HUMAN ROBOT INTERACTION USING KNOWLEDGE BASE APPROACH

by

UTSAV SHAH

THESIS

Submitted in Partial fulfillment of the requirements
for the degree of Master of Science in Mechanical Engineering at
The University of Texas at Arlington
May,2016

Arlington, Texas

Supervising Committee:

Panayiotis S. Shiakolas, Supervising Professor
Alan P. Bowling
Ralph S. Gibbs

Dedicated to my Parents

ACKNOWLEDGEMENTS

I would like to thank my principal advisor, Dr. Panayiotis S. Shiakolas, for his constant guidance, motivation and support throughout my graduate research work. I would also like to thank my graduate committee members Dr. Alan P. Bowling and Dr. Ralph S. Gibbs for accepting my request to be member of my Master's thesis committee.

I would also like to thank my colleagues at Micro Manufacturing Medical Automation and Robotics Systems (MARS) laboratory including Michael Abraham, Prashanth Ravi, Tushar Saini, Shashank Kumat, Samson Adejokun for their advice and support during my research work. Special thanks to Christopher Abrego for his guidance and motivation during my time at MARS laboratory. I also wish to thank MAE machine shop supervisor Mr. Kermit Beird and Mr. Sam Williams for their help with equipment fabrication.

Many thanks to Dr. Nomura, Dr. Moon, Dr. Agonafer whom educated me during my graduate studies in UTA. I also wish to thank Dr. Manry in EE Department for introducing me to Machine Learning.

April 25, 2016

ABSTRACT

HUMAN ROBOT INTERACTION

USING KNOWLEDGE BASE

APPROACH

Utsav Shah, MS

The University of Texas at Arlington, 2016

Supervising Professor: Panayiotis S. Shiakolas

This research investigated Human Robot Interaction modalities. The performance of a robotic prosthetic hand (RPH) was used as a test bed robot. A user wearable glove was fitted with sensors to provide tele-control of the RPH. An open hardware control and easily expandable research platform based on LabVIEW software and myRIO control hardware was developed. LabVIEW graphical programming platform provides the tools for the development to customized interfaces for visualization purposes which is desired in research. The research platform was used to calibrate and control the operation of the artificial hand using various modalities such as open loop, glove master-slave setup and knowledge base interaction. Mapping algorithms between the motion of the master glove and slave RPH were developed.

The knowledge based modality was based on artificial neural networks (ANN), where supervised learning identified appropriate grasping patterns for a set of objects based on a training data set. The training data set was developed using manual and glove control of the RPH and consists of the object geometric features and object location relative to the RPH. The training data set is then processed using the LabVIEW ANN toolkit to identify in real-time, the grasping patterns for other similar objects that include the desired motion for each RPH finger. The developed research platform and tools have been demonstrated through manual, glove and ANN control of the RPH and display of system information on the LabVIEW GUI in real-time.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Human Robot Interaction

Human Robot Interaction (HRI) describes the study of communication to understand, design and evaluate robotic systems by humans [1]. The HRI has been utilized in the physical, mental and social assistance to humans by robots. The initial step of HRI is to understand the behavior of robotic devices which correspond to human like interaction methods. Consequently, design and classification of a methodology where human type intelligence can be used to control robotic systems was further understood. Figure 1.1 is an example of HRI where the prosthetic hand is performing a hand shake with a human hand.

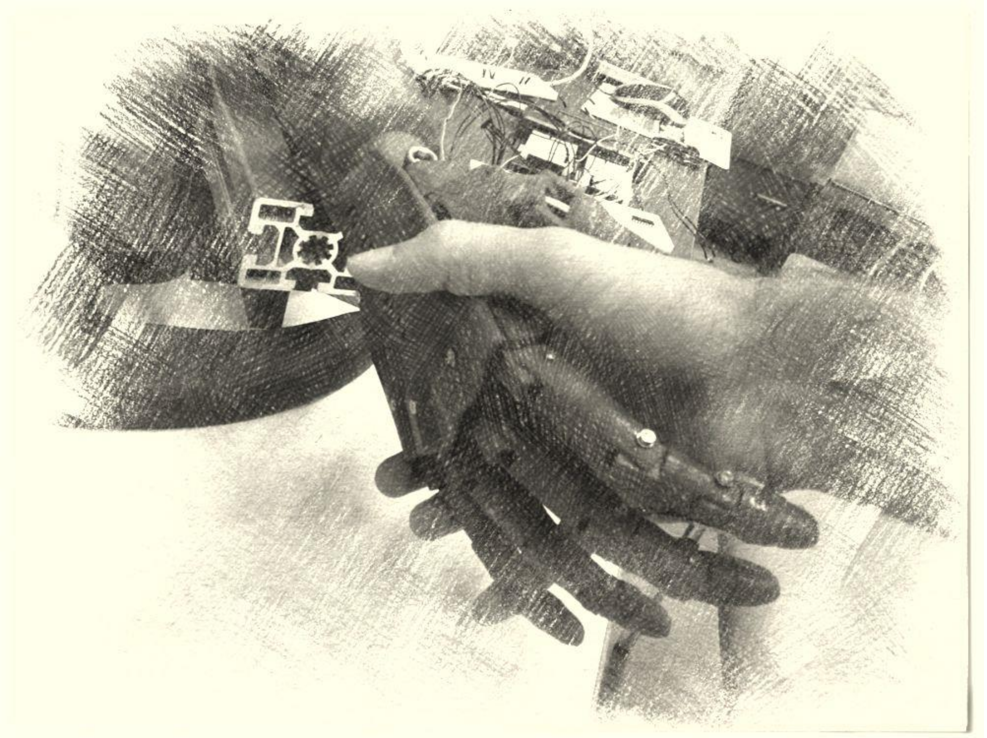


Figure 1.1 Handshaking between a prosthetic and a human hand

Robotic systems can execute various tasks in social applications such as, medical, welfare, home, offices and industries in cooperation with human [2]. There are various possible interaction techniques such as speech, gesture and body language for a robot to interact with human. For example, the master-slave hand that was developed by the University of Tokyo which uses haptic interaction to control robotic hand with human hand [3]. The feature of speech and gesture interaction is generally found in social robot applications. The Leonardo robot, for instance, developed by MIT uses social learning experiments to interact with humans [4]. The National Aeronautics and Space Administrations (NASA) also developed Robonaut, a humanoid capable to work in hazardous environment of low earth orbit and planetary exploration [5].

In order to investigate HRI, an environment that includes a testbed robot is required where the robot will be controlled by human. In this research, a prosthetic hand is used as the testbed robot and focuses on implementing HRI on a prosthetic hand to accomplish various grasping tasks. There is extensive on-going research to obtain object grasping information by classifying motion of the human hand [6].

1.1.1 HRI with Prosthetic Hand

Controlling a prosthetic hand with a human hand is the most practical approach in the field of HRI and it generally uses a data glove to map human hand motions onto the prosthetic hand. Commercial data gloves calibrate the flexion and abduction of the fingers and angular motion of palm using approximately 20 sensors on the glove [7]. Moreover, various sensing approaches are available to sense human hand motion which usually require recalibration for different users. Table 1.1 shows a summary of data gloves along with their sensing technologies.

Table 1.1 Use of sensing devices in various data gloves [7]

Data Gloves	**Sensing Device**	**Price**
VPL Data glove	Fiber Optic	$ 11000
Virtex Cyber Glove	Resistor	$9800
TUB-sensor glove developed by Technical University of Berlin	Pressure & Position sensor	-
Accele Glove developed by Washington University	Accelerometers	-
Polhemus 3-Space Fastrack (not glove)	EMG sensors	-
Mattel’s power glove	Ultrasonic sensor	$100

Most of data gloves are expensive depending on their design and sensing device. The preliminary requirement in this research is to develop a research platform where a prosthetic hand can accomplish various grasping tasks using a manual, glove guided and machine learning approaches.

1.2 Robot Learning

Robot learning is focused on developing skills in robots using machine learning. Robot learning through human demonstration is one of the difficult tasks in current intelligence system [8]. Robot learning is often characterized as machine learning where the term machine is replaced by a robotic system. Machine learning is a field of study where a learning skill can be developed on a computer without explicit programming, as defined by Arthur Samuel in 1959 [9]. Machine learning is a part of artificial intelligence which focuses on constructing a learning algorithm using an artificial neural network (ANN). A part of this research focuses on creating a learning environment where prosthetic hand can be used to grasp complex object using machine learning approaches.

1.2.1 Grasp Learning in Robots

Robotic technologies have been researched in order to accurately solve the control and perception problem of grasping for decades (Shimoga, 1996). Currently, there are difficulties in industries when it comes to handling complex parts by robots. Therefore, grasp learning is an emerging field in industrial settings due to its ability to “learn” to handle complex parts. Google is researching a 7-degree of freedom robot used to grasp various objects using a vison system and machine learning algorithm with neural network [10]. Different universities such as Carnegie Mellon University is also investigating machine learning algorithms to implement grasp learning on the industrial robot Baxter [11] . However, the novel approach to perform grasp learning using a prosthetic hand could be adapted in industries to handle complex object.

1.3 Current Research Objectives

The objective of this research is to develop an expandable, modular software and hardware environment to research grasp learning in a prosthetic hand using different HRI modalities. The software tools are based on the LabVIEW graphical programming environment. The hardware tools include mainly a robotic prosthetic hand (RPH), human wearable data glove, electrical interfacing boards and myRIO microcontroller.

Grasp learning is achieved by implementing a supervised learning algorithm based on back propagation neural network, using the LabVIEW machine learning toolkit. The HRI technique is achieved by controlling a prosthetic hand in manual, tele operated and autonomous modalities.

The learning performance is analyzed for known as well as unknown objects. Experiments on learning and evaluating for grasping various objects with a prosthetic hand using machine learning algorithms emerge as a novel approach in the development of decision making skills for robots.

1.4 Outline of Thesis

Chapter 1 provides an introduction on HRI and machine learning, examples and application areas. It also discusses various approaches to implement HRI on the RPH. The objective related to the current research work is also discussed in this chapter.

Chapter 2 discusses geometry, actuation system and grasp space of the RPH. Moreover, it discusses flex sensor glove development and terminology and introduces the back propagation learning algorithm.

Chapter 3 discusses the required hardware modules such as the National Instruments (NI) myRIO microcontroller, flex sensor interfacing board, servo interfacing board and force feedback interfacing board. The software used is LabVIEW. It also discusses the flow of information between software and hardware modules. It also explains the control algorithm for the servo motors and control of the RPH with the flex sensor glove.

Chapter 4 discusses how the system will learn grasping and evaluation of the RPH ability to grasp various objects. This chapter also discusses the performance of the artificial neural network (ANN) and explains the back propagation learn and evaluation procedure to perform grasp learning by the RPH. The chapter also describes speed control implementation for finger actuation.

Chapter 5 summarizes the research work on grasp learning in the RPH and HRI approaches and provides recommendations for future research.

CHAPTER 2

PROSTHETIC HAND ANALYSIS AND INTERFACE

Most industrial robots use parallel jaw grippers which are sufficient for handling objects in current applications. However, the future technological innovations focus on investigating grippers that can handle complex geometrical objects. Prosthetic hand development is growing due to its enhanced grasping capability of handling complex parts in the medical and industrial fields. There are extensive amounts of money spent on research and development of different prosthetic hand projects such as the APL arm ($120 million project) and the DEKA arm ($40 million project) [12], [13]. The Michelangelo Hand developed by Advance Arm Dynamics costs $100,000 (Pittman, 2012) [14] . The design and manufacturing process is also time consuming for prosthetic hand based on the degrees of freedom and actuation strength.

2.1 Robotic Prosthetic Hand Mechanism and Geometry

2.1.1 InMoov Prosthetic Hand

The discussion of prosthetic hands has concluded that the design and manufacturing process is expensive and time consuming. Instead of analyzing and creating a new design, it is more convenient to modify and improve an existing design. In this research, part files of a right hand were downloaded from the InMoov project developed by Gael Lengavin and 3D printed [15]. Afterwards, the 3D printed components, such as forearm, fingers and palm, were assembled as shown in Figure 2.1

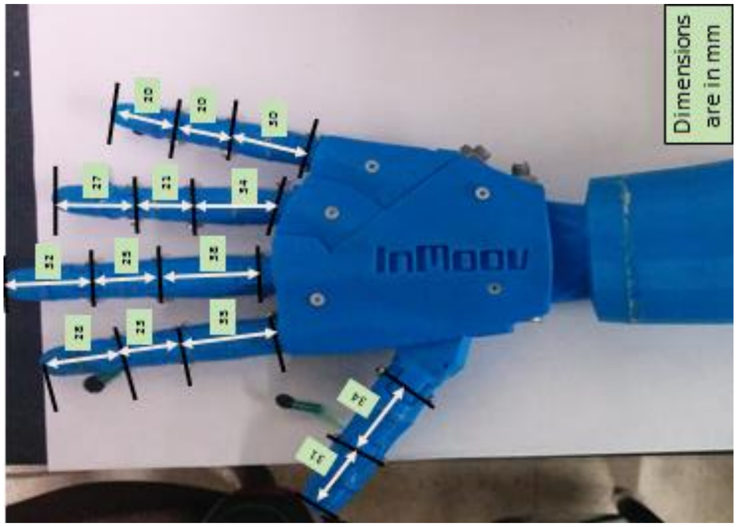


Figure 2.1 Assembled InMoov prosthetic hand with dimensions in mm

The RPH has a span of 195mm (distance from outstretched pinky to outstretched thumb) and a hand height of 212mm (distance from wrist to tip of middle finger) [16]. Each finger consists of three joints and the thumb consists of two joints. The actuation of each finger consists only of one motor.

2.2.2 Workspace

The objective of the prosthetic hand is focused on grasping similar objects. Therefore, a convenient workspace was established considering the different constraints of the RPH. The first constraint is that the RPH has no actuation in the wrist nor in the elbow joint. Currently, the RPH has actuation only for the thumb and fingers. Therefore, for this research, it is assumed that the prosthetic hand would be properly orientated with respect to Z-axis in three dimensional space so that the object that will be grasped is within the reach of the prosthetic hand. The workspace setup is shown in Figure 2.2.

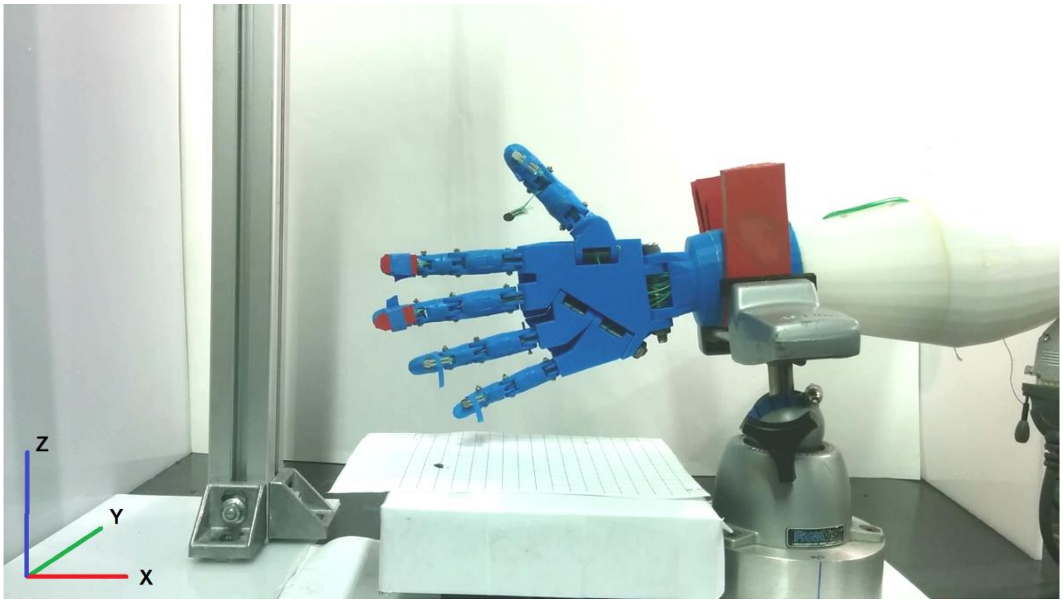


Figure 2.2 Prosthetic hand with workspace

Another important and challenging task is the handling process of objects in three dimensional space. A static setup was created where the position of the object can be adjusted with respect to Z-axis. However, the X and Y position of object is fixed to simplify grasping analysis. This setup was used to reproduce grasping operations for training purposes. Therefore, a three dimensional work space was developed where position of RPH was fixed as well as the position of the object. However, the setup allows for slight changes in hand orientation and height of the object. Figure 2.3 shows the top view and Figure 2.4 shows the front view of 3D workspace 17’’×21’’×22’’.

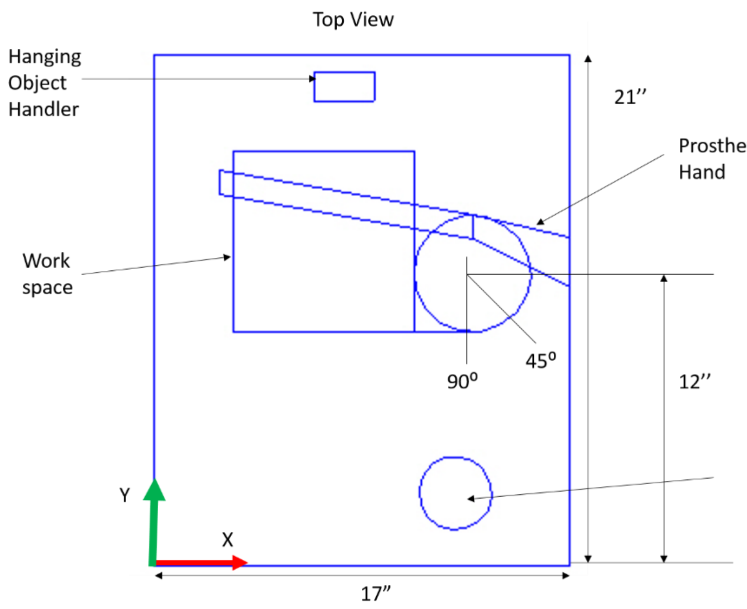


Figure 2.3 Top view of workspace (dimensions in inch)

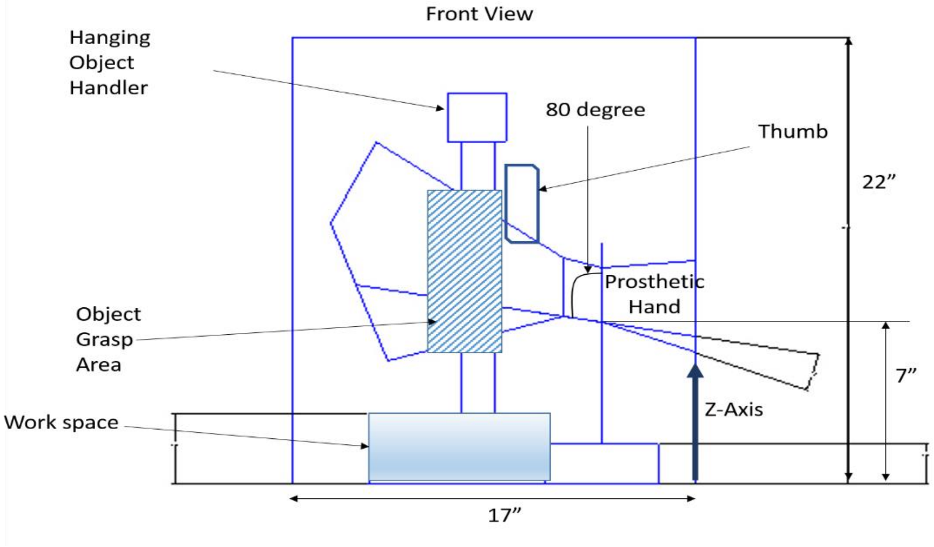


Figure 2.4 Front view of workspace (dimension in inch)

2.2 Robotic Prosthetic Hand Actuation Analysis

2.2.1 Actuation Type

There are different actuation methods available to control a RPH. The first actuation method is to apply actuation on each finger joint with geared motors as shown in Figure 2.5. However, this form of actuation, implementing a motor on each joint, increases the total size, weight and increases the control complexity of RPH [17]. Furthermore, small/compact motors with appropriate torque are expensive. However, there are benefits, such as being more stable and flexible for controlling the individual joints of the prosthetic hand.

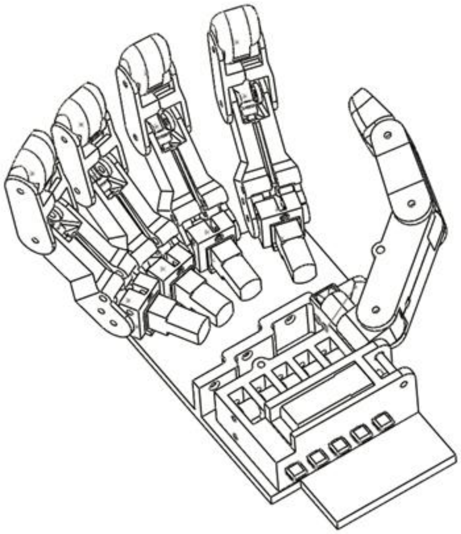


Figure 2.5 Prosthetic hand with actuation on finger joints [10]

Another actuation method used to control the joints of the fingers is a cable-pulley mechanism where two cables are passed through each joint. The cables are then fixed at the end of the fingertip as well as at the pulley mechanism. It is important to note that the cable has to be fixed permanently on motor-gear and tension in the cable attached to finger-tip should always be maintained. It should also be noted that the design of the gear attached to motor is important to achieve maximum finger rotation.

2.2.2 Prosthetic Hand Actuation

A second actuation method is employed in the RPH. Servo motors are mounted in the forearm of the assembly and connected to each finger by a cable-pulley mechanism, as shown in Figure 2.6. In this actuation method, each finger has only one degree of freedom because a single servo motor controls all three joints in each finger.

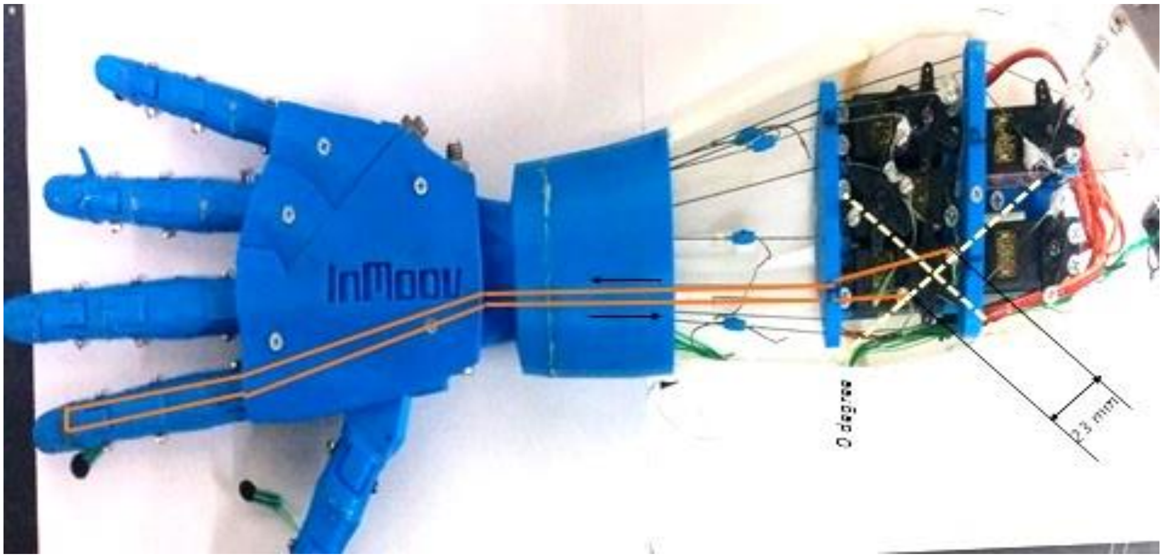


Figure 2.6 RPH actuation mechanism

The position of the line passing through the grooves is different for each finger. Therefore, the minimum and maximum possible rotation (initial and final finger position) of the servo motor is dependent on the distance d_c between the grooves from where cables are passing and attached on motor gear and the gear diameter d_g.

Each finger rotation is dependent on the linear travel distance of two cables attached to servo motor gear. Figure 2.7 illustrates that the servo motor angle range of Θ_s increases as d_c

increases. Therefore, the servo motor attached to middle finger has higher angle range than index finger because it has a larger value of d_c.

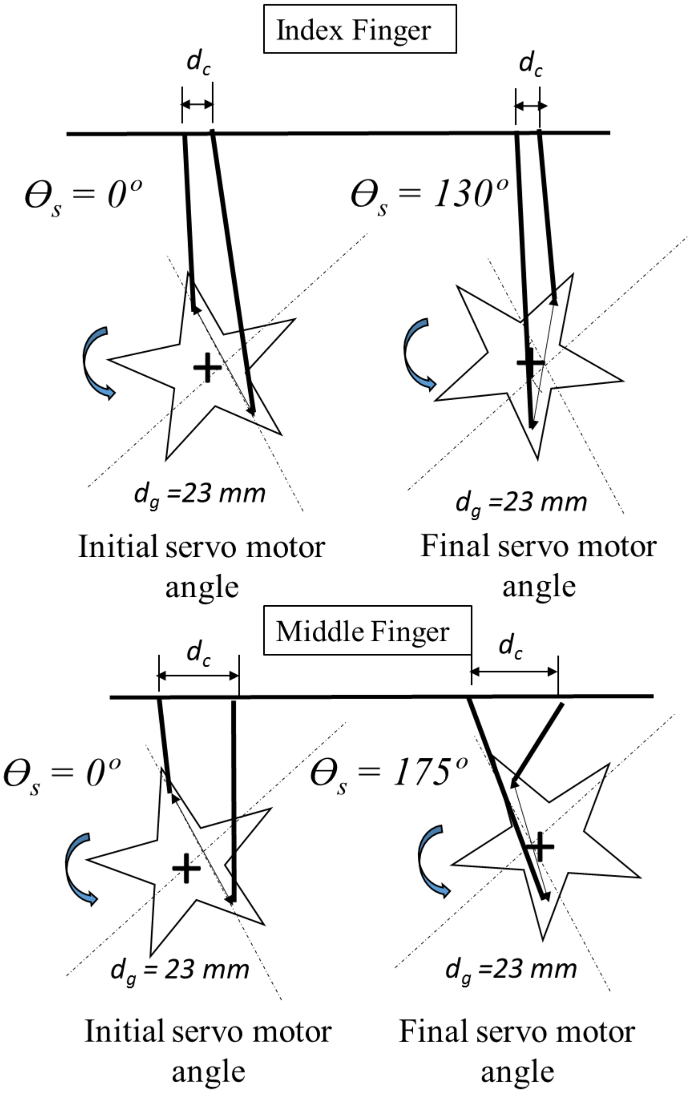
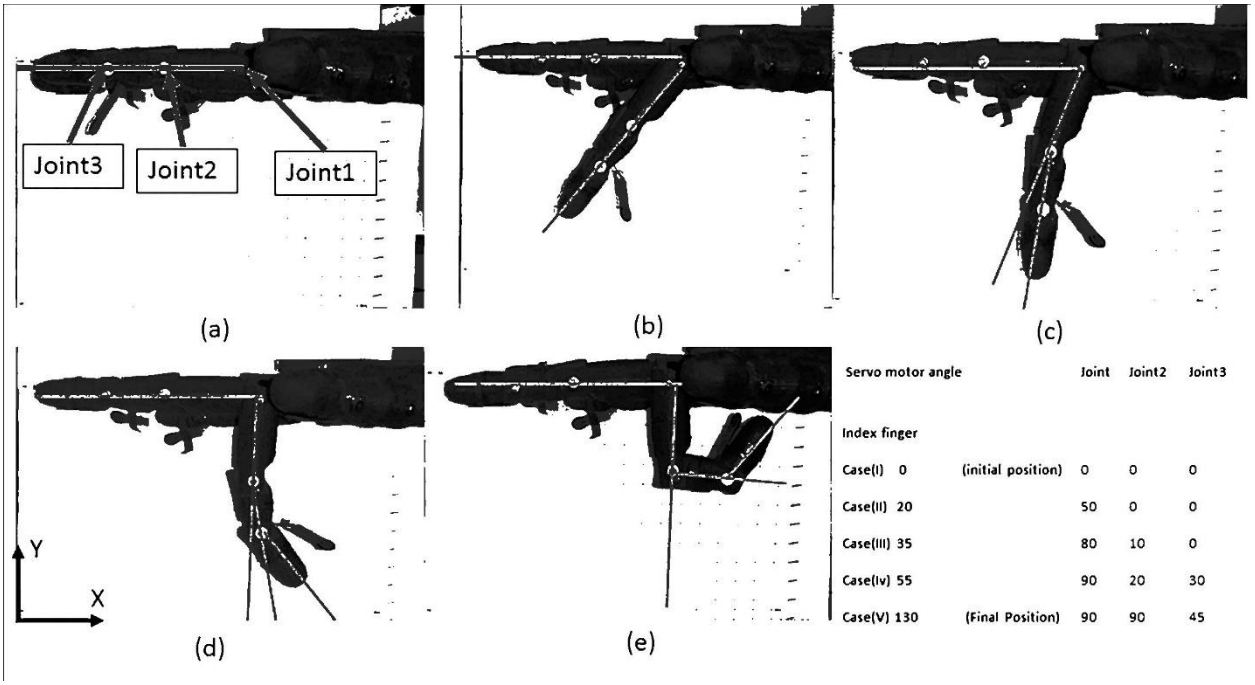


Figure 2.7 Servo motor angle difference with change in cable distance for index and middle finger

2.2.3 Behavior of Fingers

In the RPH, there are several motion constraints regarding the degrees of freedom in the fingers and thumb. Only one servo motor is provided to control the three revolute joints on each finger. Figure 2.8 shows the behavior of the index finger for different servo motor angular rotation. The index finger has proximal, intermediate and distal links that are interconnected on joint 1, joint 2 and joint 3 respectively. In this discussion, the term joint 1 angle is the angle between proximal link and X axis, Joint 2 angle is the angle from proximal link to intermediate link and Joint 3 angle is the angle from intermediate to distal link. Figure 2.8(a) shows the initial position of index finger where all three joint angles are zero degrees as well as the servo motor angle. In Figure 2.8(b), the servo motor rotation of 20 degrees sets joint 1, joint 2 and joint 3 to 50, 0 and 0 degrees respectively. The intermediate position where the servo motor rotation is 90 degrees, sets joint 1, joint 2 and joint3 to 90, 20 and 30 degrees respectively as shown in Figure 2.8(d). The final position of index finger is achieved at 130 degrees of servo motor rotation which sets joint 1, joint 2 and joint 3 equal to 90, 90 and 45 degrees respectively.



Servo motor angle		Joint	Joint2	Joint3
Index finger				
Case(I) 0	(initial position)	0	0	0
Case(II) 20		50	0	0
Case(III) 35		80	10	0
Case(Iv) 55		90	20	30
Case(V) 130	(Final Position)	90	90	45

Figure 2.8 Behavior of finger for different servo motor angular rotation

A protractor was adjusted parallel to rotational axis to measure different joint angles as shown in Figure 2.9. A more accurate approach of measuring joint angles could be a vision system. There would need to be a calibration method to be able to compare the original size and the joint angle measurements on the image.

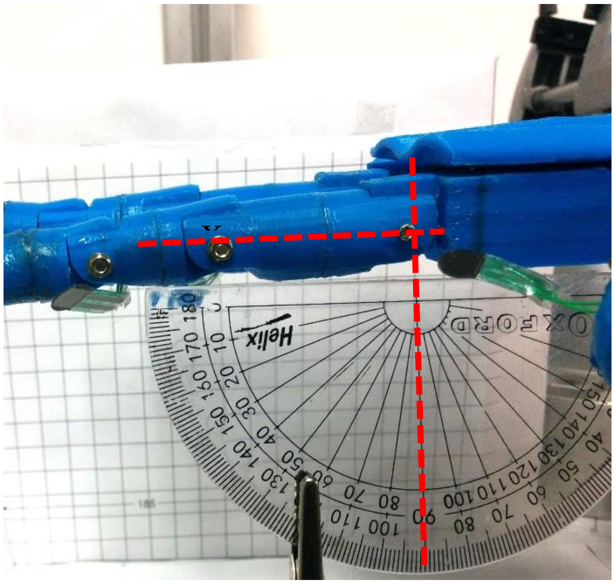


Figure 2.9 Protractor arrangement to measure joint angles of fingers

The measurements of joint angle are performed on all fingers and thumb which concluded that joint angle of fingers are different with respect to servo motor angle. Table 2.1 shows the calibration mapping of servo motor rotation to finger state. Note that thumb has only two joints. Therefore, measurement of position of joint 3 is not applicable for the thumb.

For the initial position, all fingers are at zero degrees with respect to the X-axis. However, in the final condition, the behavior of each finger is different depending on their servo motor attachment and finger geometry. Figure 2.10 shows initial and final positions of all fingers. The motion of thumb is shown in Figure 2.14 which will be explained in further discussion.

Table 2.1 Experimental relation between servo motor angle and finger joints

	Joint 1 Angle (deg)	**Joint 2 Angle (deg)**	**Joint 3 Angle (deg)**
Index finger			
0 (Initial position)	0	0	0
20	50	0	0
35	80	10	0
55	90	20	30
130(Final position)	90	90	45
Middle finger			
0(Initial position)	0	0	0
30	60	0	0
100	90	10	10
175(Final position)	90	70	30
Ring finger			
0(Initial position)	0	0	0
30	60	0	0
60	90	10	10
130(Final position)	90	50	30
Pink finger			
0(Initial position)	0	0	0
12	10	0	70
30	60	0	70
40	80	0	70
90(Final position)	80	50	70
Thumb			
0(Initial position)	0	0	NA
10	0	0	NA
20	35	0	NA
90(Final position)	60	50	NA

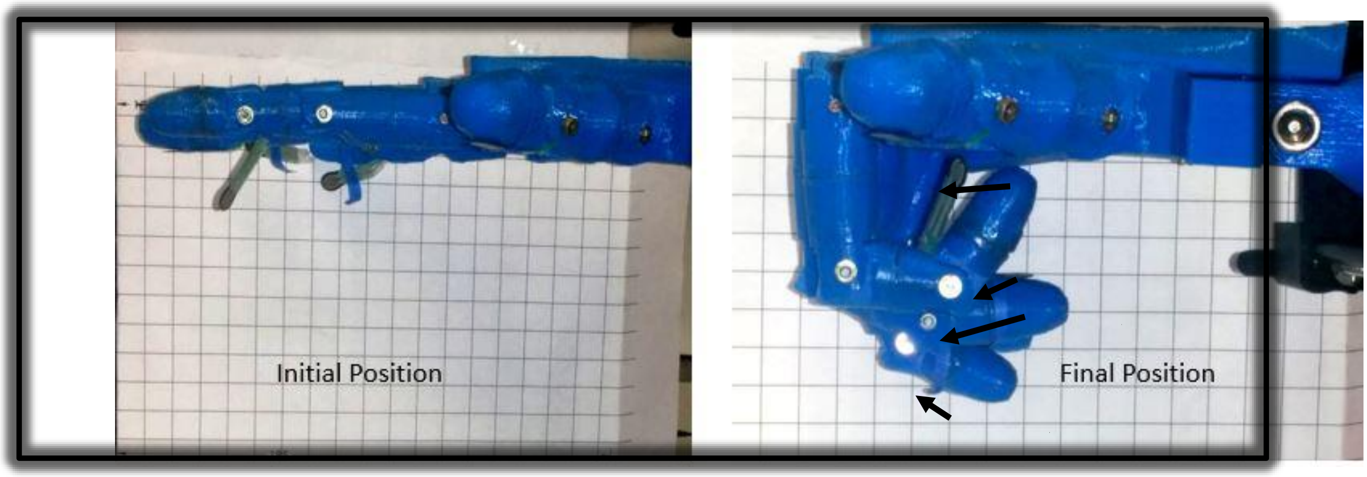


Figure 2.10 Finger positions for initial and final servo motor angle

2.3 Grasping Pattern Analysis

2.3.1 Grasping Art

The operated complexity of the RPH prohibits grasp point generation and force manipulation which are dependent on hand kinematics. Object geometry, hand constraints and the manipulation task mechanism intersect produce the grasp space constraints as shown in Figure 2.11.

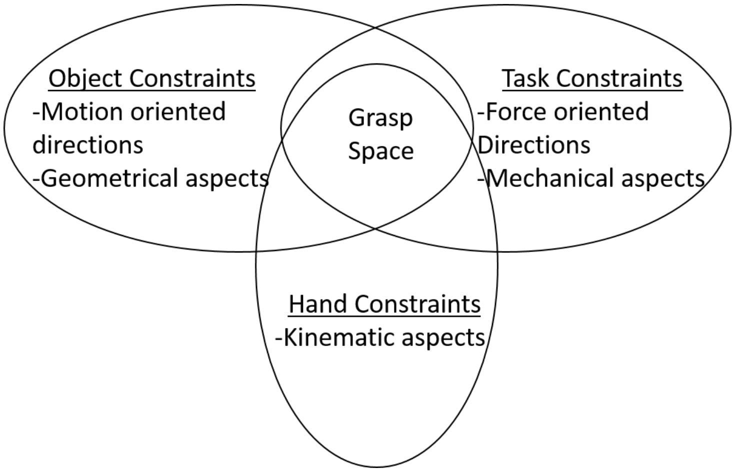


Figure 2.11 Grasp planning restrictions [18]

Object grasping can be defined as the process of placing the RPH relative to the target object for manipulation while holding the object stable[18]. However, in the current analysis, the manipulation of RPH is fixed because there is no actuation on the equivalent wrist and elbow joints. A precise grasp consists of finger contact locations, contact type and applied force characteristics [18]. However, the applied force is dependent on weight and surface finish of object which are not considered in this study. This discussion is focused on creating different grasping patterns by varying the location of the object relative to the palm and object geometry.

2.3.2 Grasp Planning

There are different types of contact grasp such as friction point contact, hard finger contact, soft finger contact, form closure grasp, force closure grasp, equilibrium grasp, stable grasp and compliant grasp [18]. This study focuses on form closure grasp where the set of contact points contain the object grasped regardless of the magnitude of the contact force. The finger contacts are entirely dependent on geometry of the object and its relative location.

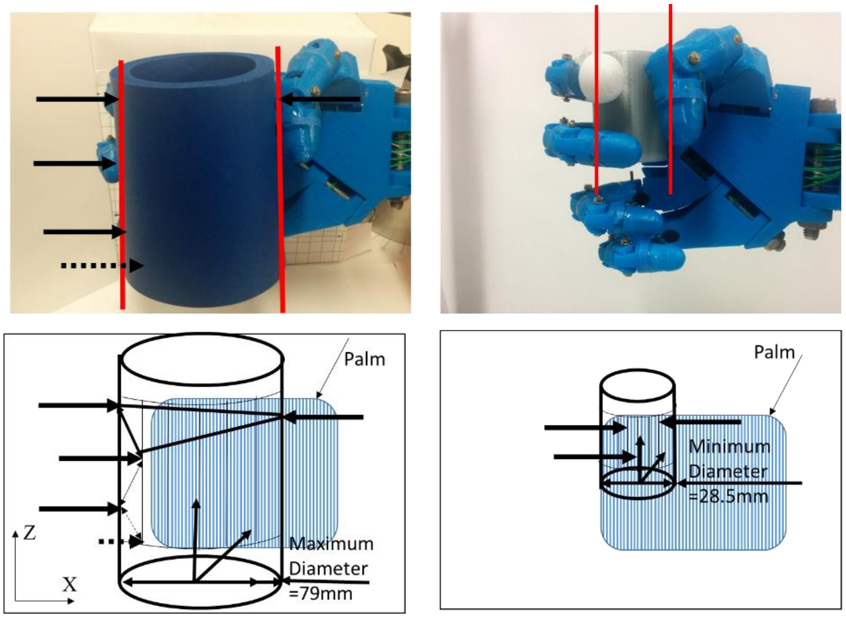


Figure 2.12 Grasping of two different dimensional cylindrical objects and contact points

The prosthetic hand has numerous contact points such as fingers, thumb and palm which allow for more efficient grasping. Figure 2.12 illustrates form closure grasping on the prosthetic hand for two cylindrical objects with different geometrical properties. The thumb always provides the first contacting support to hold any object as shown in Figure 2.12.

The palm of prosthetic hand, additionally provides a base support which restricts the sliding of object in positive Y direction marked as green arrow in Figure 2.13.

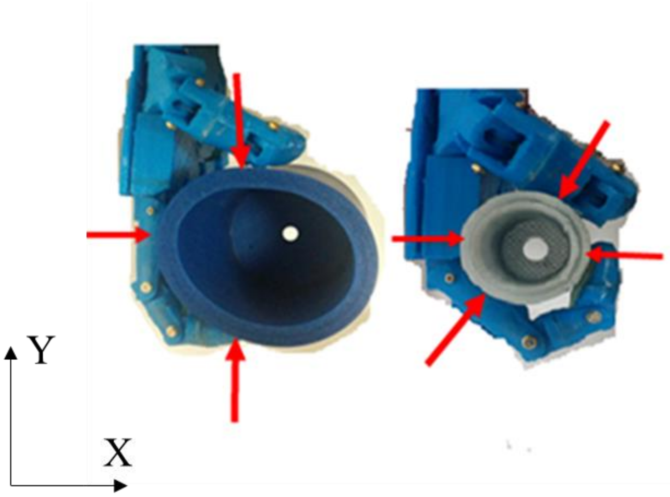


Figure 2.13 Top view of Figure 2.12

2.3.3 Grasping Space

Grasping space is a term used to describe the workable region for the RPH to grasp an object. Grasping space can be created based on physiology of the RPH. The rotational path of tip of thumb allows to create a grasping space depending on the rotation of fingers. The grasp space (80×65×45 mm) is created considering rotation path and geometrical properties of fingers and thumb. The reference frame of grasp space is 5mm offset from thumb tip as shown in Figure 2.14. The object's contact points must exist inside the grasping space as shown in Figure 2.14.

Some exterior contact points can be applicable with the support of the palm of the RPH. However, these contact points just provide extra support which is not mandatory for every grasping pattern. The X-axis distance between the supporting contact point and the remaining contact points should not be larger than grasping space to perform an efficient grasp.

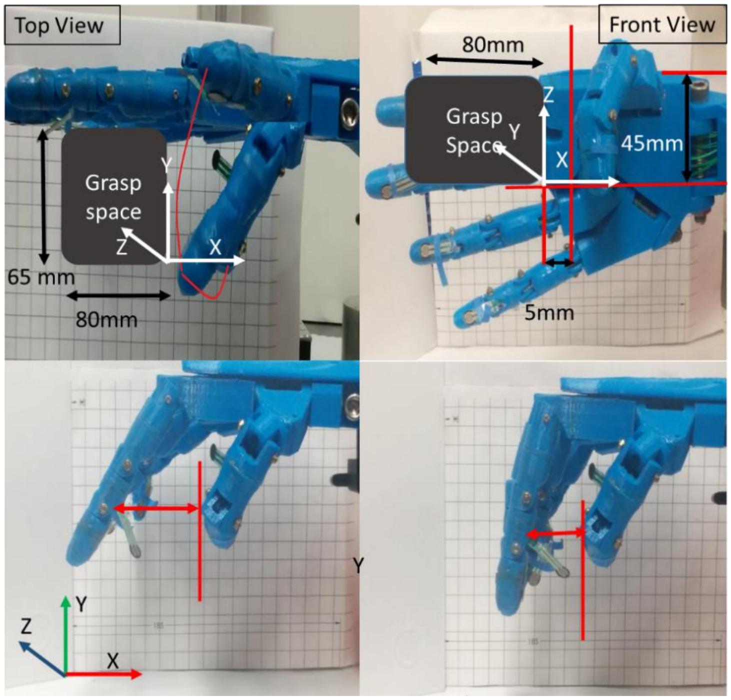


Figure 2.14 Grasp space analysis for the RPH

The solid model as show in Figure 2.15 is developed for the final position of fingers considering the joint angles represented in Table 2.1. The measurement of geometrical parameters of fingers were taken with Vernier caliper and implemented in solid model.

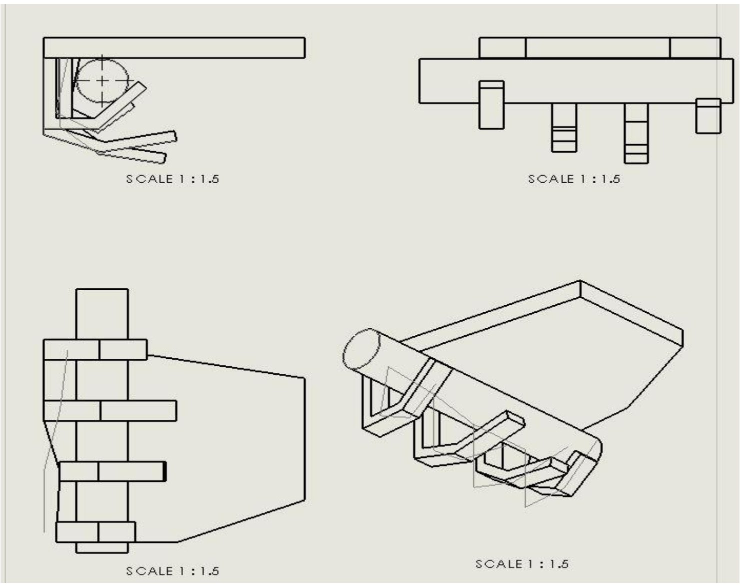


Figure 2.15 Solid model of the RPH fingers grasping a cylindrical object with a 20mm diameter

It should be noted that grasping is possible without requiring the thumb. In that case, the palm provides supporting contact and fingers provide the opposite remaining contact. However, the diameter of the object should be considerably small such as 20mm in this method, which is not convenient.

2.4 Machine Learning using Artificial Neural Network

2.4.1 Artificial Neural Network

In machine learning, artificial neural networks (an inspiration from biological neural networks) are information processing systems used to approximate functions from known or unknown inputs [19].They are used to solve complex problems with large number of interconnected processing elements[20]. The ANN with cascade connectivity consist of an input layer, hidden layer and an output layer, as shown in Figure 2.16.

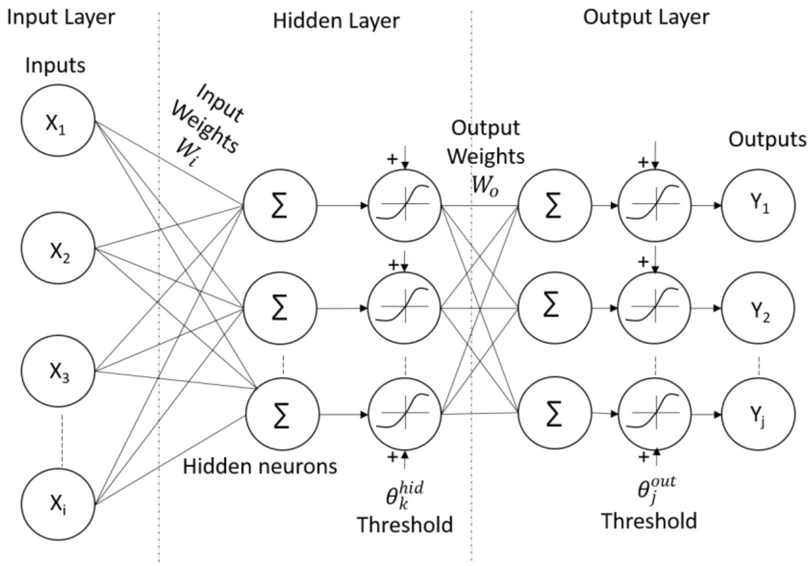


Figure 2.16 Artificial Neural Network

The approximated output,y_j, for input, x_i, and hidden neuron, k, is

$$y_j = f\left(\theta_j^{out} + \sum_{k=1}^{K} Wo_{kj} f\left(\sum_{i=1}^{N} Wi_{ik} x_i + \theta_k^{hid}\right)\right) \tag{2.1}$$

where, f is the activation function, N is the number of input neurons, K is the number of hidden neurons, Wi_{ik} is the input weight, Wo_{kj} is the output weight, x_i is the input, θ_k^{hid} are the thresholds in terms of hidden neurons and θ_j^{out} is threshold in terms of output neuron [21]. ANNs are utilized for specific applications, such as pattern recognition and data classification through learning process[20]. There are two types of learning; ANN supervised learning and unsupervised learning.

2.4.2 Supervised Learning

Supervised learning is often called as learning with a teacher, which is a widely used training method where ANN is trained through sets of known inputs and outputs. The output error is computed and weights are incrementally adjusted to reduce the error using each organized pattern [8]. These weights are adjusted by different learning algorithms such as back propagation, learning vector quantization (LVQ) and support vector machine (SVM). Figure 2.17 illustrates a network with desired output processes through supervised learning. The teacher or supervisor provides the predefined output and a supervised learning approximates the output based on the input using the ANN. The difference between the actual and approximated output is calculated using the error block and the network tries to minimize the error using the supervised learning algorithm. Supervised learning is becoming popular due to its productive results [22].

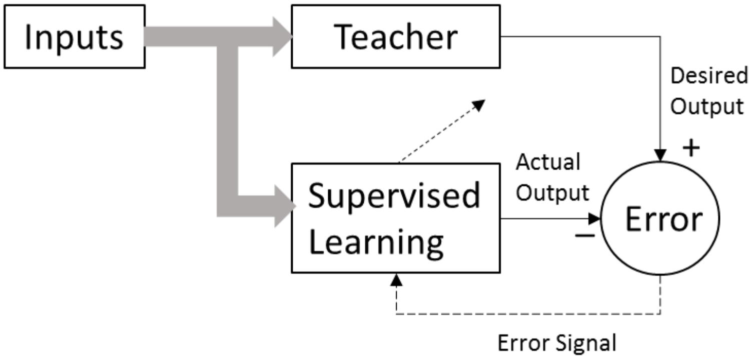


Figure 2.17 Supervised learning

2.4.3 Back Propagation Algorithm

The back propagation algorithm was first mentioned by Werbos in 1974 and “rediscovered” by Rumelhart and McClelland in 1986 [23],[24]. Back propagation neural networks are frequently used in control systems to learn system characteristics through nonlinear mapping [25]. The back propagation is widely used supervised learning algorithm for computing network weights regardless of the convergence rate. This algorithm adjusts weights and biases of a network in the direction of steepest descent depending on error [20].

A three-layer feedforward multilayer perceptron (MLP) network with cascade connectivity is shown in Figure 2.18. The diagram shows three layers; input layer, one hidden layer and an output layer. For discussion purposes 3 input, 2 hidden and 1 output neuron are considered.

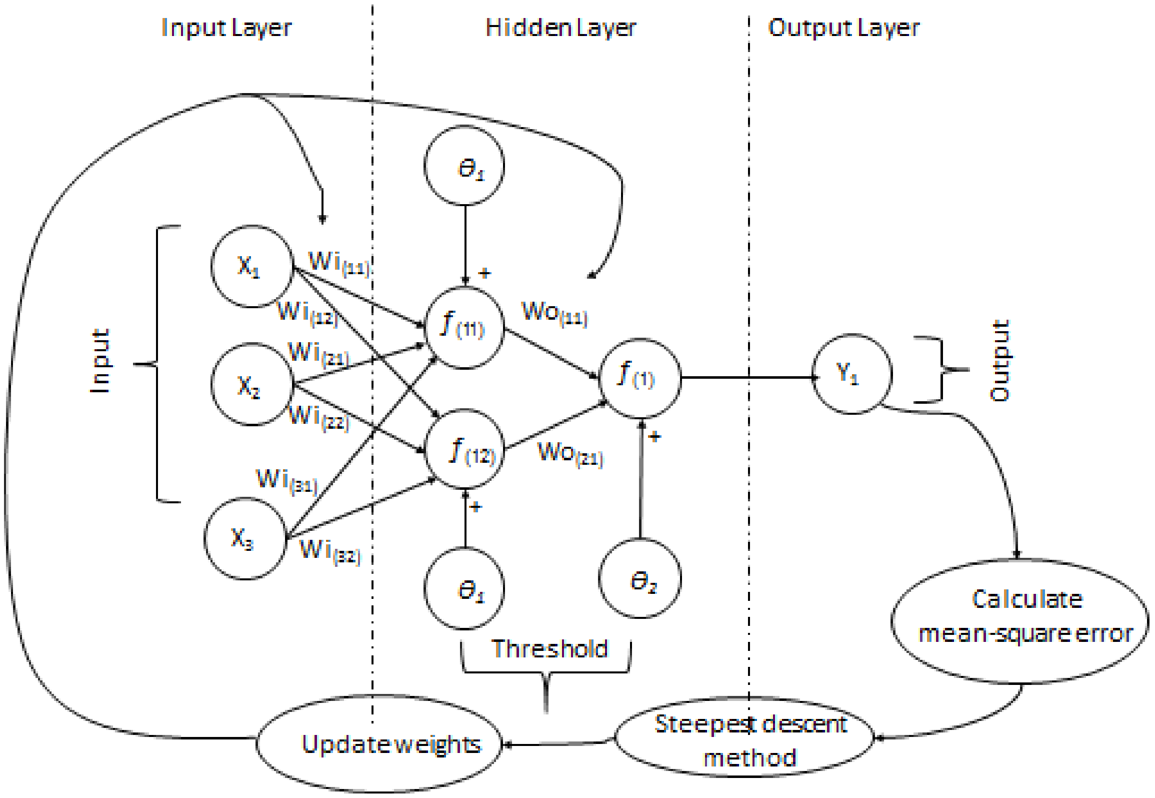


Figure 2.18 Three-layer network with back propagation algorithm for three inputs and one output.

In the initial stage, this network approximates output based on inputs, weights and hidden unit activation functions where the weights are initialized with random values. First, inputs are multiplied with weights and then added to thresholds which formulates a net hidden unit output, n as shown in Equations (2.2) and (2.3)

$$n_{(11)} = \theta_1 + X_1\, Wi_{(11)} + X_2\, Wi_{(21)} + X_3 Wi_{(31)} \tag{2.2}$$

$$n_{(12)} = \theta_1 + X_1\, Wi_{(12)} + X_2\, Wi_{(22)} + X_3 Wi_{(32)} \tag{2.3}$$

The output of net hidden unit processes through activation function to calculate hidden unit output. This activation function can be a sigmoid or a hyperbolic tangent function. For example, consider a hyperbolic tangent function as shown in Equation (2.4),

$$f_{(x)} = \tanh(x) = \left(\frac{2}{1+e^{-2x}} - 1\right) \qquad (2.4)$$

Then, the hidden unit output is calculated as shown in Equations (2.5) and (2.6) as a function of the net output.

$$f_{(11)} = f\left(n_{(11)}\right) = \left(\frac{2}{1+e^{-2(n_{(11)})}} - 1\right) \qquad (2.5)$$

$$f_{(12)} = f\left(n_{(12)}\right) = \left(\frac{2}{1+e^{-2(n_{(12)})}} - 1\right) \qquad (2.6)$$

The net output can be derived by multiplying net functions with output weights and adding output threshold as described in Equation (2.7) and the final output y_1 is approximated with output activation function as shown in Equation (2.8).

$$n_{(1)} = \theta_2 + f_{(11)}Wo_{(11)} + f_{(12)}Wo_{(21)} \qquad (2.7)$$

$$y_1 = f(n_{(1)}) \qquad (2.8)$$

2.4.4 Back Propagation Learning steps

The network is trained by finding a set of weights and threshold that minimize the error between the actual (*t*) and network output (*y*) based on the training data set according to Equation (2.9).

$$Mean\ square\ error(MSE) = \frac{1}{N_v}\sum_{p=1}^{N_v}\sum_{j=1}^{M}(t_{pj} - y_{pj})^2 \qquad (2.9)$$

where, N= number of inputs, M= number of outputs and N_v= number of patterns. For this example, number of patterns N_v=*1* and number of outputs j=*1*. Therefore, MSE is,

$$MSE = (t_1 - y_1)^2 \tag{2.10}$$

The input and output weights are updated according to steepest descent algorithm,

$$Wo_{(kj)} = Wo_{(kj)} + z. G_{kj} \tag{2.11}$$

$$Wi_{(ik)} = Wo_{(ik)} + z. G_{ik} \tag{2.12}$$

where, z is the learning factor which is 0.5 in the general case, and gradient $G_{k\mathrm{j}}$ can be obtained by using the partial derivative of error with respect to output weight,

$$G_{kj} = \frac{\partial E}{\partial Wo_{(kj)}} \tag{2.13}$$

Moreover, gradient G_{ik} can be obtained by partial derivative of error with respect to input weight,

$$G_{ik} = \frac{\partial E}{\partial Wo_{(ik)}} \tag{2.14}$$

A new set of network output is calculated with updated weights shown in Equation (2.11) and (2.12) for the next iteration. This procedure is then repeated for several iterations until the mean square error is minimized. In order to prevent the training from an infinite loop, stopping criteria are usually defined such as the maximum number of iterations or when an attemptable training error is reached.

2.5 Flex Sensor Glove

For the purpose of this research it is important to map human hand motions to RPH motions. This can be achieved with use of sensors which can sense human hand motions such as flex sensors, EMG (Electromyography) sensors and vision systems. This type of sensing can be implemented on a RPH by using a controller to communicate to the actuators the desired motion.

2.5.1 Flex sensor

In this research, the motion of the human posture is captured with a flex sensor. A flex sensor is an analog resistor which works as a variable analog voltage divider. The bending of flex sensor produces a change in resistance relative to the bend radius as shown in Figure 2.19 [26].

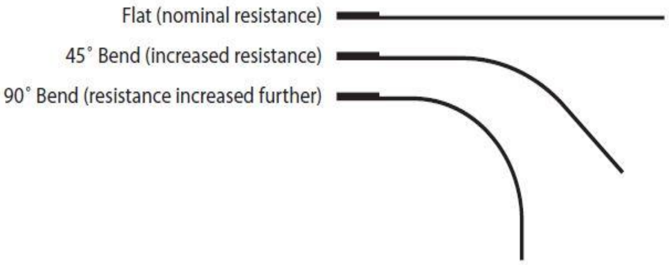

Flat (nominal resistance)
45° Bend (increased resistance)
90° Bend (resistance increased further)

Figure 2.19 Different bending conditions of flex sensor [27]

2.5.2 Implementing Flex sensors on Glove

To map human hand motions to the RPH, flex sensors were integrated on the outer surface of a glove. Five flex sensors were attached on the outside of the glove, as shown in Figure 2.20.

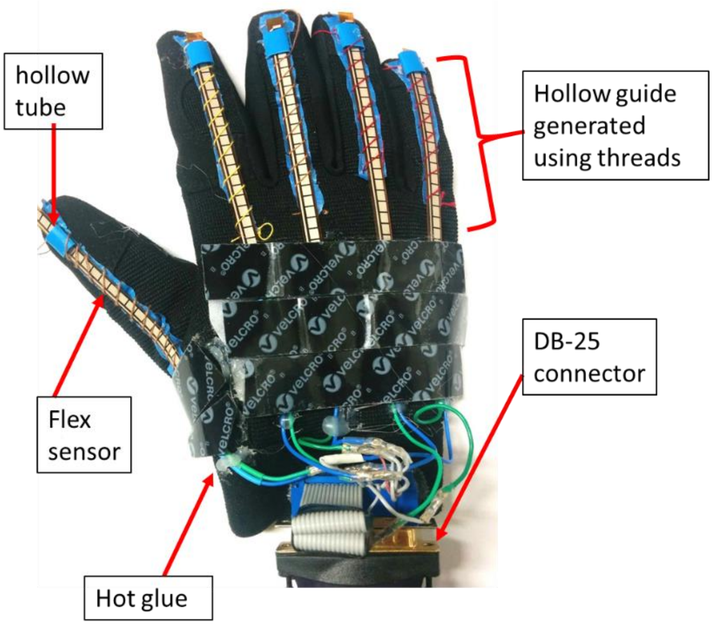


Figure 2.20 Flex sensor glove

The thread sown on to the glove was used to create a hollow gap on the glove to allow the flex sensor to slide through hollow gap smoothly. Considering the shift of flex sensor during full bending, a small hollow tube was mounted at the end of the finger to stabilize the end of flex sensor. The base of the flex sensor was kept fixed. The two wires are connected to a DB-25 connector which that connects to a flex sensor interface board and the microcontroller. A user can wear this glove and control the motion of the fingers of the prosthetic hand. The calibration procedure and flex sensor circuits are discussed in Chapter 3.

CHAPTER 3

SOFTWARE AND HARDWARE TOOLS

3.1 Hardware/Computer Interface

3.1.1 LabVIEW

LabVIEW is a programming software tool from National Instruments (NI), that allows a user to interface and control several mechanical and electrical components through a graphical programming environment. The software also allows the user to visualize information and perform computations in real-time [28]. A myRIO microcontroller from NI is used to handle inputs and outputs and all computation [29]. The myRIO microcontroller has a high clock speed, good memory, wireless connectivity, audio input-output and camera input. Additionally, LabVIEW has inbuilt myRIO module which simplifies the interface between software and hardware. LabVIEW has the resources for graphical and textual programming on a host computer toolkit and then transmitted to myRIO. LabVIEW also has its own machine learning toolkit which allows the user to perform machine learning in real-time. Figure 3.1 illustrates relationship between hardware and software tools used for this research.

The connection diagram shows that host computer uses NI LabVIEW software to select control modalities such as, manual, flex sensor glove and artificial neural network. LabVIEW software communicates with myRIO using Wi-Fi. The myRIO is then connected to three different interfacing boards: the servo, the flex sensor and the force feedback.

The servo motor interfacing and force feedback boards are each connected to the RPH with a DB-25 cable and the flex sensor sensing board is connected to the glove with DB-25 cable. The extra terminals can be used in the future to connect more sensors on the glove and the RPH. The external power supply of 12V and 5V is provided to myRIO and the servo motor interfacing board. However, the remaining two boards use 5V power supply from myRIO.

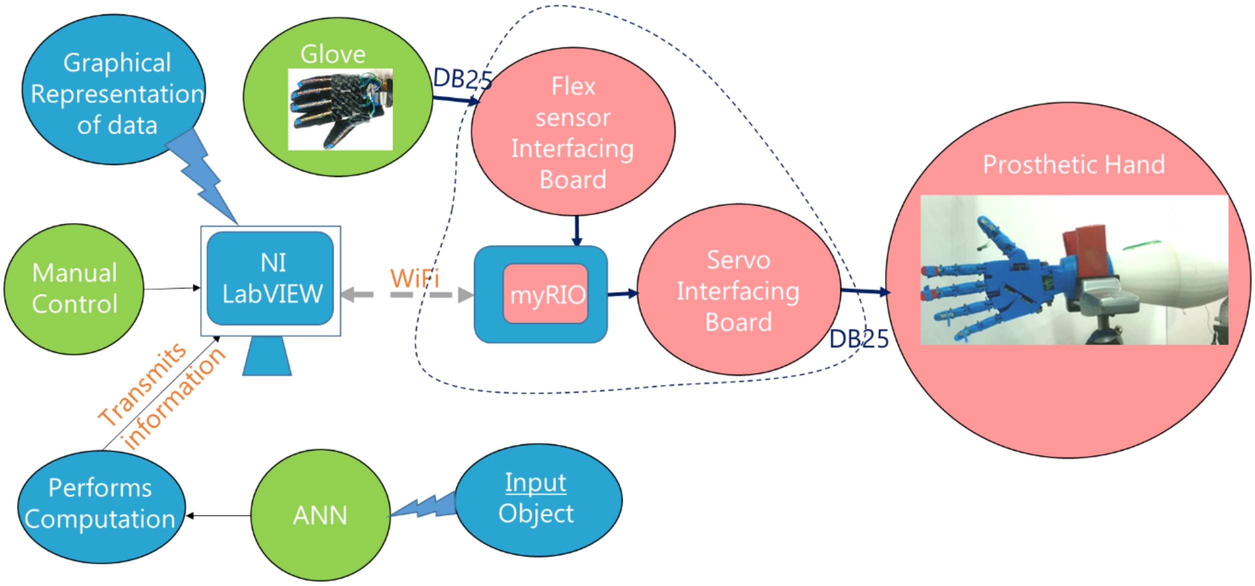


Figure 3.1 Relationship between hardware and software component

3.1.2 Servo Motor Interfacing Board

The servo motor interfacing board receives PWM signals from the myRIO and transmits those signals to five servo motors connected to the prosthetic hand. More information about controlling servo motors with PWM will be discussed in Section 3.2. Additionally, this board allows the user to adjust the voltage provided to the servo motors depending on user requirements. This board was developed using parallel circuit theory presented in [30] where the voltage applied to all motors is constant by connecting all motor positive and negative terminals in parallel. The external power supply provides the motors with 5 volts and 2 amps. The power supply is connected to the controlling board which maintains a constant 5 volts. The maximum current on each servo motor when all motors operate at the same time is 400mA. This current is appropriate to rotate a single servo motor.

3.1.3 Flex Sensor Interfacing Board

The flex sensor interfacing board receives signals from the glove and transmits them to the myRIO. More information about converting these signals to servo motor angles will be discussed in Section 3.3. The interfacing board consist of two parts, a voltage divider, amplification and filtering as shown in Figure 3.2. The voltage divider circuit is implemented on the board to adjust the sensitivity of the change in resistance so that this signal can be further proceeded through amplification and filtering.

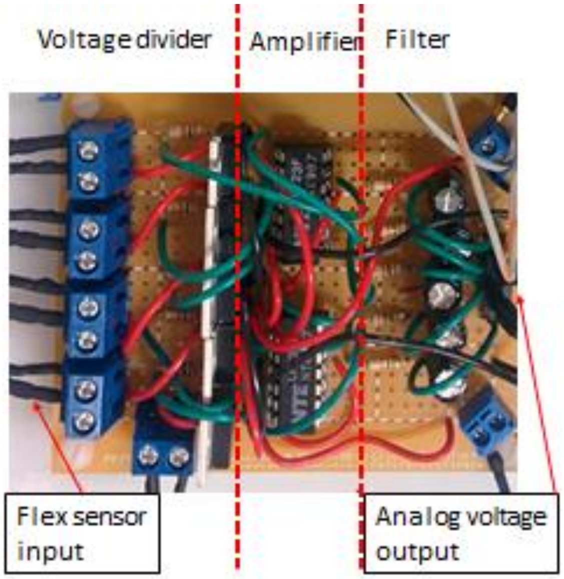


Figure 3.2 Flex sensor control board

The voltage divider circuit is a common resistor circuit which divides the voltage by maintaining a certain fixed value of resistor which can be useful in reading analog signals in terms of voltage. The voltage divider circuit can be implemented for glove sensing, by setting $V_{in}(5V)$, $R_1(3.3K\Omega)$ constant and replacing resistance R_2 with ranging flex sensor resistance as shown in Figure 3.3. The output voltage V_{FS} is calculated with Equation (3.1),

$$V_{FS} = V_{in} \frac{R_{FS}}{R_{FS}+R_1} \tag{3.1}$$

A low pass filter is added to the above circuit for noise cancelation as shown in Figure 3.3. Amplifier NTE948, suggested by manufacturer of flex sensor, is used in this circuit. The single sided operational amplifier attached to voltage output allows only low bias current to pass, which reduces error due to source impedance [27]. Also, the analog output with higher frequency can be filtered by a low pass filter as shown in Figure 3.3. The output voltage V_{out} after filtering and amplifying is shown in Equation (3.2).

$$V_{out} = V_{FS} \frac{\sqrt{R_{LPF}^2+X_C^2}}{Xc} \ where, X_c = \frac{1}{2\pi f_c C_{LPF}} \tag{3.2}$$

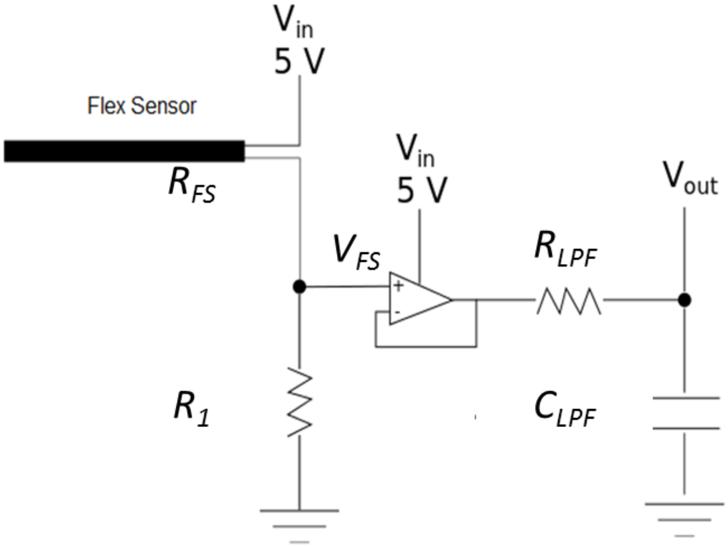


Figure 3.3 Voltage divider with filter diagram.

The cut-off frequency can be found from the resistor and capacitor of the circuit.

$$f_c = \frac{1}{2\pi R_{LPF} C_{LPF}} \tag{3.3}$$

where, R_{LPF} = Resistance of RC-Filter and C_{LPF} = Capacitance of RC-Filter. From Equation (3.3), the cut off frequency for low pass filter is *0.17Hz* with resistance of $20K\Omega$ and capacitance of $40\mu F$ as recommended by the flex sensor manufacturer. The actual analog output voltage (without amplifier and filter) and amplified and filtered analog output can be analyzed for one flex sensor for flat and full bent conditions. Figure 3.4 illustrates these two conditions where filter signal converges smoothly in direction of flat to fully bent.

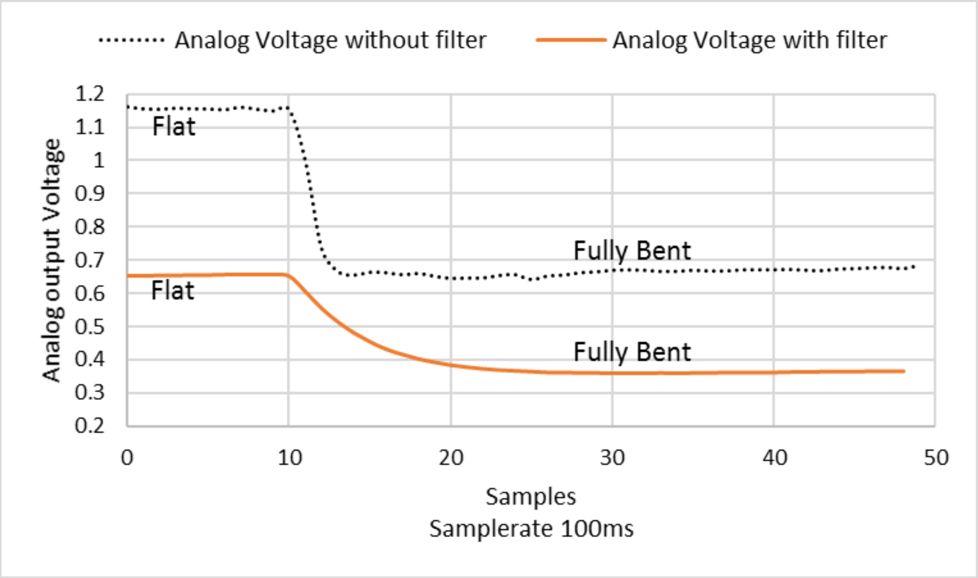


Figure 3.4 Flex sensor behavior with filter and without filter

Implementing a low pass filter increases response time and decreases noise in signal. Figure 3.5 illustrates two responses where flex sensor (flat→bent→flat) actual response rate was 500ms. It is observed in Figure 3.5 that the filtered signal had a faster response rate than the non-filtered signal.

The flex sensor interfacing board was designed and fabricated based on above specifications. All five flex sensors are connected in parallel connection to maintain constant voltage across each one. A 7805 voltage regulator is used to provide constant 5V across each voltage divider.

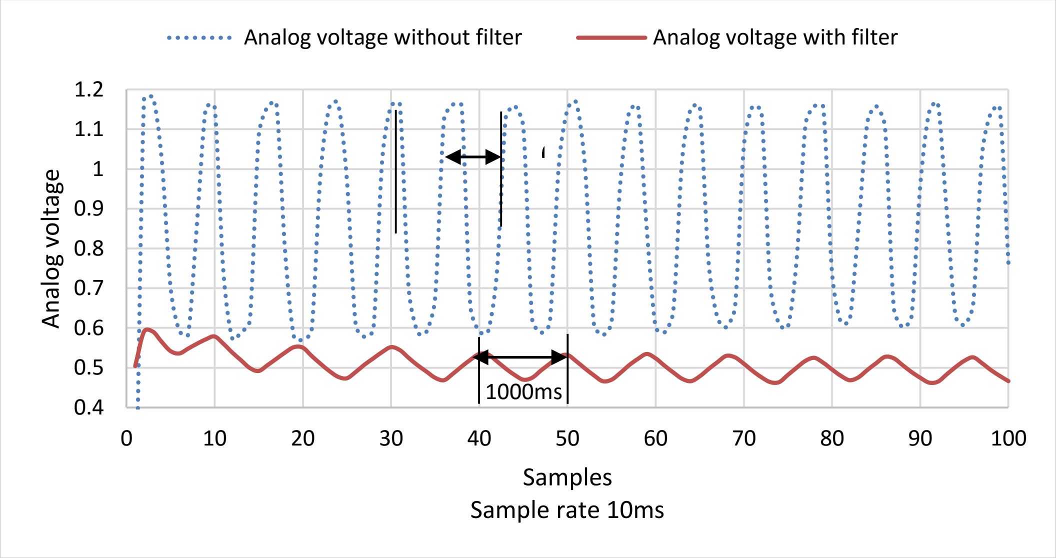


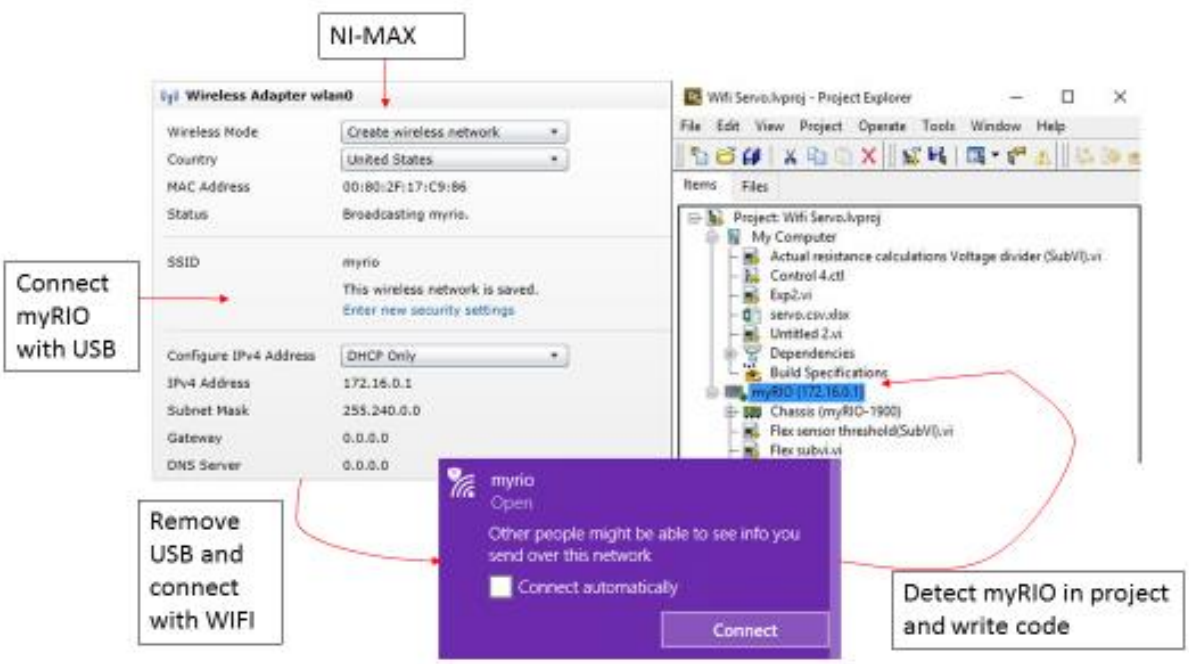
Figure 3.5 Filtered and non-filtered analog voltage response time analysis.

3.1.4 Force Feedback Interfacing Board

Force feedback interfacing board receives analog signals from the force sensor attached to the prosthetic hand and transmits them to the myRIO as analog voltage. This interface consists of five voltage dividers (one for each sensor) connected in parallel to transmit the signals from five flex sensors to microcontroller for further computation. The operating procedure is similar to flex sensor circuit. The resistance in voltage divider is adjustable and allows one to increase or decrease its sensitivity.

3.1.5 Communication with Wi-Fi and Spreadsheet.

Another major advantage of the myRIO controller is its onboard Wi-Fi which establishes wireless communication between the host computer and the controller. This eliminates the use of a tethered USB cable connection which allows the host computer to control the prosthetic hand from anywhere within the Wi-Fi range of up to 150m. The host computer running LabVIEW has the ability to deploy and run the program through Wi-Fi. In the preliminary stage, the user has to connect the myRIO to the host computer with a USB cable which detects the controller in the NI Measurement and Automation Explorer program. This procedure allows the user to setup the Wi-Fi on the controller by configuring the IP address and setting up a Wi-Fi name. Afterwards, the USB cable is removed and a connection is established to the myRIO through Wi-Fi as shown in Figure 3.6. Once the Wi-Fi connection is established, the LabVIEW project can detect the myRIO through a wireless network and the user can deploy this project.


NI-MAX
Connect myRIO with USB
Remove USB and connect with WIFI
Detect myRIO in project and write code
File Edit View Project Operate Tools Window Help
Items Files

Wireless Adapter wlan0

Wireless Mode: Create wireless network
Country: United States
MAC Address: 00:80:2F:17:C9:86
Status: Broadcasting myrio.

SSID: myrio
This wireless network is saved.
Enter new security settings

Configure IPv4 Address: DHCP Only
IPv4 Address: 172.16.0.1
Subnet Mask: 255.240.0.0
Gateway: 0.0.0.0
DNS Server: 0.0.0.0

Wifi Servo.lvproj - Project Explorer

- Project: Wifi Servo.lvproj
  - My Computer
    - Actual resistance calculations Voltage divider (SubVI).vi
    - Control 4.ctl
    - Exp2.vi
    - servo.csv.xlsx
    - Untitled 2.vi
    - Dependencies
    - Build Specifications
  - myRIO (172.16.0.1)
    - Chassis (myRIO-1900)
    - Flex sensor threshold(SubVI).vi
    - Flex subvi.vi

myrio
Open
Other people might be able to see info you send over this network
□ Connect automatically
Connect

Figure 3.6 Establishing Wi-Fi connection on myRIO using computer tools.

LabVIEW provides flexible interface with computer data files. For example, LabVIEW can read data from a spreadsheet and process it, and again write data back to the spreadsheet file. Figure 3.7, shows the Read Delimited Spreadsheet VI available in LabVIEW that was used in this work.

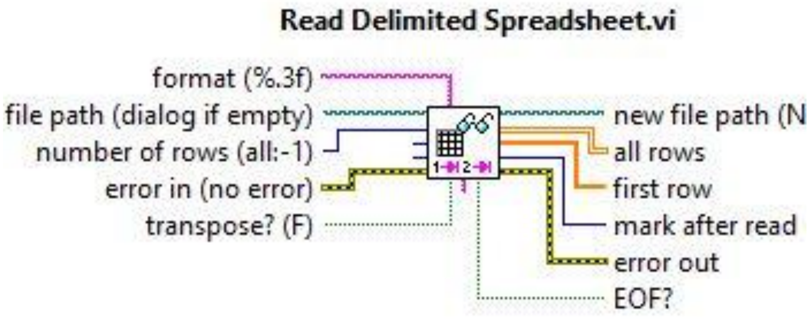


Figure 3.7 Read Delimited Spreadsheet VI in LabVIEW

3.2 Servo Motor Control

3.2.1 Pulse Width Modulation

As discussed in Chapter 2, servo motors are used to control the fingers of the RPH. However, the servo motor rotation has to be controlled accurately and precisely for gasping requirements. Servo motors operate with Pulse Width Modulation (PWM) signals which require two parameters, duty cycle and frequency. PWM produces repetitive pulse signals with desired duty cycle as shown in Figure 3.8.

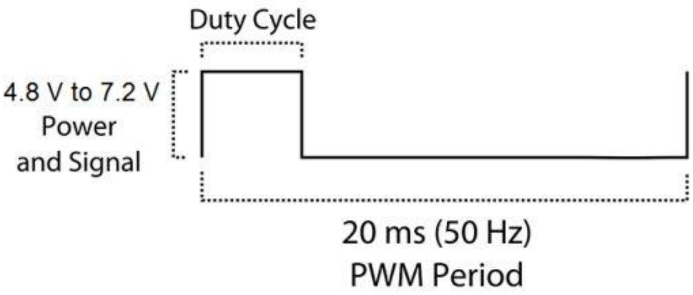


Figure 3.8 Pulse width modulation signal

However, duty cycle and frequency are different depending on the type of servo motor used. The servo motor used in the current prosthetic hand is MG995 manufactured by Tower Pro which operates in the power range of 4.8V to 7.2V [31]. Most servo motors operate with duty cycle range of 10% (pulse width 1ms) to 20% (pulse width 2ms) where 1ms is equal to 0 degrees, 1.5ms is equal to 90 degrees and 2ms is equal to 180 degrees, for 20ms cycle time. The MG995 servo motor has 20ms cycle time (50Hz). However, it operates between duty cycle range of 5% (0.5ms) to 20% (2ms) where 0.5ms equals 0 degrees, 1.25ms equals 90 degrees and 2ms equals 180 degrees. This operation was determined experimentally.

3.2.2 Servo Motor Angle to PWM calibration

An algorithm was developed to convert servo motor angle into duty cycle according to desired user input angle. The duty cycle can be obtained by multiplying pulse width with frequency. Therefore, there is a linear relation between pulse width and servo motor angle. This algorithm is based on servo motor angle span of 0 to 180 degrees. The calibration is based on linear relationship according to Equation (3.4):

$$P = mA + C \tag{3.4}$$

where, A = desired servo motor angle in degrees, P = output pulse width in millisecond, C = initial duty cycle and m =slope. The slope m of equation (3.4) is

$$m = \frac{P_f - P_i}{A_f - A_i} = \frac{2-0.5}{180-0} = 0.0083$$

where, A_f = final servo motor angle, A_i = initial servo motor angle, P_f = pulse width for final servo motor angle and P_i = pulse width for initial servo motor angle. Solving Equation (3.4) with slope equal to 0.0083, a linear equation $P = 0.0083A + 0.5$ can be obtained. Figure 3.9 shows the relation between pulse width and servo motor angle.

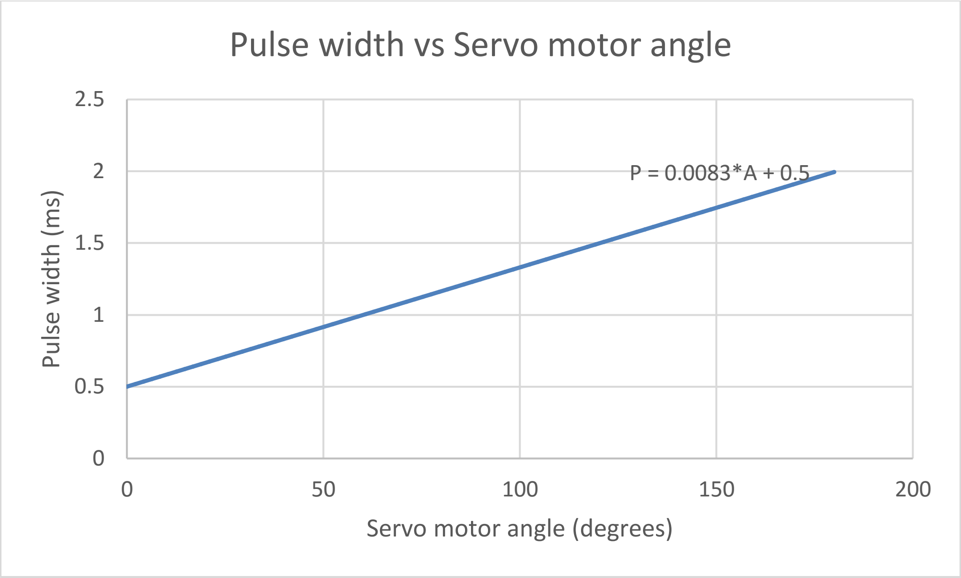
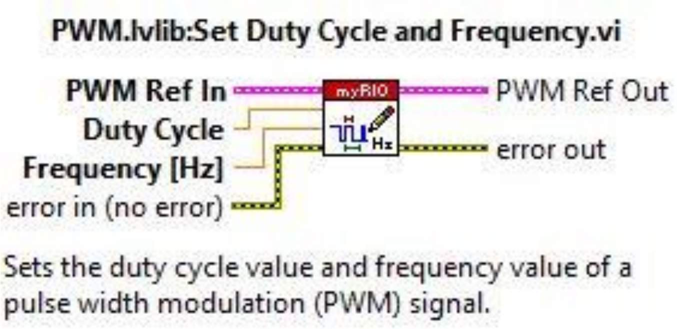


Figure 3.9 Pulse width versus servo motor angle

The LabVIEW program requires duty cycle (seconds) and frequency (Hz) to generate the PWM signal to the servo motor as shown in Figure 3.10.



PWM.lvlib:Set Duty Cycle and Frequency.vi

Sets the duty cycle value and frequency value of a pulse width modulation (PWM) signal.

Figure 3.10 Express VI to generate PWM signal in LabVIEW

A sub-VI was developed to compute the duty cycle according to Equation (3.3) and it is represented in Figure 3.11. Note that the user must specify the constant parameters such as *m*, *C* and frequency.

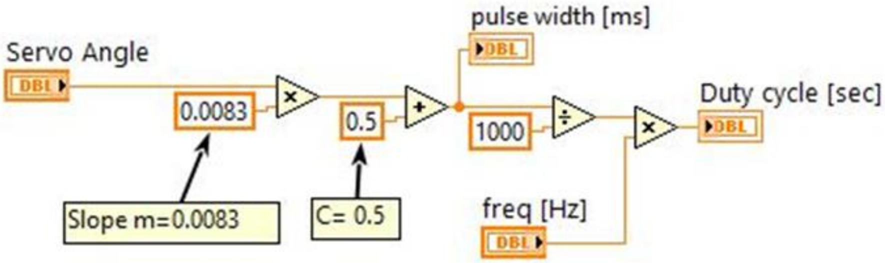


Figure 3.11 Servo motor angle to duty cycle Sub VI

3.3 Flex Sensor Read

3.3.1 Analog Voltage to Servo Motor Angle Conversion Algorithm

An algorithm was developed to convert the analog voltage output V_{out} from the flex sensor into servo motor angle depending on maximum rotational limit (final servo motor angle) of the servo motor. The linear relation between analog voltage output and servo motor angle was developed based on flat and fully bent conditions of the flex sensor as shown in Equation (3.5).

$$Output\ Angle = \frac{(U-Vout)}{(U-L)} \times Final\ servo\ motor\ angle \tag{3.5}$$

where, U is the analog voltage output of flex sensor in flat condition and L is the analog voltage output of flex sensor in fully bent condition. Note that the final servo motor angle is different for each finger. This relationship also allows the user to define the highest and lowest values of analog output voltage such that the servo motor work in the region between flat and fully bent conditions. A conditional flowchart was created to prevent malfunction of the servo motor and shown in Figure 3.12.

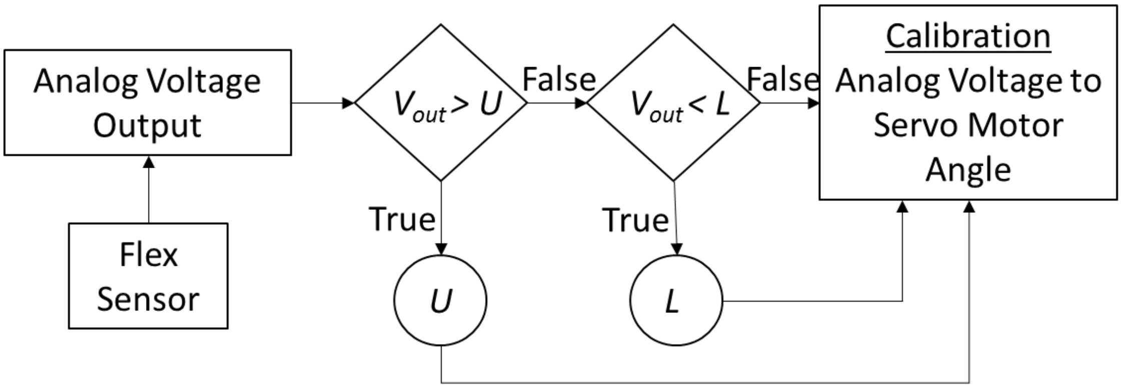


Figure 3.12 Conditional flow chart to limit analog voltage between upper and lower bound.

Even if the analog output voltage is higher than U and lower than L, the calibrated output angle must not be lower than the initial condition and higher than the final condition to avoid damaging the servo motor. The sub-VI developed in LabVIEW that implements the flex sensor bent to motor angular rotation as shown in Figure 3.13.

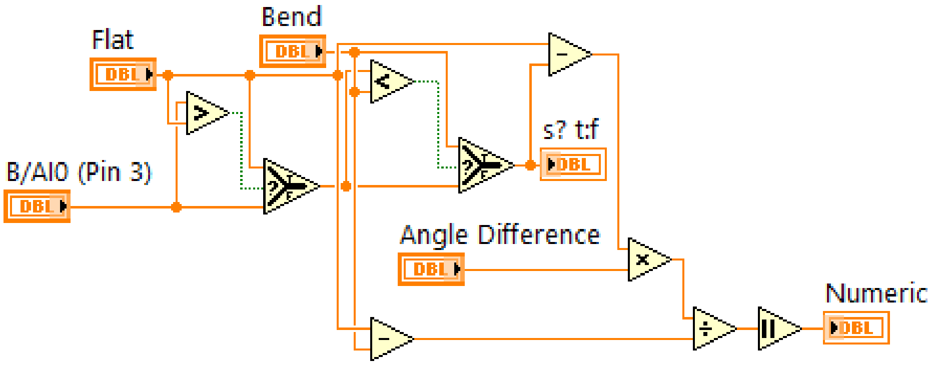


Figure 3.13 Sub VI to calibrate analog voltage to servo motor angle.

3.3.2 Calibration

An algorithm was developed to calibrate the change in analog voltage for the conditions of being totally flat and maximum bent position of flex sensor mounted on glove for home and final position of the servo motor.

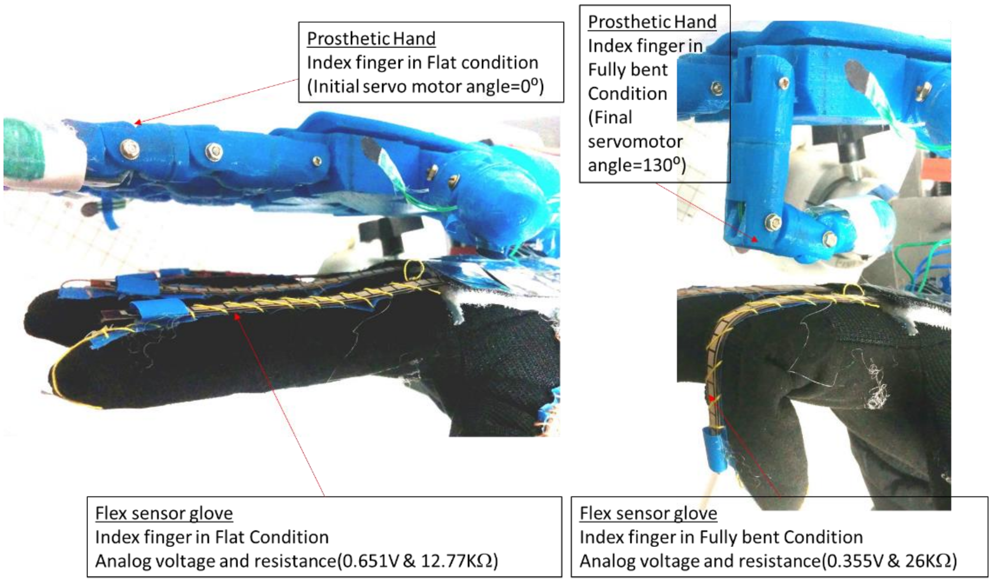

Prosthetic Hand
Index finger in Flat condition
(Initial servo motor angle=0^{o})
Prosthetic Hand
Index finger in
Fully bent
Condition
(Final
servomotor
angle=130^{o})
Flex sensor glove
Index finger in Flat Condition
Analog voltage and resistance(0.651V & 12.77KΩ)
Flex sensor glove
Index finger in Fully bent Condition
Analog voltage and resistance(0.355V & 26KΩ)

Figure 3.14 Flex sensor glove and RPH mapping for flat and fully bent condition in index finger.

Figure 3.14 shows the flex sensor in a flat and fully bent condition for the index finger and also shows the initial and final position of the RPH with desired servo motor rotation.

Table 3.1 Relation between servo motor angles and flex sensor analog voltage and resistance

Prosthetic Hand	**Servo motor angle (degrees)**		**Flex Sensor Analog Output Voltage(V)**		**Flex Sensor Analog Output Resistance(kΩ)**	
	Initial	**Final**	**Flat**	**Fully Bent**	**Flat**	**Fully Bent**
Thumb	0	90	0.651	0.355	12.776	26.000
Index	0	130	0.651	0.421	12.776	21.157
Middle	0	175	0.617	0.317	13.632	29.717
Ring	0	130	0.634	0.330	13.208	28.378
Pinky	0	90	0.552	0.252	15.650	37.972

Note that the initial and final position for servo motor is related to the open and close position of each finger and thumb which is different for each finger due to RPH construction.

Therefore, experimental evaluation of each finger and thumb is necessary for different flex sensor bent conditions. The data for home and final position for each servo motor and flex sensor has been collected empirically and shown in Table 3.1

The analog output voltage of the flex sensor is calculated in its flat position and maximum bent positions. Then, these two values were correlated with the finger and thumb's open and close positions. Therefore, the user can control each servo motor with each flex sensor. Table 3.1 shows that 0 degree corresponds to 0.651 V and 90 degrees corresponds to 0.355 V for the thumb which indicates that an inversely proportional relationship exist between servo motor angle and analog voltage. This relationship was developed for each finger. However, it is proportional to analog resistance from the flex sensor. The difference for fully bent and flat condition for analog output voltage and servo motor angle is shown in Table 3.2.

Table 3.2 Calibration from flex sensor to servo motor

Prosthetic Hand	**Servo motor angle difference**	**Flex Sensor Analog Output Voltage Difference**	**Calibration** **Flat (initial angle of servo motor)-voltage difference=Fully bent (final angle of servo motor)**
	Final-Zero	**Flat-Fully Bent**	
Thumb	90	0.296	0.651(0º)-0.296=0.355(90º)
Index	130	0.230	0.651(0º)-0.230=0.421(130º)
Middle	175	0.300	0.617(0º)-0.300=0.317(175º)
Ring	130	0.304	0.634(0º)-0.304=0.300(130º)
Pinky	90	0.300	0.552(0º)-0.252=0.300(90º)

While calibrating, analog voltage values have been taken at a lower value for the flat flex sensor condition and a higher value for a fully bent flex sensor condition. The actual values are calculated with fully flat and fully bent conditions of the human hand which will be not similar every time. This motivates the use of lower calibrated values than actual values. Table 3.3 shows the calibrated analog voltage versus the actual analog voltage for flat and fully bent conditions. However, these values will vary depending on the user.

Table 3.3 Actual versus calibrated analog voltage bounds for flex sensor

Prosthetic Hand	**Calibrated Flex Sensor Analog Output Voltage(V)**		**Actual Flex Sensor Analog Output Voltage(V)**	
	Flat	**Fully Bend**	**Flat**	**Fully Bend**
Thumb	0.620	0.360	0.651	0.355
Index	0.620	0.440	0.651	0.421
Middle	0.580	0.300	0.617	0.317
Ring	0.620	0.400	0.634	0.330
Pinky	0.520	0.300	0.552	0.252

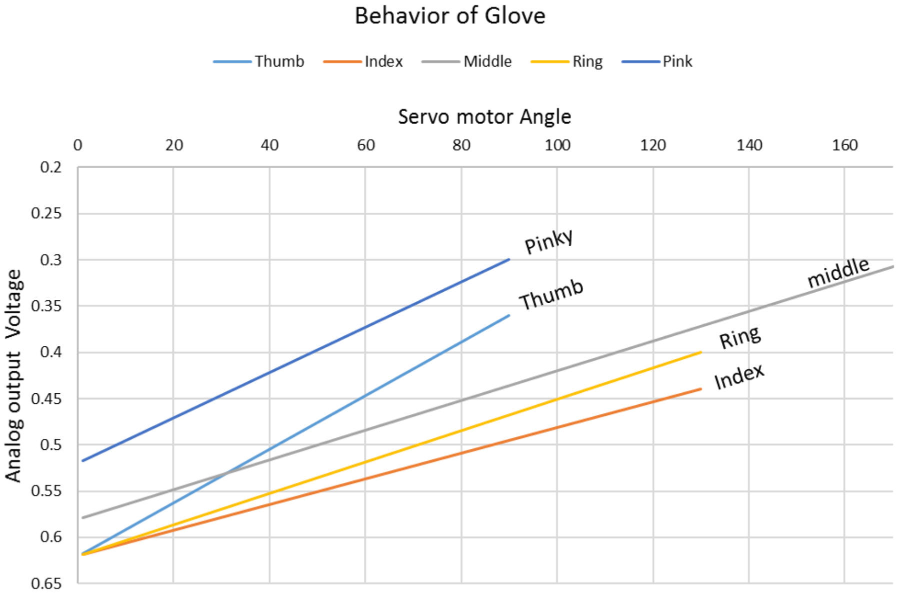


Figure 3.15 Servo motor angle vs glove flex sensor output graph

The linear relationship between the flex sensor calibrated analog voltage and servo motor angle for all fingers and thumb is shown in Figure 3.15 which illustrates that the behavior of the index and ring finger is similar. Moreover, the behavior of the thumb and pinky finger is similar. However, the behavior of middle finger is different. This behavior depends on physiology, geometry and actuation mechanism of prosthetic hand. Moreover, the relationship between the flex sensor's analog resistance and servo motor angle can be modified by recording the glove movements with RPH movements.

CHAPTER 4

GRASP LEARNING IN PROSTHETIC HAND

The previous discussion was focused on creating a versatile platform, to perform object grasping tasks using software and hardware tools. The goal of this thesis is to implement decision making in a prosthetic hand to grasp different geometrical objects based on their geometrical properties and location in three dimensional space. This procedure is divided in two stages. In the first stage, the knowledge data collected consists of various objects and desired grasping patterns, which are generated manually in order to implement supervised learning on a prosthetic hand. In the second stage, the data is passed through the back propagation learning algorithm which approximates nonlinear functions (weights) representing a relationship between the object location and geometry (input) and the grasping pattern (output) of the RPH. In evaluation part, this nonlinear relationship is used to approximate output based on input. These two stages are integrated in the LabVIEW GUI developed for this research. The first stage through manual or glove control generates the knowledge data set and the second stage has an automatic and artificial neural network control to evaluate data learning by predicting grasping pattern.

The first and second stages are implemented in the LabVIEW block diagram program using an event structure and global variables. The event structure is a conditional structure that waits until the next event occurs to perform selected task defined by user [28]. Event structure is used to handle multiple tasks in LabVIEW. Global variables are generally used to pass data among several different VIs thus eliminating multiple wire connection for complex block diagrams.

4.1 Generating Knowledge Data for Prosthetic Hand

The user has two possible controls to generate knowledge data using the developed software and hardware tools according to flow chart shown in Figure 4.1. Manual control and glove control are two options available in the LabVIEW GUI to control the RPH.

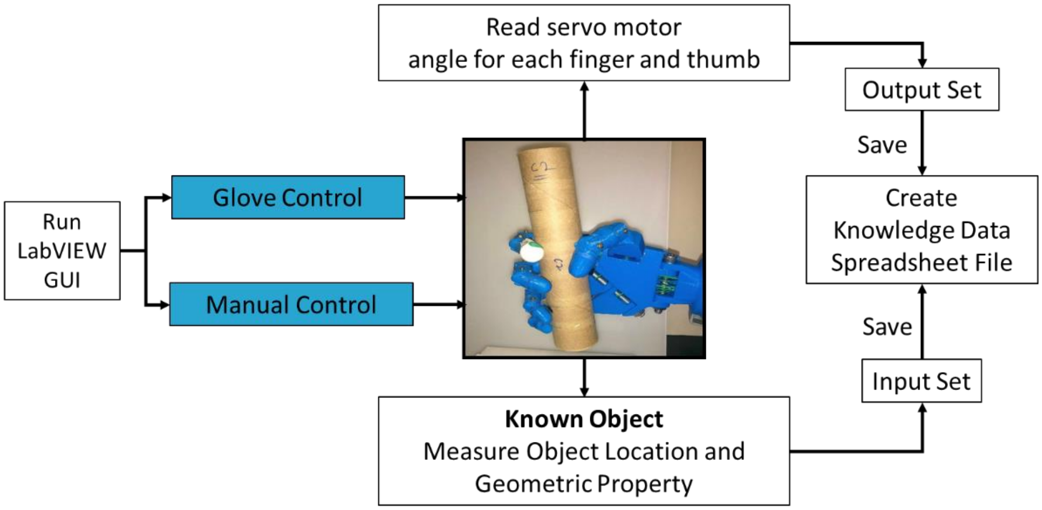
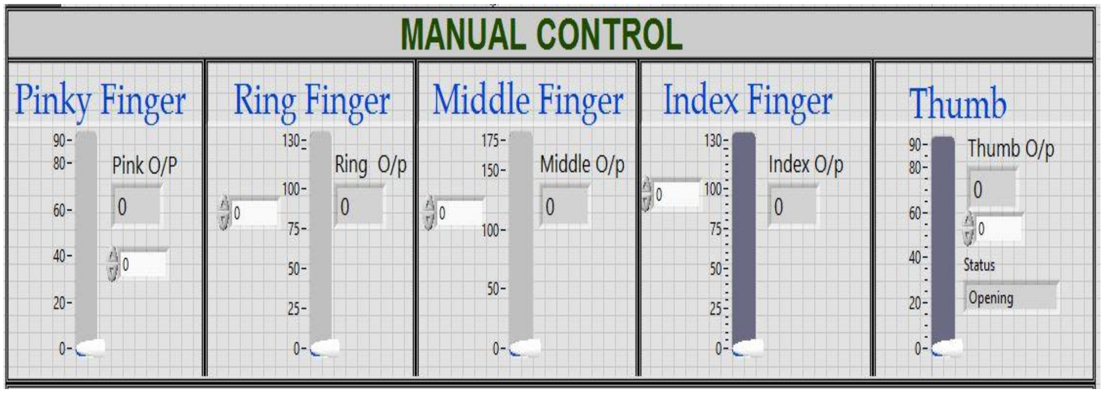

Read servo motor angle for each finger and thumb
Output Set
Save
Create Knowledge Data Spreadsheet File
Run LabVIEW GUI
Glove Control
Manual Control
Save
Input Set
Known Object
Measure Object Location and Geometric Property

Figure 4.1 Flow chart to generate knowledge data in RPH.

The function of manual control is to control each servo motor connected to fingers and thumb using a vertical slider (having the user control the slider with the mouse) as shown in Figure 4.2. The vertical slider is connected at the end of the servo motor calibration algorithm to perform this operation. Additionally, numerical control is available to manipulate the prosthetic hand with a desired servo motor angle.



MANUAL CONTROL

Pinky Finger
Pink O/P 0
0

Ring Finger
Ring O/p 0
0

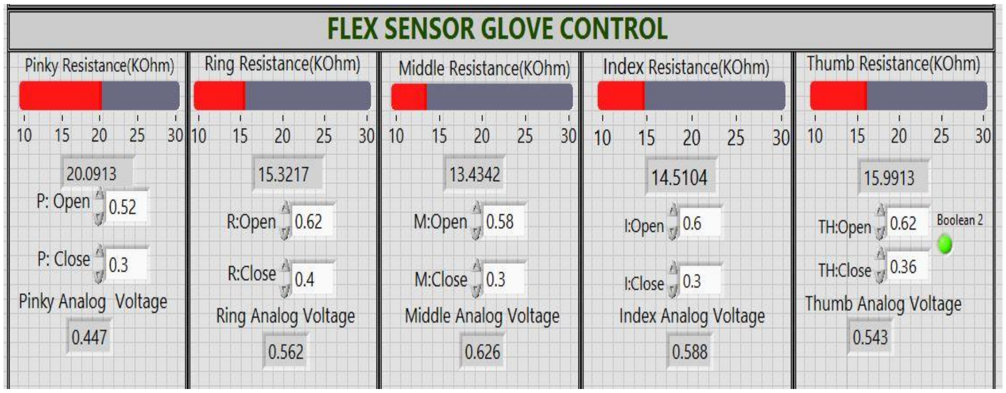
Middle Finger
Middle O/p 0
0

Index Finger
Index O/p 0
0

Thumb
Thumb O/p 0
0
Status Opening

Figure 4.2 Manual control GUI

A flexible logic scheme was created in LabVIEW to switch from manual control to glove control in real-time. Glove control maps the human hand motion to the prosthetic hand as discussed in Chapters 2 and 3. In this control modality, the GUI was created to visualize analog resistance and analog voltage in the flex sensor and mapped servo motor angle in the prosthetic hand as shown in Figure 4.3.



FLEX SENSOR GLOVE CONTROL

Pinky Resistance(KOhm) 20.0913
P: Open 0.52
P: Close 0.3
Pinky Analog Voltage 0.447

Ring Resistance(KOhm) 15.3217
R:Open 0.62
R:Close 0.4
Ring Analog Voltage 0.562

Middle Resistance(KOhm) 13.4342
M:Open 0.58
M:Close 0.3
Middle Analog Voltage 0.626

Index Resistance(KOhm) 14.5104
I:Open 0.6
I:Close 0.3
Index Analog Voltage 0.588

Thumb Resistance(KOhm) 15.9913
TH:Open 0.62
TH:Close 0.36
Boolean 2
Thumb Analog Voltage 0.543

Figure 4.3 Glove control GUI

The glove control GUI has the ability to reset analog voltage bounds for open and close conditions of data glove. Grasping of an object is performed by selecting manual or glove control individually on host computer.

Object grasping is divided in two parts. In the first part, grasping is performed on different objects with similar locations with respect to the Z-axis. The objects chosen for the initial focus of this study are cylinders and spheres, as shown in Figure 4.4.

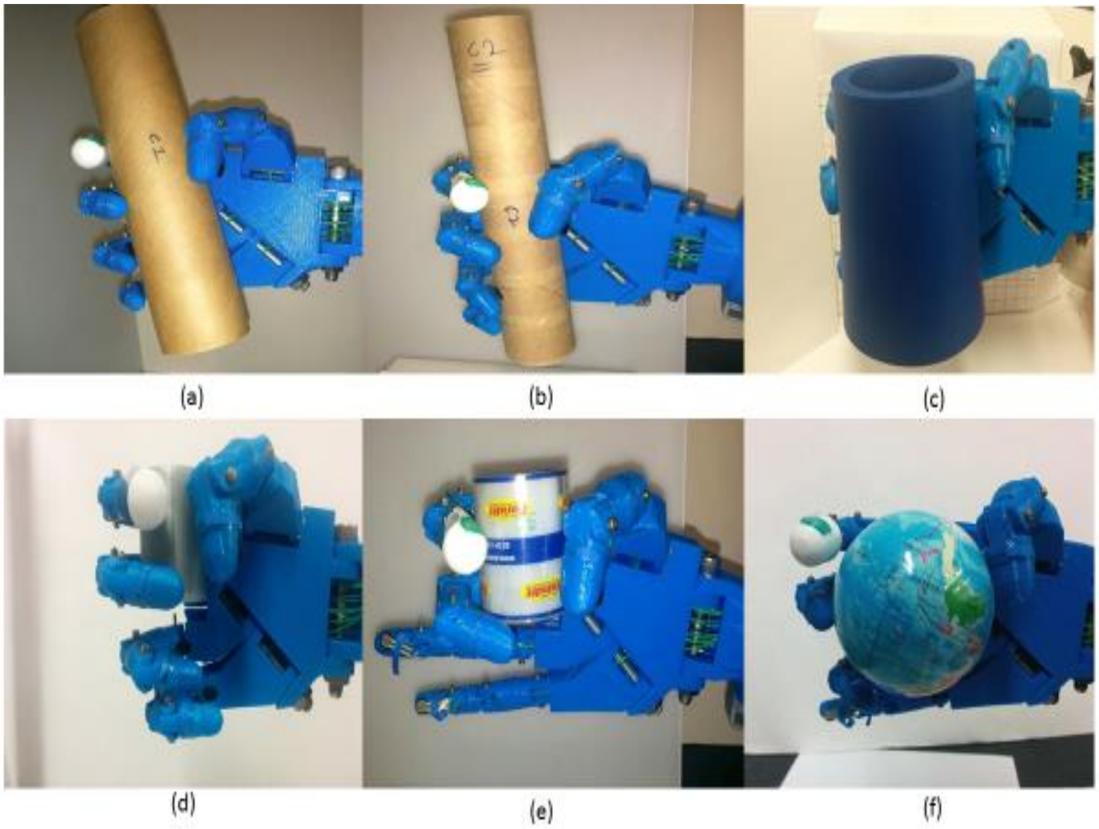


Figure 4.4 Geometrical grasping using glove and manual control.

The geometrical parameters such as diameter and height of the objects shown in Figure 4.4 are presented in Table 4.1 where ***C*** indicates cylinder and ***S*** indicates sphere.

Table 4.1 Geometrical parameters of objects used in Figure 4.4

	Characteristics	**Diameter (mm)**	**Height (mm)**
(a)	C1	51.5	197
(b)	C2	40.5	202
(c)	C3	79.1	102.3
(d)	C4	35.5	47.8
(e)	C5	49.4	51
(f)	S1	50	50

In the second part of the learning procedure, grasping is performed on the same object by changing the location of the object with respect to the Z- axis. The task is performed particularly on cylindrical objects to demonstrate the relationship between the object location and desired grasping pattern as shown in Figure 4.5.

It is observed that the grasping pattern changes for an object due to its location with reference to Z-axis. For instance, only the index finger and thumb are used to grasp the object in Figure 4.5(d) because it is not necessary to use all fingers for grasping based on the location of the object. These results show that the location of the object is an important parameter that needs to be considered to generate a proper grasping pattern.

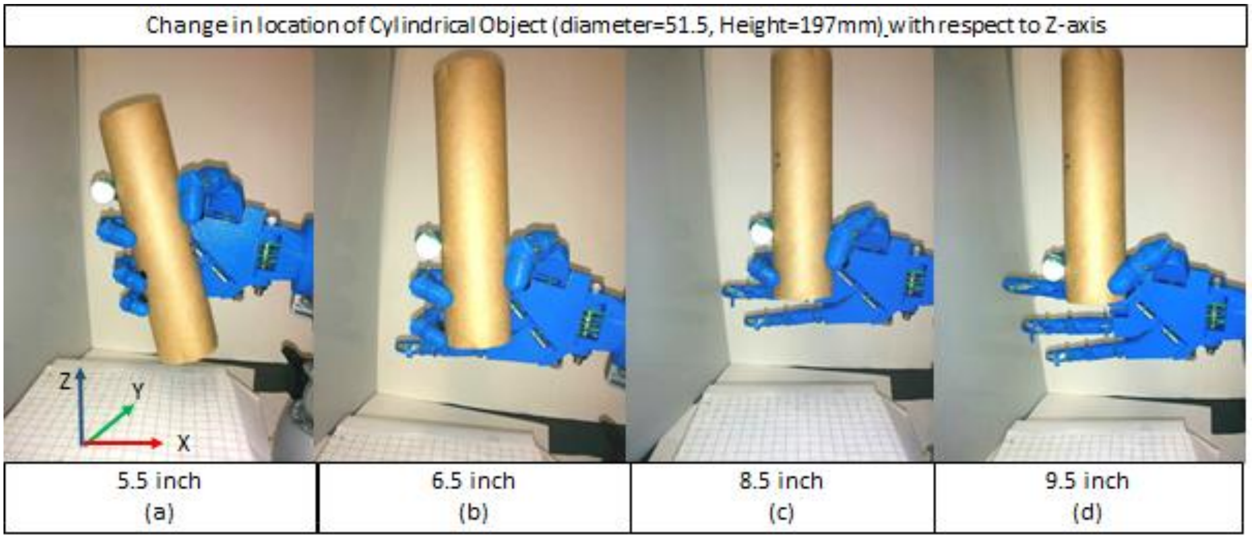

Change in location of Cylindrical Object (diameter=51.5, Height=197mm) with respect to Z-axis
Z
Y
X
5.5 inch
(a)
6.5 inch
(b)
8.5 inch
(c)
9.5 inch
(d)

Figure 4.5 Cylindrical object grasping considering various locations.

After performing the grasping task on several cylindrical and spherical objects, the geometrical data and location (input) of object and desired servo motor angles (output) were saved in a spreadsheet as shown in Appendix C. This dataset is the knowledge data for grasp learning. Two major behaviors observed during grasping experimentation.

1. Increase in object diameter decreases the servo motor angle for cylindrical objects.
2. Change of the location of object changes the resulting grasping pattern.

The main purpose of creating knowledge data is to develop these behaviors in a prosthetic hand through machine learning.

4.2 Learning and Evaluation using Back Propagation Algorithm

The previous sections explained the procedures used to create knowledge data set which is required for implementing supervised learning in the prosthetic hand. This section focuses on implementing the decision making algorithm in the prosthetic hand through learning.

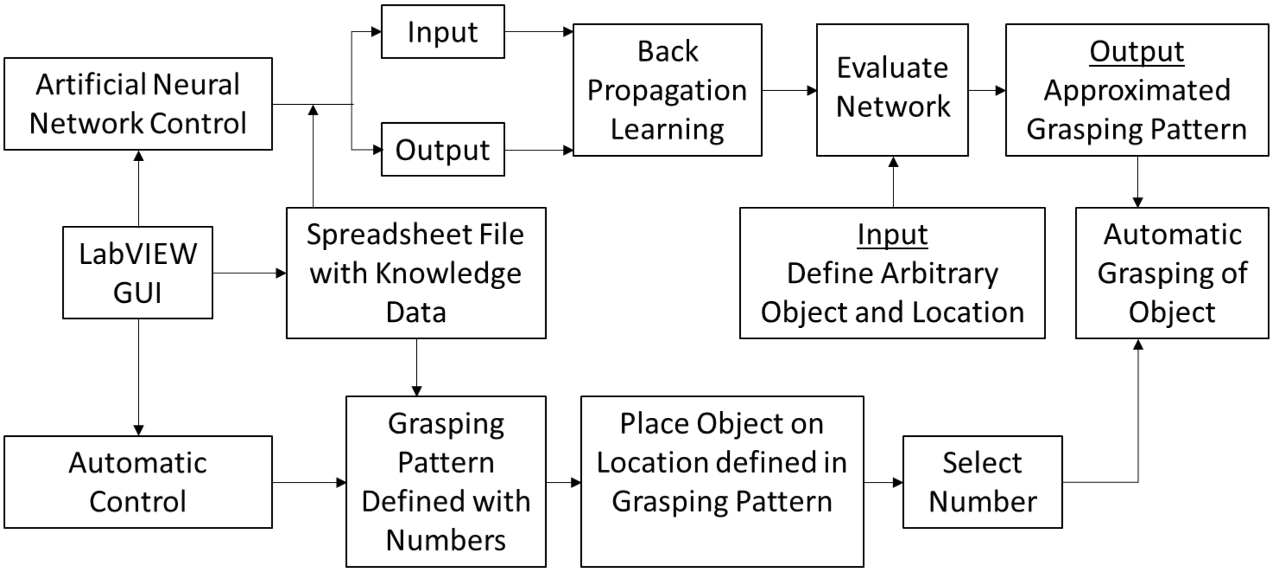

Input
Output
Artificial Neural Network Control
Back Propagation Learning
Evaluate Network
Output Approximated Grasping Pattern
LabVIEW GUI
Spreadsheet File with Knowledge Data
Input Define Arbitrary Object and Location
Automatic Grasping of Object
Automatic Control
Grasping Pattern Defined with Numbers
Place Object on Location defined in Grasping Pattern
Select Number

Figure 4.6 Flow chart to implement artificial neural network control.

Figure 4.6 shows the procedure for artificial neural network (ANN) learning and evaluating to grasp various objects using the RPH. Additionally, automatic control provides the flexibility to perform the object grasping using knowledge data by selecting object characteristic in LabVIEW GUI.

4.2.1 Knowledge Data Learning

The machine learning toolkit in LabVIEW was used to learn grasping patterns in the RPH. The spreadsheet with knowledge data is imported into LabVIEW block diagram. The knowledge data was transformed in two matrices such as input object location and geometry and output object grasping pattern as shown in Appendix C. This data contains training data sample for a three-layer back propagation (BP) learning algorithm. The back propagation learning algorithm requires a user to define the inputs, hidden neurons and output as discussed in Section 2.4.3. The inbuilt LabVIEW BP learn VI used to “learn” a grasping pattern is shown in Figure 4.7.

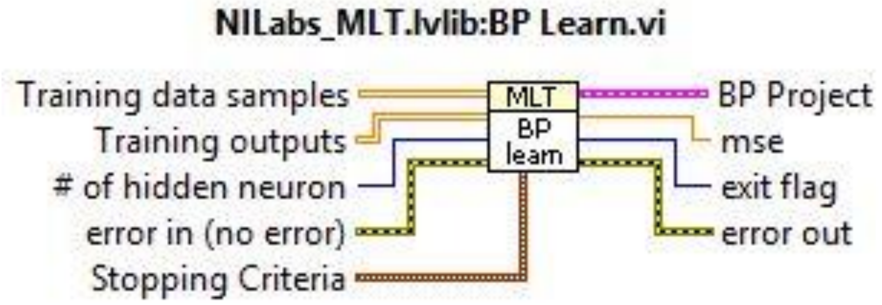
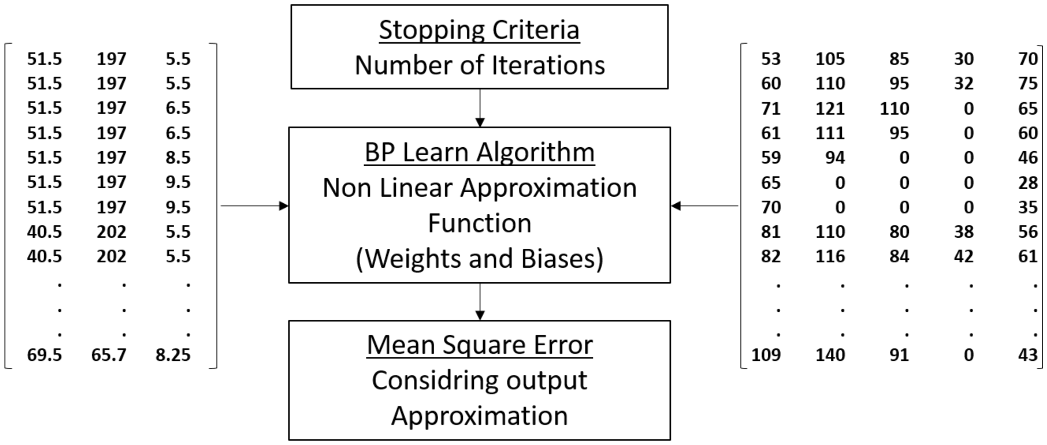


Figure 4.7 BP Learn VI description

The BP Learn VI requires training data samples (input matrix) and training outputs (desired output matrix) generated from knowledge data. It also requires, the number of hidden neurons and the stopping criteria which allows one to define the maximum number of iterations. The output of the VI is the minimum mean square error (MSE) based on the performance of artificial neural network. The BP learn VI consisting back propagation learning algorithm trains the knowledge data and calculates the mean square error for a desired number of iterations as shown in Figure 4.8.

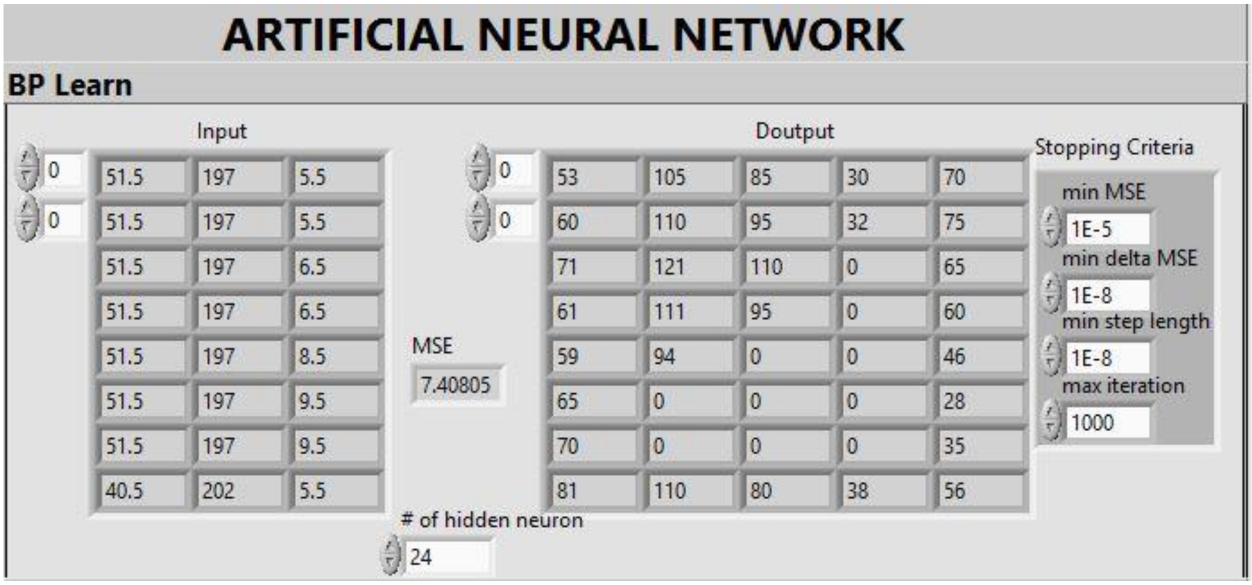


$$\begin{bmatrix} 51.5 & 197 & 5.5 \\ 51.5 & 197 & 5.5 \\ 51.5 & 197 & 6.5 \\ 51.5 & 197 & 6.5 \\ 51.5 & 197 & 8.5 \\ 51.5 & 197 & 9.5 \\ 51.5 & 197 & 9.5 \\ 40.5 & 202 & 5.5 \\ 40.5 & 202 & 5.5 \\ \vdots & \vdots & \vdots \\ 69.5 & 65.7 & 8.25 \end{bmatrix}$$

$$\begin{bmatrix} 53 & 105 & 85 & 30 & 70 \\ 60 & 110 & 95 & 32 & 75 \\ 71 & 121 & 110 & 0 & 65 \\ 61 & 111 & 95 & 0 & 60 \\ 59 & 94 & 0 & 0 & 46 \\ 65 & 0 & 0 & 0 & 28 \\ 70 & 0 & 0 & 0 & 35 \\ 81 & 110 & 80 & 38 & 56 \\ 82 & 116 & 84 & 42 & 61 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 109 & 140 & 91 & 0 & 43 \end{bmatrix}$$

Figure 4.8 Knowledge data learning consisting input, output, stopping criteria and MSE

The graphical representation of back propagation learning is developed as shown in Figure 4.9. The GUI also represents input data set, output data set and MSE using indicators. Moreover, it also provides facility to define maximum number of iterations and number of hidden neurons.



ARTIFICIAL NEURAL NETWORK

BP Learn

Input

0
0

51.5	197	5.5
51.5	197	5.5
51.5	197	6.5
51.5	197	6.5
51.5	197	8.5
51.5	197	9.5
51.5	197	9.5
40.5	202	5.5

MSE
7.40805

Doutput

0
0

53	105	85	30	70
60	110	95	32	75
71	121	110	0	65
61	111	95	0	60
59	94	0	0	46
65	0	0	0	28
70	0	0	0	35
81	110	80	38	56

Stopping Criteria

min MSE
1E-5
min delta MSE
1E-8
min step length
1E-8
max iteration
1000

# of hidden neuron
24

Figure 4.9 Back propagation learning GUI

There are two possible approaches to increase the performance of the back propagation learning (minimize the MSE). In first approach, the behavior of MSE can be checked by increasing the number of hidden neurons by keeping maximum number of iterations constant. Figure 4.10 shows that with increase in the number of hidden neurons, the MSE decreases drastically in the beginning and afterwards there is no significant change for 28 samples and 1000 iterations. Likewise, the relationship between the MSE and number of hidden neurons is a decreasing exponential in the beginning, however, afterwards there is a first order relationship. This concludes that there is no remarkable change in MSE after increasing more than 5 hidden neurons as shown in Figure 4.10

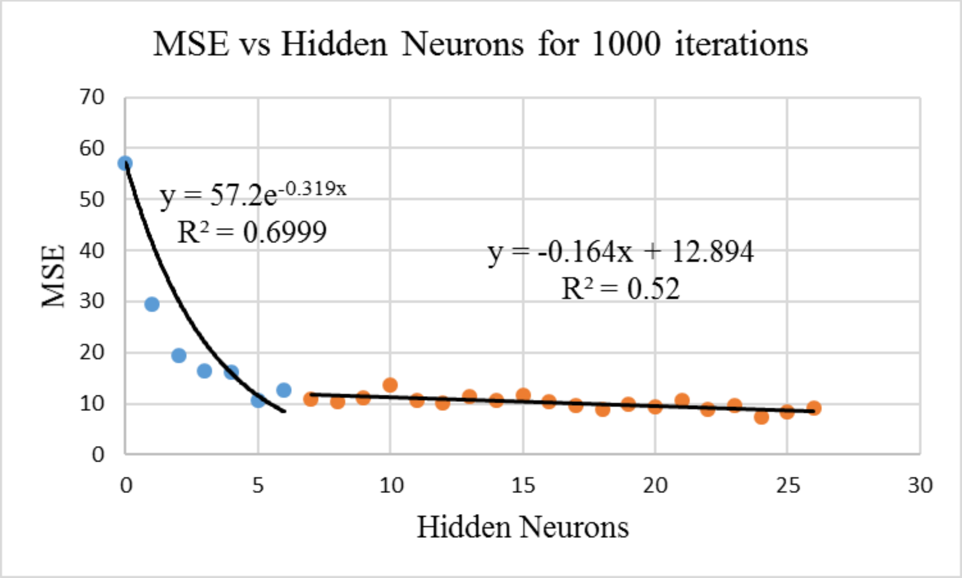


Figure 4.10 MSE versus hidden neurons graph for 28 patterns

The convergence behavior of MSE error depends on the number of patterns in the knowledge data. Figure 4.10 shows the behavior for 28 patterns available in knowledge data. It is observed that the MSE is 7.4 for 24 hidden neurons which is lowest when 26 hidden neurons are used in experiment. In second approach, the MSE decreases by increasing the number of iterations and keeping the number of hidden neurons constant (24 Hidden neurons) as shown in Figure 4.11. Similarly, in the beginning, there is a decreasing exponential relationship between the MSE and the number of iterations. However, after 1000 iterations the behavior becomes linear and there is no significant change in MSE if the number of iterations increases.

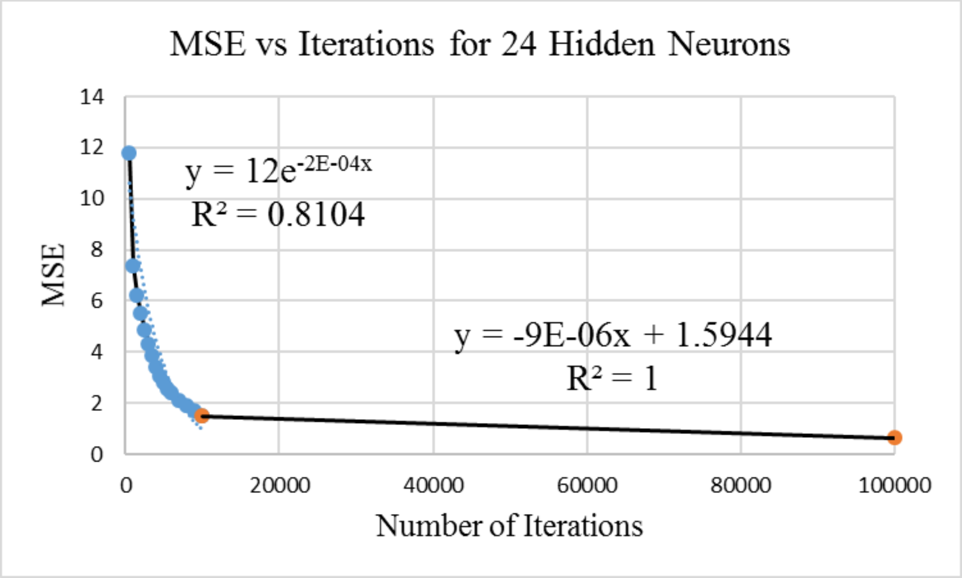


Figure 4.11 MSE versus iterations graph for 28 patterns.

In this case, the MSE is 0.65 for 100,000 iterations which is lowest value of MSE calculated during experimentation. However, increasing the amount of iterations increases the computation time which might be prohibitive for large data sets. The computing time is not important in supervised learning because the network will save nonlinear approximation function (weights and biases) after back propagation learning and will use this saved function to approximate grasping pattern based on object location and geometry.

4.2.2 Evaluating Knowledge Data

Once a nonlinear relationship is developed between the object location and geometrical parameter (input) and grasping pattern (output), the next and last step is to evaluate the learned network in evaluating the outputs based on inputs. The BP Evaluate VI in LabVIEW is used to evaluate learned network (BP Learn) for arbitrary inputs as shown in Figure 4.12.

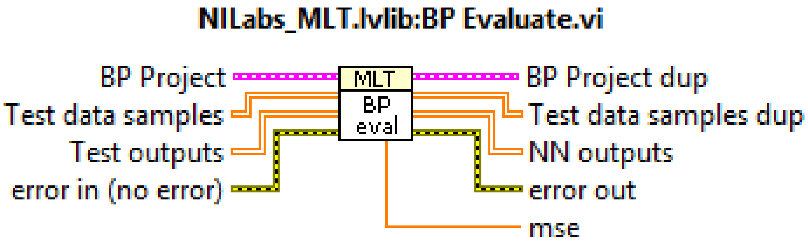
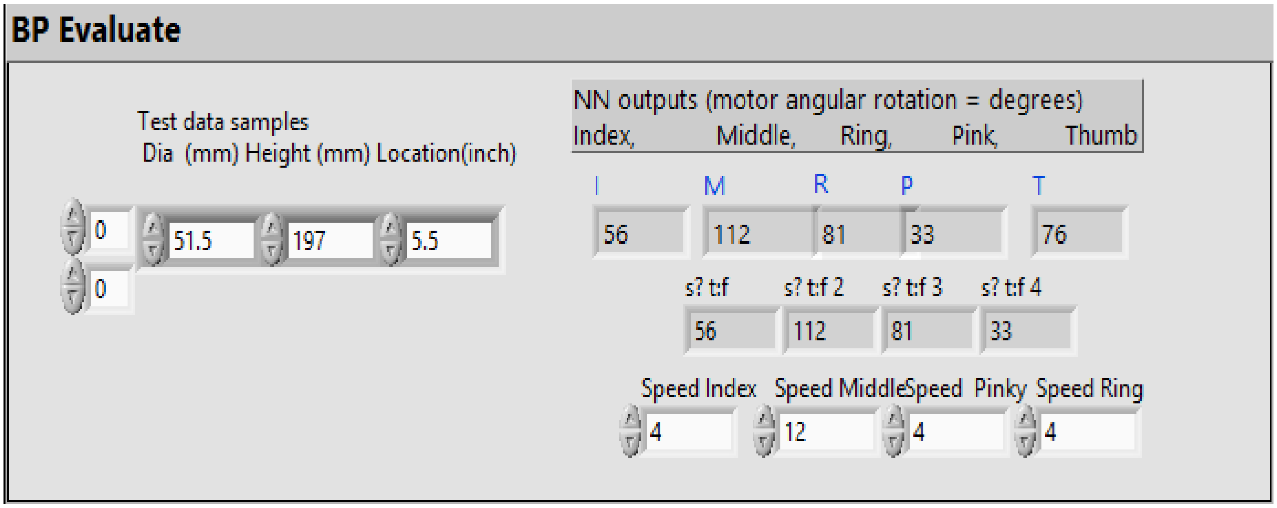


Figure 4.12 BP Evaluate VI description.

BP Evaluate VI requires two important components such as learned back propagation project (weights evaluated by BP learn VI) and test data samples (arbitrary Inputs for testing) as shown in Figure 4.12. Additionally, it requires test outputs which are known outputs. However, it is not necessary to provide test outputs since the target is to evaluate this outputs. At the end, the VI approximates outputs in terms of servo motor angles (grasping pattern) based on nonlinear relationship with inputs (object diameter, height and location) as shown in the GUI in Figure 4.13



BP Evaluate

Test data samples
Dia (mm) Height (mm) Location(inch)

0	51.5	197	5.5
0			

NN outputs (motor angular rotation = degrees)
Index, Middle, Ring, Pink, Thumb

I	M	R	P	T
56	112	81	33	76

s? t:f	s? t:f 2	s? t:f 3	s? t:f 4
56	112	81	33

Speed Index	Speed Middle	Speed Pinky	Speed Ring
4	12	4	4

Figure 4.13 Back propagation evaluate GUI with speed control.

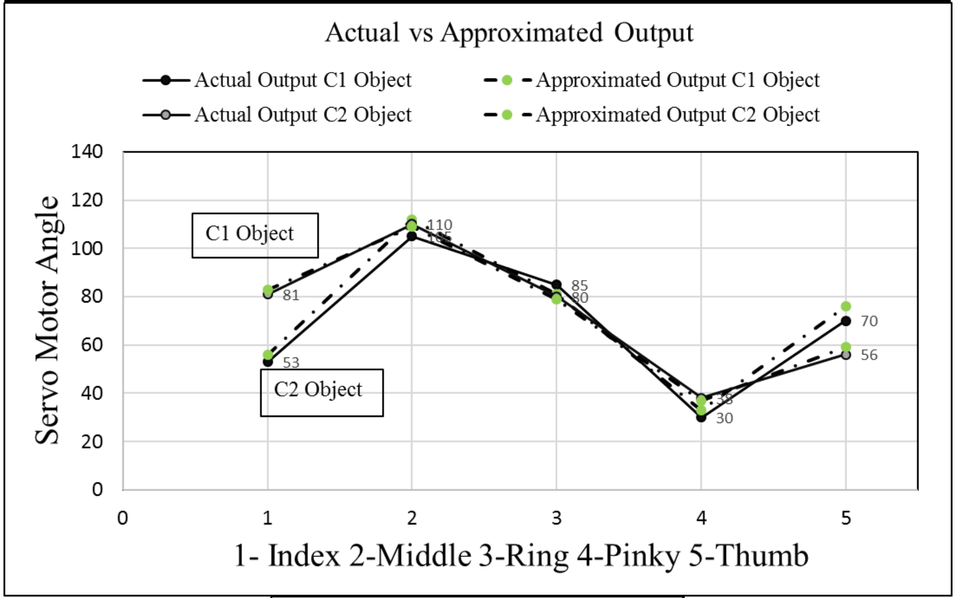
The machine learning evaluation process will be analyzed in three different stages

1. Evaluate grasping pattern for known object.
2. Evaluate grasping pattern for unknown object.

3. Speed control and angle joint limits on Evaluated grasping pattern.

The input set with information of object geometrical features and location, is a 1×3 matrix [diameter (mm), height (mm), location (inch)], and the output set with information of the grasping pattern is a 1×5 matrix [index, middle, ring, pinky, thumb] in degrees. In the first stage, a known object C1 (input: [51.5, 197, 5.5]) is entered in evaluation process where the network approximates grasping pattern output as [56, 112, 81, 33, 76]. It is observed that there are differences of [-3, -7, 4, -3, -6] between actual and approximated output. Moreover, the evaluation is also performed on C2 object to evaluate the network performance and the difference between actual and approximated output is [-2, 1, 1, 1, -3] as shown in Figure 4.14.

Object	Actual Ouput					Approximated Output				
C1	53	105	85	30	70	56	112	81	33	76
C2	81	110	80	38	56	83	109	79	37	59



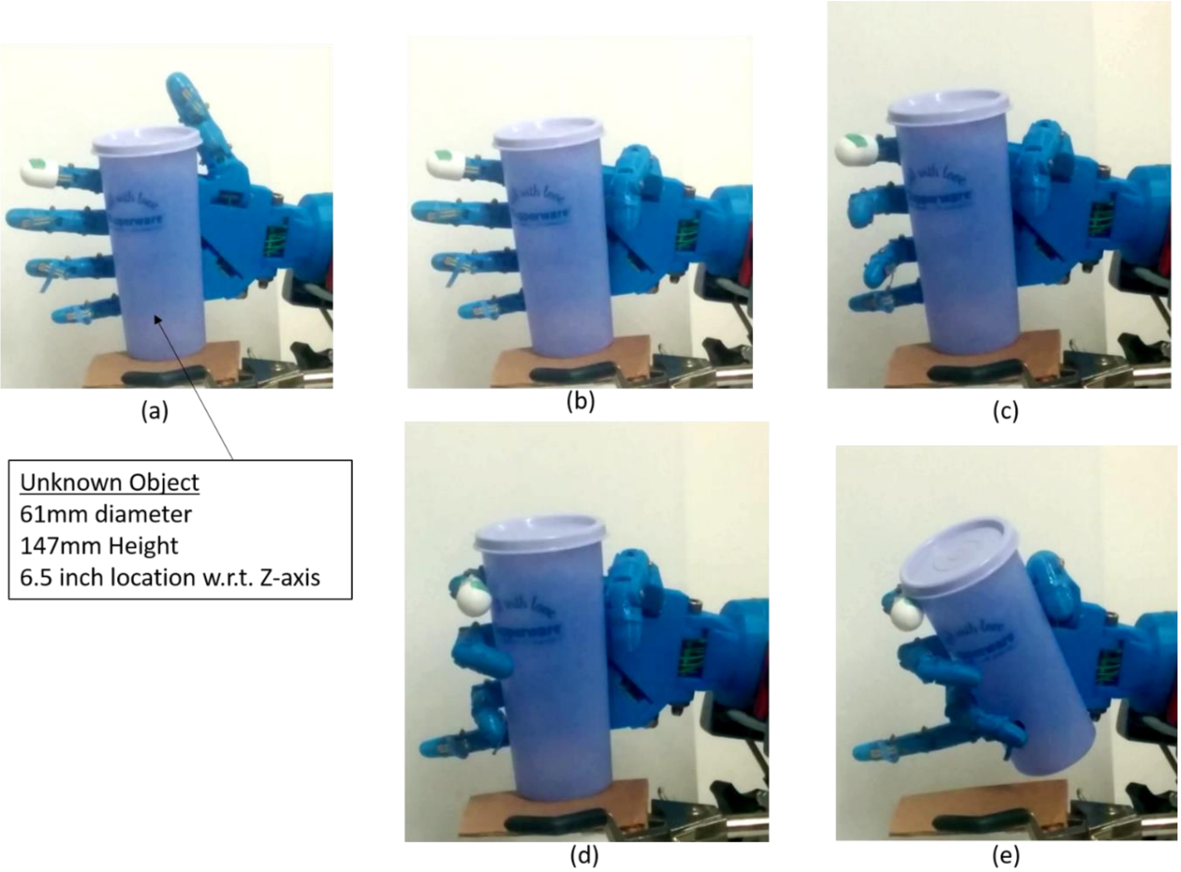
Difference				
-3	-7	4	-3	-6
-2	1	1	1	-3

Figure 4.14 Relationship for Actual vs ANN output.

The difference in the angular rotation of each servo motor is within acceptable range and is relatively low with respect to actual output. Note that these results are calculated for a learned network with 24 hidden neurons, 1000 iterations and 28 patterns. However, more accurate outputs could be achieved by increasing knowledge data (patterns).

Based on current prosthetic hand physiology and actuation, the servo motor angle difference of ± 5 degree for Thumb, ± 10 degree for Index, ±15 degree for Middle, ±10 degree for Ring and ±5 degree for pinky are the acceptable angle difference.

In the second stage, the unknown input property set of an unknown object [input: 61, 147, 6.5] is entered in evaluation process. The network approximates the grasping pattern as [output: 51, 132, 116, 6, 74]. The grasping pattern is shown as lapsed photography sequence in Figure 4.15.

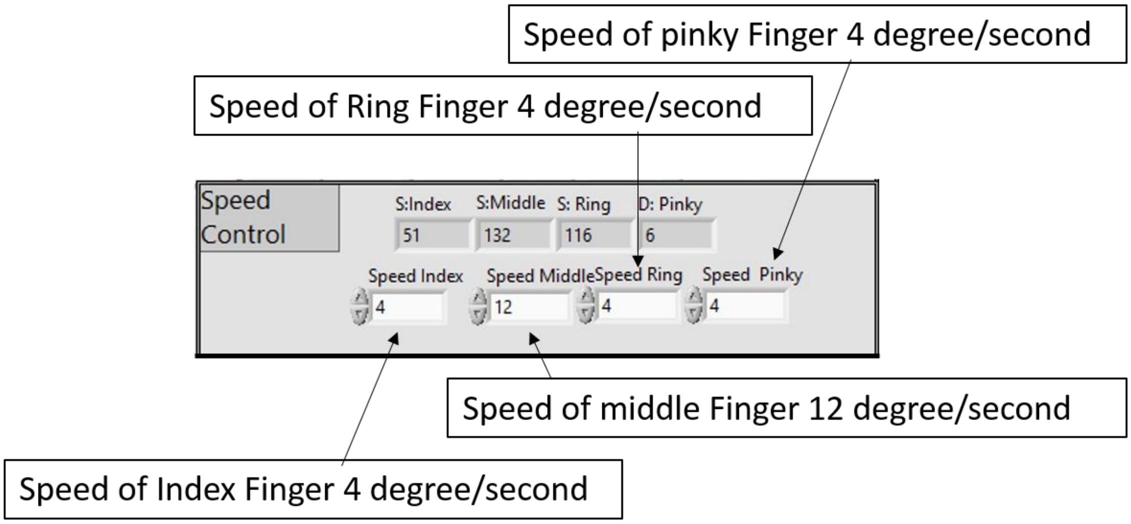


Unknown Object
61mm diameter
147mm Height
6.5 inch location w.r.t. Z-axis

Figure 4.15 Grasping considering object location

It is observed that for an object with a height of 6.5 inches from base, the pinky finger was not used in knowledge data. For the unknown object, the estimated servo motor angle for pinky finger is 6 degrees which is almost negligible. Moreover, as the object diameter increases, the servo motor angle for the index finger decreases, which is observed for this unknown object grasping pattern. The index finger servo motor angle is 60 degrees for a 51.5mm diameter object and the approximated servo motor angle is 51 degrees for a 61mm diameter object in the evaluation procedure. These two observations conclude that the neural network output approximation is satisfactory in most of cases. It is important to mention that this approximation is expected to properly work within the minimum and maximum input range used for training. An input outside that range might produce an unstable output which is not in range of servo motor angles for each respective finger.

The third stage is concerned with the safety of RPH and the object. It is possible that neural networks may approximate an output outside of servo motor angle limits which can damage the RPH. Therefore, a conditional algorithm considering servo motor angle limits was developed in LabVIEW Mathscript which restricts the output in servo angle limit for each finger. The Mathscript allows to perform text base programming in LabVIEW. Moreover, the order of finger and thumb closing and its closing speed can cause considerable damage to the object. In the RPH, the thumb always has to close first to grasp most objects due to the mechanical constraints of hand as discussed in Section 2.2. A conditional logic in the LabVIEW program is developed which will always close the thumb first and then the other fingers in the grasping operation. Moreover, the speed control algorithm is created in LabVIEW program to control the closing speed of fingers as the ratio of the derived servo motor angular rotation divided by the desired time in degrees per second. The LabVIEW GUI is created to control the rotational speed of each finger with numerical control as shown in Figure 4.16. The finger stops closing at the approximated servo motor angle given by the BP Evaluate VI.



Speed of pinky Finger 4 degree/second

Speed of Ring Finger 4 degree/second

Speed Control

S:Index 51

S:Middle 132

S: Ring 116

D: Pinky 6

Speed Index 4

Speed Middle 12

Speed Ring 4

Speed Pinky 4

Speed of middle Finger 12 degree/second

Speed of Index Finger 4 degree/second

Figure 4.16 Speed control of each finger LabVIEW GUI

The object grasping can be performed on various objects using the above three stages. The machine learning performance should be further analyzed for grasping various objects.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

5.1 Conclusions

A research platform has been successfully developed to investigate human robot interaction and robot learning on a RPH. The hardware and software modules have an ability to expose the grasping capabilities of prosthetic hand. The goal of this research was to train a RPH to grasp various objects using HRI and ANN. The technique that integrates manual, glove and artificial neural network control helps to improve grasping capabilities of the RPH. The programming software LabVIEW controls the entire grasping process with three different controls in real-time through a wireless network connection. The successful grasping is performed on known as well as unknown objects by prosthetic hand using machine learning module in LabVIEW. The grasping results are verified experimentally in a laboratory setting. Only selected objects can be grasped due to mechanical constraints of the RPH.

The current servo motor interface board can handle five servo motors in real-time to perform actuation on prosthetic hand. The linear mapping is successfully performed on RPH using data glove. The mapping can be performed by polynomial relation. However, it requires vision system to capture the hand motion and glove motion. The flex sensor interface board is developed in laboratory which transmits glove signals to RPH. The interface board has capability to expand inputs and outputs based on improvements in the hand and glove designs.

The feed forward three-layer artificial neural network based on back propagation algorithm is used to generate nonlinear system architecture between object geometry and location (inputs) and desired grasping pattern (outputs). The inputs and outputs are generated using the data glove and manual control to grasp various objects. This data was then stored in spreadsheet file which simplifies editing and updating the knowledge data set.

The back propagation neural network algorithm allows the user to train the RPH to grasp various objects considering their geometry and location. The learning process has ability to update training data in real-time with the use of a spreadsheet file. The approximated grasping pattern successfully grasped known and unknown, objects with acceptable servo motor angle without considering contact force. Grasping can be improved by providing more knowledge data patterns to train and reduce the mean square error of the neural network. Therefore, grasping a large number of various object could improve the learning process of the RPH.

Another approach to reduce the mean square error is to increase the number of hidden neurons and number of iterations. However, both of these approaches increase computation time. Speed control on each finger was successfully implemented to avoid damage to the object during grasping process. The grasping performance of the RPH has improved by introducing machine learning.

5.2 Future Work

Even though the RPH is able to adequately grasp objects, the user is not able to control each finger joint individually. Additional issues to be studied relate to degrees of freedom of thumb that acts as a supporting part for grasping any object. It is recommended that improvements in fingers and thumb actuation design are implemented for better grasping performance. The current actuation depends on two components such as, the gear design attached to the motor and tension of the cable attached to the gear. These components are important to achieve consist results in object grasping using RPH.

It was also noticed that every time an object had to be positioned near the palm in order to perform grasping because there is no actuation on the wrist or elbow joint. Therefore, it is assumed that the hand will perform grasping after finding the location of the object. Instead of changing the entire concept, only a few output parameters in knowledge data will increase by introducing wrist and elbow actuation in prosthetic hand.

The current flex sensor glove is mapped based on linear mapping. A new mapping scheme could be implemented by mapping several bending positions of prosthetic hand and data glove with a vision system.

The back propagation algorithm is used for supervised grasp learning in the RPH. However, there are several other supervised learning algorithms available in LabVIEW that should be experimented for future work. Additionally, there are deep learning and reinforcement learning algorithms available to handle large data. However, they require expensive computation resources and time to understand the procedure. Moreover, if the variations in object properties are larger, the neural network classification algorithm can be implemented on object to group object based on its features before supervised learning.

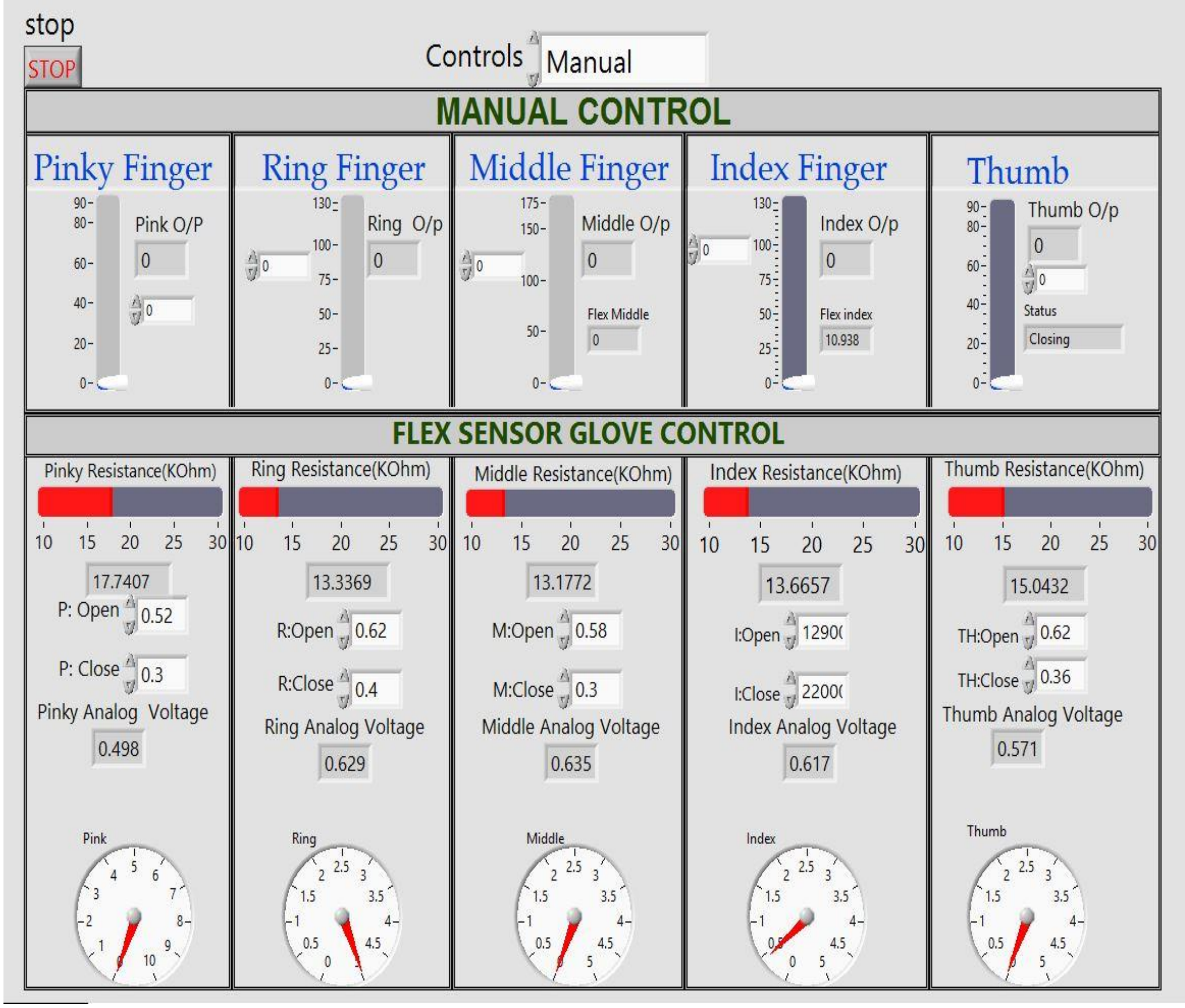
During the grasp evaluating procedure, the geometrical parameter and location of object were entered manually. However, a vision system can be incorporated to recognize the object shape and location thus automating the entire procedure. A limited number of objects were taken for the preliminary experiment. However, versatility in the object grasping can be achieved by increasing number of objects.

Currently, Wi-Fi connection is used to interface software and hardware. However, LabVIEW has ability to use web services to receive commands through internet. Therefore, it would be a more expandable platform if web services were used to control the RPH.

These recommendations show how this research is expandable and where the current state is. The final goal is to improve human robot interaction and robot learning in RPH.

APPENDIX A

LABVIEW GUI



stop

STOP

Controls: Manual

MANUAL CONTROL

Pinky Finger: Pink O/P 0; input 0

Ring Finger: Ring O/p 0; input 0

Middle Finger: Middle O/p 0; input 0; Flex Middle 0

Index Finger: Index O/p 0; input 0; Flex index 10.938

Thumb: Thumb O/p 0; input 0; Status Closing

FLEX SENSOR GLOVE CONTROL

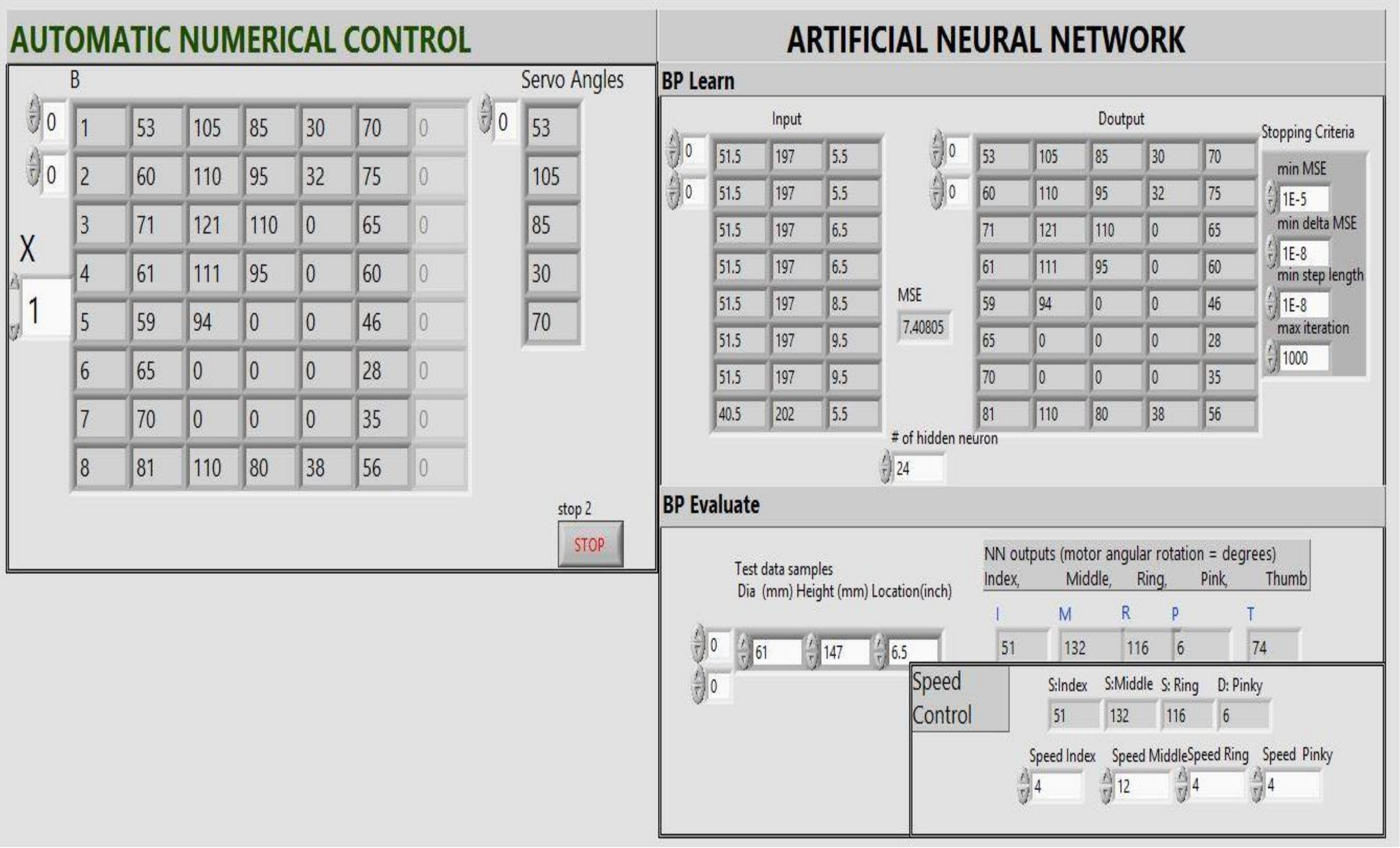
Pinky Resistance(KOhm) 17.7407; P: Open 0.52; P: Close 0.3; Pinky Analog Voltage 0.498

Ring Resistance(KOhm) 13.3369; R:Open 0.62; R:Close 0.4; Ring Analog Voltage 0.629

Middle Resistance(KOhm) 13.1772; M:Open 0.58; M:Close 0.3; Middle Analog Voltage 0.635

Index Resistance(KOhm) 13.6657; I:Open 1290(; I:Close 2200(; Index Analog Voltage 0.617

Thumb Resistance(KOhm) 15.0432; TH:Open 0.62; TH:Close 0.36; Thumb Analog Voltage 0.571



AUTOMATIC NUMERICAL CONTROL

B

0

0

X

1

1	53	105	85	30	70	0
2	60	110	95	32	75	0
3	71	121	110	0	65	0
4	61	111	95	0	60	0
5	59	94	0	0	46	0
6	65	0	0	0	28	0
7	70	0	0	0	35	0
8	81	110	80	38	56	0

Servo Angles

0

53
105
85
30
70

stop 2

STOP

ARTIFICIAL NEURAL NETWORK

BP Learn

Input

0

0

51.5	197	5.5
51.5	197	5.5
51.5	197	6.5
51.5	197	6.5
51.5	197	8.5
51.5	197	9.5
51.5	197	9.5
40.5	202	5.5

MSE 7.40805

Doutput

0

0

53	105	85	30	70
60	110	95	32	75
71	121	110	0	65
61	111	95	0	60
59	94	0	0	46
65	0	0	0	28
70	0	0	0	35
81	110	80	38	56

# of hidden neuron 24

Stopping Criteria

min MSE 1E-5

min delta MSE 1E-8

min step length 1E-8

max iteration 1000

BP Evaluate

Test data samples
Dia (mm) Height (mm) Location(inch)

0

0

61 147 6.5

NN outputs (motor angular rotation = degrees)
Index, Middle, Ring, Pink, Thumb

I	M	R	P	T
51	132	116	6	74

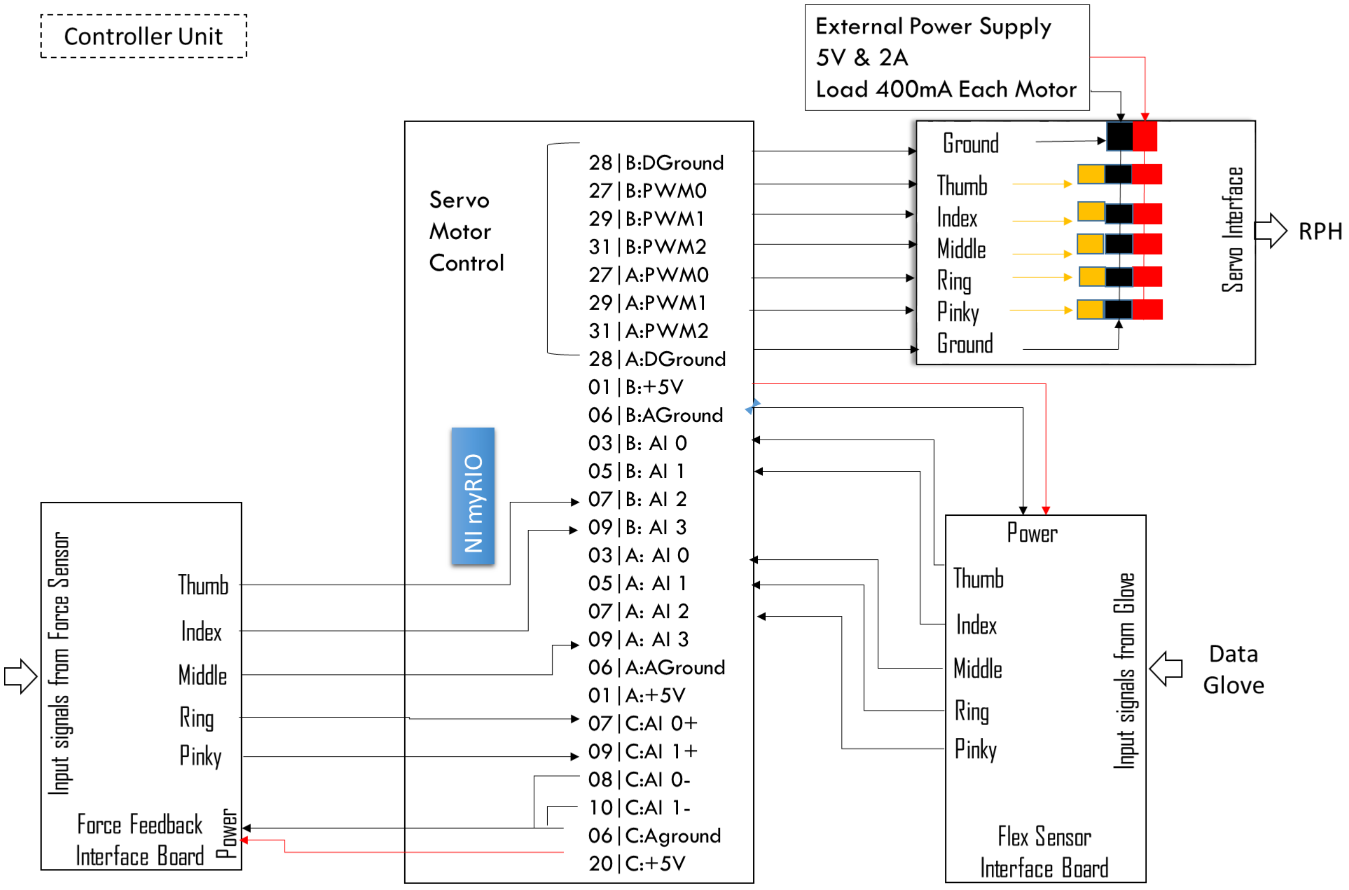
Speed Control

S:Index	S:Middle	S: Ring	D: Pinky
51	132	116	6

Speed Index	Speed Middle	Speed Ring	Speed Pinky
4	12	4	4

APPENDIX B

WIRING DIAGRAM


Controller Unit
External Power Supply
5V & 2A
Load 400mA Each Motor
Servo
Motor
Control
28 | B:DGround
27 | B:PWM0
29 | B:PWM1
31 | B:PWM2
27 | A:PWM0
29 | A:PWM1
31 | A:PWM2
28 | A:DGround
01 | B:+5V
06 | B:AGround
03 | B: AI 0
05 | B: AI 1
07 | B: AI 2
09 | B: AI 3
03 | A: AI 0
05 | A: AI 1
07 | A: AI 2
09 | A: AI 3
06 | A:AGround
01 | A:+5V
07 | C:AI 0+
09 | C:AI 1+
08 | C:AI 0-
10 | C:AI 1-
06 | C:Aground
20 | C:+5V
NI myRIO
Ground
Thumb
Index
Middle
Ring
Pinky
Ground
Servo Interface
RPH
Input signals from Force Sensor
Thumb
Index
Middle
Ring
Pinky
Force Feedback
Interface Board
Power
Power
Thumb
Index
Middle
Ring
Pinky
Input signals from Glove
Flex Sensor
Interface Board
Data
Glove

APPENDIX C

KNOWLEDGE DATA

Input				Output Servo Motor Angles (degrees)								
Object Dia/Width (mm)	Object Height (mm)	Object location on Z-axis (inches)	Characteristic	Nomenclature (-M=Manual Grasping) (-G=Glove Grasping)	Index	Middle	Ring	Pink	Thumb	Orientation		
51.5	197	5.5	1	C1-M	53	105	85	30	70	80		Cylinder
51.5	197	5.5	2	C1-G	60	110	95	32	75	80		
51.5	197	6.5	3	C1-G	71	121	110	0	65	80		
51.5	197	6.5	4	C1-M	61	111	95	0	60	80		
51.5	197	8.5	5	C1-M	59	94	0	0	46	80		
51.5	197	9.5	6	C1-M	65	0	0	0	28	80		
51.5	197	9.5	7	C1-G	70	0	0	0	35	80		
40.5	202	5.5	8	C2-M	81	110	80	38	56	80		
40.5	202	5.5	9	C2-G	82	116	84	42	61	80		
40.5	202	6.5	10	C2-M	72	123	90	0	50	80		
40.5	202	6.5	11	C2-G	74	121	100	0	55	80		
40.5	202	7.5	12	C2-M	79	127	121	0	45	80		
40.5	202	7.5	13	C2-G	78	137	121	0	55	80		
40.5	202	8.5	14	C2-M	79	137	0	0	60	80		
40.5	202	8.5	15	C2-G	81	147	0	0	70	80		
40.5	202	9.5	16	C2-M	84	0	0	0	45	80		
40.5	202	9.5	17	C2-G	90	0	0	0	48	80		
79.1	102.3	5.5	18	C3-M	38	117	90	40	45	80		
79.1	102.3	8.5	19	C3-G	60	152	0	0	72	80		
79.1	102.3	9.5	20	C3-M	65	0	0	0	36	80		
35.5	47.8	9.5	21	C4-G	100	0	0	0	40	80		
35.5	47.8	9.5	22	C4-M	95	0	0	0	35	80		
44.4	51	9	23	C5-M	117	175	0	0	67	80		
35.5	47.8	9	24	C4-M	98	0	0	0	90	80		
50	50	8.5	25	S1-M	35	107	85	0	90			Sphere
50	50	8.5	26	S1-G	48	106	90	0	70	80		
50	50	9.5	27	S1-G	69	0	0	0	69	80		
69.5	65.7	8.25	28	G-apple	109	140	91	0	43	80		

REFERENCES

[1] M. a. Goodrich and A. C. Schultz, “Human-Robot Interaction: A Survey,” *Found. Trends® Human-Computer Interact.*, vol. 1, no. 3, pp. 203–275, 2007.

[2] K. Kosuge and Y. Hirata, “Human-Robot Interaction,” in *Robotics and Biomimetics, 2004. ROBIO 2004. IEEE International Conference on.*, 2004, pp. 8–11.

[3] S. Nakagawara, H. Kajimoto, N. Kawakami, S. Tachi, and I. Kawabuchi, “An encounter-type multi-fingered master hand using circuitous joints,” *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2005, no. April, pp. 2667–2672, 2005.

[4] C. Breazeal, J. Gray, and M. Berlin, “An Embodied Cognition Approach to Mindreading Skills for Socially Intelligent Robots,” *Int. J. Rob. Res.*, vol. 28, no. 5, pp. 656–680, 2009.

[5] T. B. Martin, R. O. Ambrose, M. a. Diftler, R. . J. Platt, and M. J. Butzer, “Tactile gloves for autonomous grasping with the NASA/DARPA Robonaut,” *IEEE Int. Conf. Robot. Autom. 2004. Proceedings. ICRA ’04. 2004*, vol. 2, no. April, pp. 1713–1718, 2004.

[6] T.-H.-L. Le, M. Jilich, A. Landini, M. Zoppi, D. Zlatanov, and R. Molfino, “On the Development of a Specialized Flexible Gripper for Garment Handling,” *J. Autom. Control Eng. Vol*, vol. 1, no. 3, pp. 255–259, 2013.

[7] A. P. Gracia, “Literature Review and Proposed Research Identification of Hand Motion,” 2006.

[8] Y. Yang, Y. Li, C. Ferm"{u}ller, and Y. Aloimonos, “Robot Learning Manipulation Action Plans by ‘Watching’ Unconstrained Videos from the World Wide Web,” *Twenty-Ninth AAAI Conf. Artif. Intell.*, 2015.

[9] P. Simon, *Too Big to Ignore: The Business Case for Big Data*. New Jersey: John Wiley & Sons, Inc., 2013.

[10] E. Ackerman, “How google wants to solve robotic grasping by letting robots learn from themselves,” *IEEE Spectrum*, 2016. [Online]. Available: http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/google-large-scale-robotic-grasping-project.

[11] L. Pinto and A. Gupta, “Supersizing Self-supervision: Learning to Grasp from 50K Tries and 700 Robot Hours,” in *ICRA*, 2016.

[12] M. Belfiore, “A Prosthetic Arm That Gives Amputees the Sense of Touch,” 2015. [Online]. Available: http://www.bloomberg.com/news/articles/2015-10-08/a-prosthetic-arm-that-gives-amputees-the-sense-of-touch. [Accessed: 08-Oct-2016].

[13] “DEKA advanced prosthetic arm gains FDA approval,” *US Department of Veteran Affairs*, 2014. [Online]. Available: http://www.research.va.gov/currents/spring2014/spring2014-34.cfm. [Accessed: 16-May-2016].

[14] T. Padir and J. Schaufeld, “Design of a Human Hand Prosthesis,” *Fac. Worcester Polytech. Inst.*, p. 74, 2012.

[15] G. Langevin, “InMoov Oper Source 3D Printed Life sized Robots.” [Online]. Available: http://inmoov.fr/project/.

[16] C. E. Abrego and M. R. Sobhy, “Developing an Educational and Research Human Robot Interaction Enviornment for a Mechanical Finger/Hand,” in *Proceedings pf the 2015 International Mechanical Engineering Congress and Exposition*, 2015.

[17] A. L. Crawford and A. Perez-Gracia, “Design of a Robotic Hand With a Biologically-Inspired Parallel Actuation System for Prosthetic Applications,” *Vol. 2 34th Annu. Mech. Robot. Conf. Parts A B*, vol. 2, no. PARTS A AND B, pp. 29–36, 2010.

[18] R. July and R. C. C. Systems, “Automatic grasp planning for multifingered robot hands,” *J. Intell. Manuf.*, pp. 297–316, 1992.

[19] “Artificial Neural Network,” *Wikipedia*, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Artificial_neural_network. [Accessed: 02-May-2016].

[20] Piyabongkarn, “Digital Control of Magnetic Levitation System through xPC Real-Time Operating System: Classical Feedforward and Artificial Neuaral Network,” The University of Texas at Arlington, 2000.

[21] S.-C. Wang, “Interdisciplinary Computing in Java Programming,” Boston, MA: Springer US, 2003, pp. 81–100.

[22] Simon Haykin, *Neural Network A Comprehensive Foundations*, Second. Pearson Education, 1999.

[23] P. J. Werbos, “Beyond Regression : New Tools for Prediction and Analysis in the Behavioral Sciences,” Harvard University, 1974.

[24] M. B. M. T. H. H. B. Demuth, *Neural Network Design*. Boston: PWS Publication, 1996.

[25] D. H. Nguyen and B. Widrow, “Neural Networks for Self-Learning Control Systems,” *IEEE Control Systems Magazine*, vol. 10, no. 3. pp. 18–23, 1990.

[26] A. Syed, Z. T. H. Agasbal, T. Melligeri, and B. Gudur, “Flex Sensor Based Robotic Arm Controller Using Micro Controller,” vol. 2012, no. May, pp. 364–366, 2012.

[27] “Flex Sensor,” 2014. [Online]. Available: https://www.sparkfun.com/datasheets/Sensors/Flex/flex22.pdf.

[28] “LabVIEW System Design Software,” *National Instruments*, 2016. [Online]. Available: http://www.ni.com/labview/.

[29] “NI myRIO,” *National Instruments*. [Online]. Available: http://www.ni.com/myrio/.

[30] R. Powell, *Introduction to Electric Circuits*. London NWl 3BH: Arnold, a member of the Holder Headline Group, 1995.

[31] “Servo Motor MG995.” [Online]. Available: http://www.electronicoscaldas.com/datasheet/MG995_Tower-Pro.pdf.

BIOGRAPHICAL INFORMATION

Utsav Shah was born in Gujarat, India in 1993. He received his Bachelor of Technology in Mechatronics Engineering from Ganpat University, Gujarat, India in 2014. He then joined University of Texas at Arlington to pursue his Master of Science degree in Mechanical Engineering in Fall 2014. While at UTA, Utsav worked as a Graduate Teaching Assistant for graduate level course Introduction to Robotics. His research interest includes robotics, automation, control systems and machine learning and hopes to work in a related field after receiving his Master of Science in Mechanical Engineering Degree.