implemented for accurate and effective operation of the prosthetic finger.

References:

[1] K. D. Charles Pfeiffer, and Constantinos Mavroidis, "Shape Memory Alloy Actuated Robotic Prostheses: Initial Experiments," Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on, vol. 3, pp. 2385-2391, 1999.

[2] J. B. Rosmarin and H. H. Asada, "Synergistic Design of a Humanoid Hand with Hybrid DC Motor-SMA Array Actuators Embedded in the Palm," presented at International Conference on Robotics and Automation, Pasadena, CA, USA, pp. 773-778, 2008.

[3] G. Roznowski and M. Drzewiecki, "A New Approach to the Prosthetic Finger Design," presented at Modern Problems of Radio Engineering, Telecommunications and Computer Science, pp. 545-547, 2004.

[4] F. Rothling, R. Haschke, J. J. Steil, and H. Ritter, "Platform Portable Anthropomorphic Grasping with the Bielefed 20-DOF Shadow and 9-DOF TUM Hand," presented at Inernational conference on Intelligent Robots and Systems, San Diego, CA, USA, pp. 2951-2956, 2007.

[5] V. Bundhoo, E. Haslam, B. Birch, and E. J. Park, "A shape memory alloy-based tendon-drive atuation system for biomimetic artificial fingers, Part 1: Design and Evaluation," Cambridge Universtiy, Cambridge, pp. 1-16, 18 March 2008.

[6] V. Bundhoo and E. J. Park, "Design of an Artificial Muscle Actuated Finger Towards Biomimetic Prosthetic Hands," presented at Advanced Robtoics, ICAR '05. Proceedings., 12th International Conference, pp. 368-375, 2005.

[7] Y. Bar-Cohen, T. Xue, M. Shahinpoor, J. O. Simpson, and J. sMITH, "Low-mass muscle actuators using electroactive polymers (EAPs)," presented at Proceedings of SPIE's 5th Annual

International Symposium on Smart Structures and Materials, San Diego, pp. 3324-3332, 1998.

[8] K. J. DeLaurentis, A. Fisch, J. Nikiteczuk, and C. Mavroidis, "Optimal Design of Shape Mmory Alloy Wire Bundle Actuators," Robtoics and Automation. Proceedings. ICRA '02. IEEE International Conference, vol. 3, pp. 2363-2368, 2002.

[9] M. L. L. a. V. M. Z. Minoru Shinohara, "Age Effects on force produced by intrinsic and extrinsic hand muscles and finger interaction during MVC tasks," *Journal of Applied Physiology*, vol. 95, pp. 1361-1369, 2003.

[10] Y. A. Arslan and Y. Hacioglu, "Modeling and Control of a Humanoid Robot Hand Finger," presented at Proceedings of 5th International Symposium on Intelligent Manufacturing System, pp. 1141-1149, 2006.

[11] K. S. Fu, *Robotics: Control, sensing, vision, and intelligence*: New York; London: McGraw-Hill, pp. 12-81, 1987.