

A TWO STAGE EVENT BASED DATA DRIVEN CONTROLLER FOR
IMPROVED GRASPING OF AN ARTIFICIAL HAND

by

CHRISTOPHER EDWARD ABREGO

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2018

Copyright © by CHRISTOPHER EDWARD ABREGO 2018

All Rights Reserved

To my father and mother who have raised me to be hard working and have encouraged me to always avoid being mediocre.

ACKNOWLEDGEMENTS

I would like to take this space to recognize some influential people throughout the years of this research. First and foremost, I would like to thank my father, Gilberto, and mother, Alicia, who have supported me in my efforts to follow my dream. They have created a nourishing environment so that I can be confident that I have their support. They have encouraged me to aim high and never give up. I want to also thank my brother, Ivan, who has been by my side in difficult moments. I hope that I have been an example to my brother that great things can be accomplished when work hard and efficiently. It wouldn't be just if I didn't recognize and thank my girlfriend, Ana, who helped me extensively throughout the dissertation phase. I want to also acknowledge some students that have helped me in different areas of research, such as in hardware or software. I want to especially thank Eddie Ordonez, Saul Gutierrez, and Henry Nguyen, who are/were undergraduate students that assisted me in research at different stages in the development and control of the hardware platform. I want to also thank Utsav Shah and Abdul Hafiz, both graduate students that have helped in the research process as well. Kermit Beard and Sam Williams machinist that have also assisted in the development of the hardware platform. Finally, I want to recognize my supervising professor, Dr. Panos Shiakolas, who have helped me at every step during my doctoral research. I thank him for also opening up the doors to various branches of research in Human-Robot Interaction and for allowing me to research in his lab.

Gloria Patri, et Filio, et Spiritui Sancto

April 9, 2018

ABSTRACT

A TWO STAGE EVENT BASED DATA DRIVEN CONTROLLER FOR IMPROVED GRASPING OF AN ARTIFICIAL HAND

CHRISTOPHER EDWARD ABREGO, Ph.D.

The University of Texas at Arlington, 2018

Supervising Professor: Panos S. Shiakolas

Committee Members:

Alan Bowling

Kamish Subbarao

Kent Lawrence

Christopher McMurrough

The human hand is one of the greatest (if not the greatest) tool known to mankind for grasping objects. So much so, that researchers have been investigating the development of artificial biomimetic hands in an effort to mimic their functionality and dexterity with the intent to apply the technology to various robotic platforms; ranging from end effectors for industrial pick-and-place robotics, to upper-limb prosthetics, to humanoids. There are certain features that make this endeavor challenging such as the mechanical design, actuation and sensorization, and functionality as well as the interaction from both the view point of interacting with an end user and view point of interacting with an environment.

At UT Arlington, the Manufacturing Automation and Robotic Systems lab has developed a Human-Robot Interaction (HRI) software platform along with a 5-finger 8-DOF biomimetic artificial hand (H2) for research purposes. This research focuses on the grasping component of HRI. The aim of this research is to investigate approaches and develop methodologies for autonomous to semi-autonomous object grasping in physical space. Grasp research has progressed in the investigation of pure kinematic grasping starting with a 5-finger 5-DOF hand (H1) to currently an improved 5-finger 8-DOF artificial biomimetic hand with a dexterous 4-DOF thumb. The method used for grasp pattern prediction is based on Artificial Neural Networks trained with experimental data on the HRI platform. A methodology is developed for the prediction of grasp patterns for objects of non-uniform geometric features based on the object and artificial hand geometric dimensions. This information is used to establish the normalized grasp and length ratios which do not discriminate on object category and further allow for their integration in the training data sets for grasp learning.

It was observed that pure kinematic grasping produced accurate predictions based on object characteristics, however it was noticed that there was an issue of *undergrasping* which sometimes results in unsuccessful grasps. A two stage data/state driven event based controller was proposed to address the unsuccessful grasp scenario. The event based controller has been researched and developed to provide reliable grasping on low compliant convex objects. The controller first stage follows a kinematic objective of properly positioning the fingers for grasping based on the object. The final position of the fingers is predicted by a trained non-discriminatory 3-layer Artificial Neural Network based on the characteristics of the desired object. The controller second stage incorporates sensor information for torque/force feedback to ameliorate “under” grasping and reliably hold the object. This controller has been

verified with the H2 platform with an over 95% success rate and the controller algorithm has also been shown to be transplantable by successfully performing on other robotic hands such as the H1.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iv
ABSTRACT	v
LIST OF ILLUSTRATIONS	xi
LIST OF TABLES	xxii
Chapter	Page
1. Introduction and Proposal	1
1.1 Human-Robot Interaction	1
1.1.1 Research Focus	4
1.2 Grasp Learning	5
2. Pre-Comprehensive Research	16
2.1 Single Finger Platform	16
2.2 Artificial Hand	22
2.2.1 Grasp Learning	25
2.3 Improved Biomimetic Thumb	30
2.4 Machine Learning Object Recognition Sans Vision	32
3. Grasp Analysis and Performance of First Iteration Hand, H1	40
3.1 Grasping Objects	40
3.2 Grasp Observations	51
3.3 Artificial Neural Network Analysis	54
3.3.1 Cylinder Prediction	57
3.3.2 Sphere Prediction	60
3.4 Artificial Neural Network Prediction	61

3.4.1	Software Implementation	61
3.4.2	Hardware Implementation - Grasping Known Objects	64
3.4.3	Hardware Implementation - Grasping <i>Unknown</i> Objects	69
3.4.4	Conclusion - Hardware Implementation	74
3.5	Non Discriminatory Artificial Neural Network	74
3.5.1	Sphere and Cylinder Predictions	76
3.5.2	Software Implementation	77
3.6	ANN Architecture Comparison	79
3.6.1	Cy and CySp: Comparison of Predictive Performance	79
3.6.2	Sp and CySp: Comparison of Predictive Performance	81
3.6.3	Grasping Near-Cylinders and Near-Spheres	83
4.	Grasping Analysis and Verification with Improved Hand H2	89
4.1	ANN Training	90
4.1.1	Cylinder Prediction	91
4.1.2	Sphere Prediction	92
4.2	Artificial Neural Network Prediction	92
4.2.1	Software Implementation	93
4.2.2	Hardware Implementation - Grasping <i>Known</i> Objects	96
4.2.3	Predicting Grasp Patterns of <i>Unknown</i> Objects	101
4.3	Non Discriminatory ANN	107
4.3.1	Software Implementation	109
4.4	ANN Architecture Comparison	111
4.4.1	Cy and CySp: Comparison of Predictive Performance	111
4.4.2	Sp vs. CySp	113
4.5	Grasping <i>Unknown</i> Near Spheres/Cylinders	114
5.	Event Based Grasp Controller	120

5.1	Sensor Fusion Feedback - Hardware	121
5.1.1	QTC Force Sensor	121
5.1.2	Servo Motor Current Sensors	122
5.2	Sensor Fusion Feedback - Algorithm	124
5.3	Grasp Comparison - Test Objects	131
5.3.1	Cylinder	131
5.3.2	Sphere	133
5.4	Grasp Comparison - Unknown Objects	135
5.4.1	Unknown Cylinders	135
5.4.2	Unknown Spheres	137
5.5	Grasping Near Cylinders/Spheres	138
6.	Conclusions and Recommended Future Research	142
6.1	Future Outlook	145
Appendix		
	REFERENCES	171
	BIOGRAPHICAL STATEMENT	183

LIST OF ILLUSTRATIONS

Figure	Page
1.1 Sub branches of robotics and the area of scope concerned in this research. Grasping begin a subset of human-robot interaction and human-robot interaction being a subset of robotics.	2
1.2 Comparison between two versions of the artificial hand, A) Previous artificial hand (H1) with under-actuated thumb B) New artificial hand (H2) with fully motorized thumb.	6
2.1 Single mechanical finger with force sensor on the second joint.	17
2.2 Front panel of the system in which the user interacts directly to control the system.	18
2.3 Image of the artificial hand with a artificial skin sleeve used to dress manikins.	22
2.4 Springs used to retract the fingers back to their original open position.	23
2.5 Grasp limits based on the geometry of the artificial hand. On the left, the maximum diameter cylinder that can be grasped is about 79mm. On the right, the minimum diameter cylinder is about 28.5mm.	25
2.6 Mechanical hand demonstrating grasping simple objects such as cylin- ders of different sizes and a sphere.	26
2.7 Visual of ANN. Note that the layers are fully connected and there is no direct communication between input and output layer.	27
2.8 Time lapse images of the artificial hand grasping the unknown object based on input characteristics, [1].	29

2.9	Improved design of the mechanical thumb (view of the back of the hand).	31
2.10	Thumb joint kinematics [2].	32
2.11	Thumb with improved mobility. Naming convention for each motor on the thumb starting from the base of the thumb, moving toward the tip of the thumb.	33
2.12	Image of the user grasping a sphere while using the interaction glove. .	34
2.13	Algorithm flow used to identify objects.	35
3.1	Procedure for grasping considering both software analysis and hardware validation.	40
3.2	Image of objects with their dimensions marked which are used for the prediction algorithm.	42
3.3	Dimensions of the artificial hand including the palm height and the arc grasp. Note, that the largest object that the hand can grasp has a diameter no larger than 3.70". This was found through experimentation. The yellow arrow indicates the positive direction of the height.	43
3.4	Spheres and cylinders that are used for grasping.	44
3.5	Image of the computer with the real-time video feed of the artificial hand on the left hand side, and the front panel of the LabVIEW VI used to control the actuation of the servo motors for grasping.	44
3.6	Distribution of grasps of cylinders with with different finger combinations respect to their relative size, G_r and L_r	45
3.7	Distribution of grasping spheres with different finger combinations with respect to their relative size, G_r and L_r	46
3.8	Different cylinders grasped at various heights.	47

3.9	Cylinders that are grasped with 3 fingers considering them at a height of 2.5 inches. Note that the objects grasped are small and only require 3 fingers for grasping.	48
3.10	Pie graphs of the total fingers used to grasp cylinders at different heights.	48
3.11	Number for fingers used to grasp different spheres at various heights. .	49
3.12	Pie graphs of the total finger used to grasp spheres at different heights.	50
3.13	Image of the same cylinder (C_1 , $L_r = 2.069$ and $G_r = 0.549$) grasped at different heights. On the right, the normalized close percent values at the respective height. The top image shows the cylinder being grasped at height 3.5" and progresses down to 1" at the bottom image.	52
3.14	Areas of contact on the fingertips with respect to the object G_r . The different colors on the fingertips represent the contact area on the finger for an object with a certain G_r range.	53
3.15	Image of the general architecture of ANNs. Inputs are the diameter (D_o), length (L_o), and relative height (H_o) of the object. The outputs are the motor angles index (θ_I), middle (θ_M), ring (θ_R), pinky (θ_P), and thumb (θ_T).	55
3.16	The Absolute Percent Difference and Mean Square Error of the resulting ANN architectures when the number of iterations for optimization is varied as functions of number of nodes in the hidden layer. This training was performed with 27 data sets.	56
3.17	The resulting Absolute Percent Difference and MSE of the various ANN architectures. The training was performed with 24 training data sets. .	60

3.18	Angular errors between the actual and predicted angles for grasping cylinders. The predictions were performed with Cy-9/2600 trained with 76 data sets. Black bars indicate <i>overgrasping</i> and red bars indicate <i>undergrasping</i>	62
3.19	Angular errors between the actual and predicted angles for grasping cylinders. The predictions were performed with Cy-13/4600 trained with 76 data sets. Black bars indicate <i>overgrasping</i> and red bars indicate <i>undergrasping</i>	62
3.20	Angular error between the actual and ANN predicted angles for grasping sphere S1. The predictions were performed with Sp-15/5000 with 24 data sets. Blue bars indicate <i>overgrasping</i> and red bars indicate <i>undergrasping</i>	63
3.21	Angular error between the actual and ANN predicted angles for grasping spheres. The predictions were performed with Sp-9/3600 trained with 24 data sets. Black bars indicate <i>overgrasping</i> and red bars indicate <i>undergrasping</i>	64
3.22	Images of the final grasp from the resulting prediction of Cy-9/2600 trained with 76 data sets.	66
3.23	Images of the final grasp from the resulting prediction of the Cy-12/4600 trained with 76 data sets.	66
3.24	Images of the final position of the fingers in grasping based on the Sp-9/3600. The object grasped is a Sphere S1.	68
3.25	Images of the final position of the fingers in grasping based on the Sp-15/5000 predictions. The object grasped is a Sphere S1.	68
3.26	Objects used to evaluate the prediction performance of the ANN. . . .	69

3.27	Final grasp positions of the fingers for object UC 1. The ANN was trained with Cy-12/4600.	71
3.28	Final grasp positions of the fingers for object UC2 with Cy-12/4600. .	72
3.29	Object used as an unknown sphere to grasp.	72
3.30	Final grasp positions of the fingers for object US1 with Sp-15/5000. . .	73
3.31	Schematic of grasp planning with discriminating objects, top, and non-discriminating objects, bottom.	76
3.32	Performance of the ANN based on the APD and MSE. The ANN was trained with 100 training sets.	77
3.33	Angular error between the actual and ANN predicted angles for grasping sphere. The predictions were performed with CySp-10/3000 trained with 100 data sets. Blue bars indicate <i>overgrasping</i> and red bars indicate <i>undergrasping</i>	78
3.34	Angular error between the actual and ANN predicted angles for grasping. The predictions were performed with CySp-8/5000 trained with 100 data sets. Blue bars indicate <i>overgrasping</i> and red bars indicate <i>undergrasping</i>	79
3.35	Comparison of the angular errors between the actual grasp and the predictions of the various ANN architectures. This is for grasping object C22 at height 2.5" 3.5".	80
3.36	Comparison of the angular error between the actual grasp and the predictions of the various ANN architectures. This is grasping object S1 at height 1.0" and 1.5".	82
3.37	Two objects that are neither completely spherical or cylindrical.	83
4.1	Thumb on the second generation hand with individually control joints.	89
4.2	From left to right, abduction, adduction, extension, flexion.	90

4.3	The APD and MSE of the resulting ANN architectures when the number of iterations for optimization is varied as functions of number of nodes in the hidden layer. This training was performed with 76 data sets. . .	91
4.4	The Absolute Percent Difference and Mean Square Error of the resulting ANN architectures when the number of iterations for optimization is varied as functions of number of nodes in the hidden layer. This training was performed with 24 data sets.	93
4.5	Angular error between the actual and ANN predicted angles for grasping cylinder C22. The predictions were performed with Cy-11/2400 trained with 76 data sets. Blue bars indicate <i>overgrasping</i> and red bars indicate <i>undergrasping</i>	94
4.6	Angular error between the actual and ANN predicted angles for grasping cylinder C22. The predictions were performed with an Cy-13/4600 trained with 76 data sets. Blue bars indicate <i>overgrasping</i> and red bars indicate <i>undergrasping</i>	95
4.7	Angular error between the actual and ANN predicted angles for grasping sphere S1. The predictions were performed with Sp-11/3000 trained with 24 data sets. Blue bars indicate <i>overgrasping</i> and red bars indicate <i>undergrasping</i>	95
4.8	Angular error between the actual and ANN predicted angles for grasping sphere S1. The predictions were performed with Sp-12/5000 trained with 24 data sets. Blue bars indicate <i>overgrasping</i> and red bars indicate <i>undergrasping</i>	96
4.9	Images of the final grasp on object C22 from the resulting prediction of Cy-11/2400 trained with 76 data sets.	99

4.10	Images of the final grasp on object C22 from the resulting prediction of Cy-13/4600 trained with 76 data sets.	100
4.11	Images of the final grasp of object S1 from the resulting prediction of Sp-11/3000 trained with 24 data sets.	101
4.12	Images of the final grasp of object S1 from the resulting prediction of Sp-12/5000 trained with 24 data sets.	101
4.13	Final grasp positions of the fingers for object UC1. Cy-13/4600 was trained with 76 data sets.	102
4.14	Final grasp positions of the fingers for object UC 1. Cy-11/2400 was trained with 76 data sets.	102
4.15	Final grasp positions of the fingers for object UC2 with Cy-11/2400, trained with 76 data sets.	104
4.16	Final grasp positions of the fingers for object UC2 with Cy-13/4600 trained with 76 data sets.	105
4.17	Final grasp positions of the fingers for object US1 with Sp-11/3000 trained with 24 data sets.	106
4.18	Final grasp positions of the fingers for object US1 with Sp-12/5000 trained with 24 data sets.	106
4.19	The Absolute Percent Difference and Mean Square Error of the resulting ANN architectures when the number of iterations for optimization is varied as functions of number of nodes in the hidden layer. This training was performed with 100 data sets.	108
4.20	Angular error between the actual and ANN predicted angles for grasping sphere S1 and cylinder C22. The predictions were performed with CySp-15/2000 trained with 100 data sets. Blue bars indicate <i>overgrasping</i> and red bars indicate <i>undergrasping</i>	110

4.21	Angular error between the actual and ANN predicted angles for grasping sphere S1 and cylinder C22. The predictions were performed with CySp-13/5000 trained with 100 data sets. Blue bars indicate <i>overgrasping</i> and red bars indicate <i>undergrasping</i>	110
4.22	The angular error the actual and predicted joint angles for grasping. The results are provided by Cy-11/2400 and Cy-13/4600 trained with 76 data sets and CySp-15/2000 and CySp-13/5000 trained with 108 data sets.	112
4.23	The angular error the actual and predicted joint angles for grasping. The results are provided by Sp-11/2000 and SP-12/5000 trained with 24 data sets and CySp-15/2000 and CySp-13/5000 trained with 108 data sets.	113
5.1	Images of the QTC sensors attached to the different locations of the finger tips.	122
5.2	Frequency of sensor activation based on experimentally grasping objects.	124
5.3	This is a histogram of the final torques required to grasp all the objects. There were 112 grasps consisting of different grasp poses.	125
5.4	Decision making diagram for closed loop grasping algorithm	126
5.5	Schematic of ideal scenario in terms of sensor data during grasping event. The top graph is the angular position of the motors and on the bottom graph is the torque values from the sensors on the motors.	129
5.6	<i>Undergrasping</i> scenario during grasping event. The top graph is the angular position of the motors and on the bottom graph is the torque values from the sensors on the motors.	130

5.7	<i>Overgrasping</i> scenario during grasping event. The top graph is the angular position of the motors and on the bottom graph is the torque values from the sensors on the motors.	131
5.8	Comparison of grasp pattern predictions and actual motor angles for grasping C22.	132
5.9	Images of the final grasp of all the open and closed loop predictions (top is at height 3.5" and bottom is at 2.5").	132
5.10	The stages of progression, from left to right, of the improvement of the development of the artificial hand (H1 and H2) as well as the control schemes utilized for grasping C22.	134
5.11	Comparison of grasp pattern predictions and actual motor angles for grasping S1.	135
5.12	Images of the predicted grasps on object S1 for the open and closed loop control schemes. On the top the grasps are made at a height of 1.0" and the bottom row are made at 1.5".	135
5.13	The stages of progression, from left to right, of the improvement of the development of the artificial hand (H1 and H2) as well as the control schemes utilized for grasping S1.	136
5.14	Comparison of all the grasps that were performed with the various control algorithms. Red dots indicate the location of <i>undergrasping</i>	136
5.15	Comparison of all the grasps that were performed with the various control algorithms. The red dots indicate the location of <i>undergrasping</i>	137
5.16	Compared grasps on object US1 at a relative height 3.5".	138
5.17	Various images of the comparison for grasping object UO1. The green dots indicate locations where there is <i>undergrasping</i> . The top row grasp is at height 1.0" and progresses down to the last row of 3.5".	139

5.18	Various images of the comparison for grasping object UO1. The red dots indicate locations where there is <i>undergrasping</i> . The top row grasp is at 1.5” and progresses down to the last row of 3.5”.	140
1	Images of all the objects used for grasping. Dimensions of the objects are presented in Table 1.	147
2	Absolute difference between the ANN prediction and the actual values, with respect to the number of iterations for optimization and number of nodes in the hidden layer. This was performed with 76 data sets for training.	150
3	Absolute Difference of the resulting architectures. Training was performed with 24 data sets.	151
4	Different resulting hand postures depending on objects and hand chosen for study (program used is <i>GraspIt!</i>) [3].	152
5	Final grasp positions of the fingers for object UC 1. The ANN was trained with an Cy-9/2600.	153
6	Final grasp positions of the fingers for object UC 2. The ANN was trained with Cy-9/2600.	154
7	Final grasp positions of the fingers for object US1 with Sp-9/3600. . .	155
8	Absolute difference of the various ANN architectures. The ANN was trained with 100 data sets.	156
9	The average Absolute Difference across all data sets for the different architectures. These results were gathered from 76 training data sets. .	157
10	The average Absolute Difference across all data sets for the different architectures. These results were gathered from 24 training data sets. .	158

11	The average Absolute Difference of each ANN architecture with 108 training sets. This set incorporated both spheres and cylinders for training.	159
12	Comparison of grasps of objects OU2 at height 1.0", with all 6 ANN architectures.	160
13	Comparison of grasps of objects OU2 at height 1.0", with all 6 ANN architectures.	161
14	Comparison of grasps of objects OU2 at height 1.0", with all 6 ANN architectures.	162
15	Comparison of grasps of objects OU2 at height 1.0", with all 6 ANN architectures.	163
16	Comparison of grasps of objects OU2 at height 1.0", with all 6 ANN architectures.	164
17	Comparison of grasps of objects OU2 at height 1.5", with all 6 ANN architectures.	165
18	Comparison of grasps of objects OU2 at height 1.5", with all 6 ANN architectures.	166
19	Comparison of grasps of objects OU2 at height 1.5", with all 6 ANN architectures.	167

LIST OF TABLES

Table	Page
2.1 Comparison between ANN output motors angles and actual motor angles, [1].	28
2.2 K-Nearest Neighbor test results. Percent accuracy based on method Closest or Majority Vote (MV).	38
2.3 Test result for the ANN and SVM algorithms. Percent accuracy based on both methods.	38
3.1 Table of the geometric properties of the objects used to test the ANN prediction abilities.	70
3.2 Final finger positions for grasping object UC 1. The units for the fingers are in degrees. The first 4 rows are results of Cy-9/2600, while the last 4 rows are results from Cy-12/4600.	70
3.3 Final finger positions for grasping object UC 2. The units for the fingers are in degrees. The first 4 rows are results of Cy-9/2600, while the last 4 rows are results from Cy-12/4600.	71
3.4 Final finger positions for grasping object US 1. The units for the fingers are in degrees. The first 4 rows are results of Sp-9/3600, while the last 4 rows are results from Sp-15/5000.	73
3.5 Summary of the average absolute error for all the architectures across all the fingers.	81
3.6 Summary of the average absolute error for all the architectures across all the fingers.	82

3.7	Dimensions of objects UO1 and UO2.	84
3.8	ANN predictions by various architectures on unidentified object UO1. The values in bold indicate when the finger has moved however, no contact was made with the object.	85
3.9	ANN predictions by various architectures on unidentified object UO2. The values in bold indicate when the finger has moved however, no contact was made with the object.	86
4.1	Final finger positions for grasping object UC1. The units for the fingers are in degrees.	103
4.2	Final finger positions for grasping object UC2. The units for the fingers are in degrees.	104
4.3	Final finger positions for grasping object US1. The units for the fingers are in degrees.	107
4.4	Average (all motor angles) absolute percent error of different ANNs with respect to <i>known</i> objects. Values presented are in terms of percentage.	114
4.5	ANN predictions by various architectures on <i>unknown</i> object UO1 (ap- ple). The motor angle values in underlined bold indicate an active finger not making contact with the object.	118
4.6	ANN predictions by various architectures on <i>unknown</i> object UO2 (or- ange). The motor angle values in underlined bold indicate an active finger not making contact with the object.	119
5.1	Average and Standard deviation of torque values (lb-in) required for a successful grasp, based on finger.	125
1	Objects, spheres and cylinders that were grasped by the mechanical hand. Diameter and height of the object were give as well as its nor- malized dimension relative to the height of the hand and the grasp space.	148

2	ANN predictions by various architectures on unidentified object UC1. The values in bold indicate when the finger has moved however, no contact was made with the object.	168
3	ANN predictions by various architectures on unidentified object UC2. The values in bold indicate when the finger has moved however, no contact was made with the object.	169
4	ANN predictions by various architectures on unidentified object US1. The values in bold indicate when the finger has moved however, no contact was made with the object.	170

CHAPTER 1

Introduction and Proposal

1.1 Human-Robot Interaction

The study of Human-Robot Interaction (HRI) seeks to improve, facilitate, and expand the way we, humans, interact with robots. In recent history, robots have been a topic of fascination, fear, wonder and inspiration, which has not only reflected itself in popular culture, music and films, but also in academia, research, and industry. It is important to get terms right, so one of the first questions that needs to be answered before proceeding is, “*what is a robot?*”. The word “robot”, can be traced back to have Czech origin, *robota*, which in Czech means servitude [4]. To do justice to the origins of the word, and to not be ambiguous with language, the term robot will be considered a physical agent with a primary objective of being assistive in nature.

Since the conception of robots and in the context of this research, there are some general properties that still remain in all robots. For instance, robots are usually a mechanical construction, composed of electrical communication and actuation, are able to perform physical actions, are programmed to mimic human intelligence and attempt to mimic anatomical components of a biological system (human and non-human animals), hence the term biomimetic is used to categorize the types of robots that are in scope of this research.

With the advancements in technology, robots have branched out into many areas that were once only imagined. Robots have flourished in these areas because they have proved to be a more reliable and effective solution than previous solutions. Just to mention a few applications, robots can be found in manufacturing, assembly,

medicine, rehabilitation, space exploration, tele-manipulation, prosthetics, assistive living, and even entertainment applications such as theater stage actors, drink servers, musicians and dancers [5].

Figure 1.1 shows the macro and micro level of this research. The top most layer, robotics is the study of robots and a subcategory is HRI which seeks to improve and facilitate communication with robots. This research will focus on machine learning in autonomous grasping.

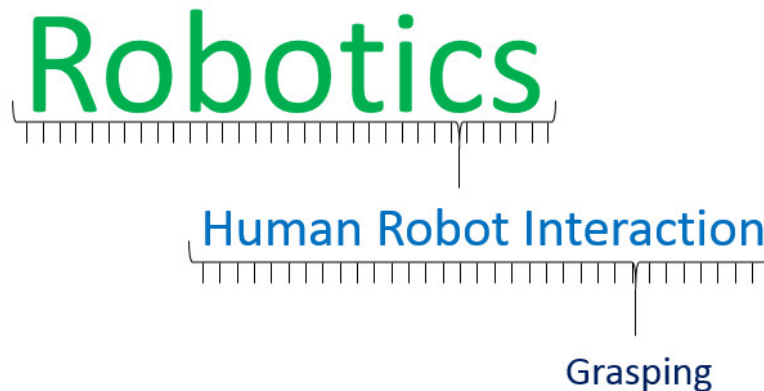


Figure 1.1: Sub branches of robotics and the area of scope concerned in this research. Grasping begin a subset of human-robot interaction and human-robot interaction being a subset of robotics.

Interaction modes between humans and robots have improved, with the advancement of technologies, and now there is a wide spectrum of modalities that can fit almost any application. One can imagine the first types of interaction modes were levers, buttons, and switches to have the robot perform a certain action. Later, joysticks and mechanical manipulators were used to “drive” robots. Then, with the development of computers, perhaps interaction shifted to keyboard and mouse interface to interact and command the robot. The improvement of graphical user interfaces (GUIs) have also improved interaction. To further improve tele-manipulation, modal-

ities such as voice and gesture recognition to control a robot emerged. The desire for tele-presence has motivated the development of modalities in the attempt to transport the user (or the actions of the user) to a remote or alternative location to perform an operation.

There are ways of interaction with robots, some more efficient than others depending on the situation and application. For instance, some tasks require humans-in-the-loop as opposed to complete robotic autonomy, such as robot assisted surgery where cooperation combines benefits of both robotic precision and human judgment [6, 7, 8]. On the other hand, there are repetitive and mundane tasks where robotic automation will outperform humans. Among one of the subcategories of HRI is grasping which depending on the application can be performed in an autonomous or a cooperative operation.

It seems that the most fascinating modality is cognition or neural signals to control robots. Cognitive remote control seeks, among other objectives, to facilitate interaction for those humans who lack the natural faculties that previous interaction modalities would require, such as arms, hands, legs, etc. With the increase in technology, one could wonder to the future advancements and potential for HRI. The goal, from the beginning of HRI, has been to consider the robot a perfect communication partner to help carry out tasks. The goal is to reduce the residual effects of communication, such as frustration, trust or disappointment which come from unmet expectations, so that the user can focus more on task completion instead of the user focusing on communication.

1.1.1 Research Focus

The research that has been performed, and is proposed, aims to contribute towards improving the current state of HRI in the branch of grasping. Contributions in the field of HRI require knowledge in technologies and methodologies that will enhance and facilitate communication and task execution. As a result, the discipline of mechanical engineering alone is not enough and topics from electrical and computer science are needed. It is also important to mention that one side of HRI research is concerned with how the human operator interacts, hence there is also a subjective dimension in HRI research, particularly of grasping with a high DOF biomimetic hand.

In the vast spectrum of HRI and to further focus the scope of this research, we consider a particular type of robot, biomimetic. The term biomimetic refers to a robot that attempts to mimic the biological process of living organisms, particularly humans. There is a natural tendency to anthropomorphise robots for the sake of familiarity and comfort, but also for advantages in task execution [9]. However, there are also negative connotations and potential anxiety that arises with anthropomorphic robots [10]. In this research, the topic of HRI will be explored using an artificial hand as a platform for research and hardware validation.

In terms of tele-manipulation and HRI research there are benefits in having biomimetic hands, as opposed to nonhuman-like manipulators. Ciocarlie considered the human hand the most versatile end-effector known [3]. Matter puts forth the idea that multi-fingered hands provide for a unique sensing modality that cannot be achieved with other manipulators [11]. This is accomplished by incorporating position, force, tactile, and proximity sensors into the manipulator. Moreover, multi-fingered hands also provide an opportunity to obtain information about the mechanical and physical attributes of objects and tasks from a human reflection perspective.

Likewise, in support of biomimetic hands, Li describes the efficiency of manipulation through relatively fast and small motions of fingers [12]. Consider the wide variety of motions and coordination that hands and fingers perform for instance when tying shoe laces to peeling an orange to holding a spoon.

The line of research that has been investigated concerns the exploration and improvement of HRI. Here, a biomimetic artificial hand will be the platform where hardware execution and validation will be performed. The technologies incorporated, as well as the design and control of HRI platforms will be investigated. It should be noted that the focus of this research is not to develop a platform, but to contribute to the body of knowledge in HRI in the category of machine learning decision making for grasp execution.

1.2 Grasp Learning

The application of machine learning algorithms aims to improve grasping as well as its understanding. Based on research conducted thus far with the biomimetic artificial hand, it was proposed to investigate the application of machine learning algorithms to improve grasping. Grasping will further be explored with the incorporation of an improved thumb (in terms of dexterity and controllability) on the biomimetic artificial hand and furthermore with the incorporation of an event based two stage data driven controller. Figure 1.2 shows the previous and current artificial hand with the improved controllable thumb.

Grasping is not a new field in robotics and neither is the desire to improve grasping. The study of grasping can be performed in virtual space through a simulation or in physical space. Furthermore, the study of grasping is limited by the hand characteristics selected for grasping. For instance, a primitive (low quality, dexterity and controllability) biomimetic hand will not be able to evaluate or assess the use of

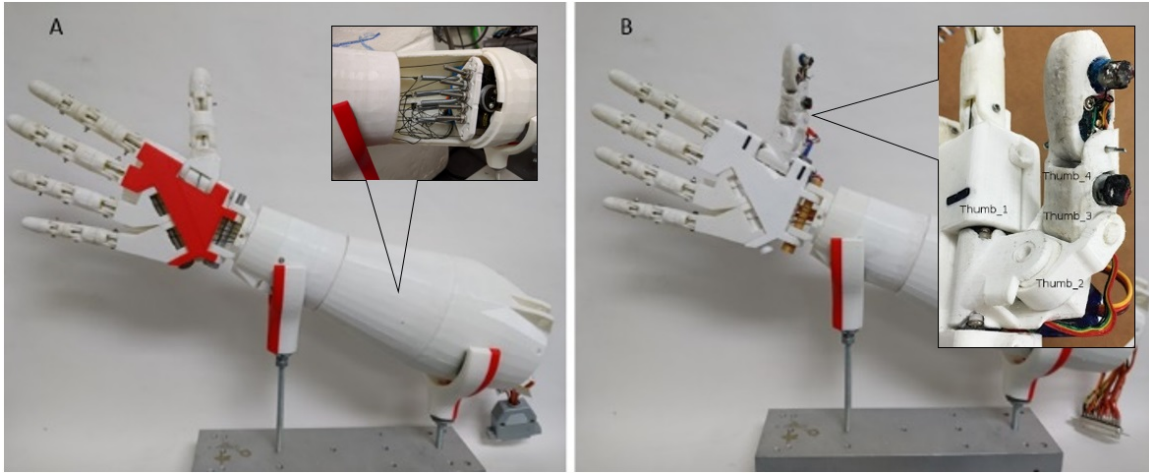


Figure 1.2: Comparison between two versions of the artificial hand, A) Previous artificial hand (H1) with under-actuated thumb B) New artificial hand (H2) with fully motorized thumb.

learning algorithms that require features that the physical artificial hand itself does not pose. For instance, if it is proposed to develop an algorithm that improves power grasping, however the physical biomimetic hand doesn't have the dexterity or flexibility to move the fingers accordingly, then no matter how complex and complete the algorithm, it will not be able to perform better than the quality of the hand used for investigation. Other limitations could be due to the lack of actuators for motion and sensors for observation. Another example is that a normal (perpendicular) force sensor embedded on the gripper will not be able to help observe the torque or vibration, similarly one motor will not be able to control the multiple degrees of freedom of an under-actuated hand. This is very important to emphasize, hardware will limit the capability and quality of a grasp, hence algorithms can only be effective up to the point of hardware capabilities. A brief study of machine learning algorithms has already begun for certain aspects of grasping. Grasping will focus on considering both kinematics of the fingers of the artificial hand and force reaction upon contact.

To help simplify grasping, Mason *et al.* researched synergies of the hand when grasping a variety of objects [13]. Along the hypothesis that hand postures for grasping can be reduced to a few principal components, “eigenpostures”, Mason studied postural synergies of the hand for different grasps. Using single value decomposition analysis on the kinematic position of the user hand when grasping, it was found that “eigenpostures” were similar across subjects and grasps.

Similarly, Ciocarlie *et al.* studied grasp synthesis considering low-dimensional posture subspaces for an artificial hand [3]. The case is made that the added versatility of anthropomorphic artificial hands has come with the cost of increased complexity. Figure 4 in the Appendix shows the different grasp poses that result with different hands and objects. As more mechanical components developed on the artificial hand, so did the degrees of freedom. However, it is known that 80% of the variance in hand postures are contained in two-dimensional subspaces, hence many hand postures used for grasping have the same eigenpostures. Ciocarlie developed an optimization algorithm for selecting and planning grasping.

Among some of the algorithms of choice to classify stability of grasping is by using Gaussian Mixture Models (GMM). Note that in terms of grasping, stability is a term used to characterize a grasp of an object where the risk of the object leaving or slipping away from the hand is minimal. Huang *et al.* considered using GMMs to facilitate real-time grasping in robots for HRI [14]. The GMM was chosen because of its ability to effectively extrapolate missing data. To accomplish this, a three-step approach was taken. The system computes a variety of stable grasps for a given object, then computes the probability distribution of grasps based on the computed grasps, and finally generates a grasp. About 1000 testing grasps were generated for experimentation. In the experiments, grasps of complex objects had a success rate of 85%.

Santello *et al.* studied the temporal synergies of hand movement and determined the influence of sensory cues to control these synergies [15]. Subjects were asked to reach to and grasp various objects under experimental conditions. The motion of the arm and of the hand (15 degrees of freedom) were recorded. A principal components analysis was developed to provide a concise description of the spatio-temporal patterns underlying the motion. It was noted that having the participants see the object during the reaching movement had no influence on the kinematics, and the effect of the physical presence of the object became manifested primarily after the fingers had contacted the object. After discriminant analysis, two principal components accounted for more than 75% of the variance. For both components, there was a strong positive correlation in the rotations of metacarpophalangeal and proximal interphalangeal joints of the fingers. The first principal component exhibited a pattern of finger extension reversing to flexion, whereas the second principal component became important only in the second half of the reaching movement.

Dillmann *et al.* looked into programming by demonstration (PbD) of service robots to simplify programming [16]. The goal was to transfer skills from a human operator to a robotic manipulation system. Optimization techniques were used to simplify processing observational data such as sensors and vision acquisition. To focus on grasping, the user would wear a glove and grasp a variety of objects. Kinematic and force data gathered from the glove sensors were then processed using artificial neural networks to distinguish various grasps, such as power and precision grasps.

One of the most successful machine learning algorithms for classification is Support Vector Machines (SVM). Zollner *et al.* researched PbD for biomimetic robotic manipulators, such as arms and hands [17] [18]. When classifying the motion of the human hand and arm, support vector machine (SVM) classifiers were used to find patterns and similarities in each grasp. Particularly, this task was performed using

SVMLight to classify dynamic grasps. They found that the dynamic grasps corresponded to twenty six classes. The training data set contained 2600 vectors and the SVM classifier used 486 support vectors for classification.

Similarly, Dang *et al.* used machine learning for the perception and determination of stable grasps [19]. The only data that is used to determine a stable grasp is tactile feedback and kinematic data that give the position and orientation of the hand. A SVM is used to predict if the grasp is stable or not. They collected about 15000 robotic grasps from 936 objects across 23 different classes for training. Using a Ferrari-Canny grasp metric [20], an error threshold quality was determined to establish if a grasp is stable or not. As a result of the SVM classifier, an overall accuracy of about 70.4% was achieved.

By using a variety of machine learning algorithms, Laaksonen *et al.* investigated grasp stability [21]. Laaksonen first considered reducing features of the robotic hand, since there would be a total of about 169 to 241, depending on the hand used to practice. To reduce the amount of features, methods such as Partial Component Analysis, Image Moments, Histogram using 10 bins, Spatial Partitioning, Local Binary Patterns and Row Column sums were used. Next, grasping was considered as a classification problem, the output would be classified as either stable or unstable. The different classifiers used were SVMs, Gaussian Mixture Models (GMM), k -Nearest Neighbors (KNN), and AdaBoost. After experimentation with 11 different data sets and all dimensional reduction types and all classifiers, it was found that AdaBoost was the best performing classifier with the highest success percentages.

Miller *et al.* researched and developed *Graspit!*, an open-source grasp simulator that considers seven different manipulators to virtually grasp objects [22]. This program helps visualize grasping and investigate how grasping will help. In the study, Miller used the virtual Barrett Hand to investigate grasping. About 1600 grasps were

simulated, consisting of 100 random roll and spread angle combinations. SVM regression was used to create a mapping between object shape, grasp parameters and the quality of grasp. Given these qualities as a fixed length vector, the SVM returns a single scalar which estimates the quality of grasp.

Bekiroglu *et al.* used machine learning algorithms to assess grasping stability based on haptic data from sensors [23]. They studied the planning stage of grasping (the moments before grasping is executed) and the grasping stage (closed loop online control). To do so, Bekiroglu considered grasp stability as a probability distribution where grasp stability depends on different measure and known factors. Also, support vector machines and AdaBoost are used to model the instantaneous model while Hidden Markov models were used for the general time series case. Both AdaBoost and SVM were used as classifiers for their probabilistic outputs regarding stability. Moreover, the Hidden Markov models (HMMs) were generally used to represent stable and unstable grasps.

HMMs have been used to recognize continuous grasping sequences to further study grasping. Bernardin *et al.* recorded the motion of users and analyzed the patterns recognized [24]. Programming by Demonstration was the motivation behind the research, and HMMs were used in the data acquisition portion of the study, using information from a data glove and tactile sensors. The commercially available HMM toolkit Hidden Markov Model Toolkit (HTK) was used for recognition routines. Grasping classification was achieved by separating grasps according to their purpose, hand shape and contact points. The features for the HMM, 16 finger joint angle values and the outputs were 13 pressure values. The system was able to fuse inputs from two sensors, detect grasping phases while demonstrating and classified the different grasps.

Alternatively, Carrozza *et al.* did not use machine learning algorithms to control the artificial hand developed, the CyberHand [25]. In the control of the artificial hand was distributed. There was a high and low level control architecture for grasping. The high level control component looked into classifying and interpreting the users intentions and suggest grasping initiatives. This command would then be sent to the lower level control component, which would observe and correct the motion of the actual hand, by using PID algorithms to control the motion of the motors actuating the fingers.

Another method is to consider a hybrid approach and open the control field to both motion (position) and force control, as opposed to only position control. Raibert and Craig investigated using a hybrid system that considered both position and force in robotic manipulation by using a two prong jaw manipulator and a Scheinman force-sensing wrist[26]. The position control loop used a constant gain, proportional-integral-derivative (PID) control law. Force control was achieved by combining proportional-integral (PI) control with a saturation-type feedback limiter and a simple feed-forward term. Zhang and Paul also looked into hybrid control of both force and position [27]. Similarly, Khatib performed a similar analysis with a 2-prong end-effector [28] [29]. The same concept of hybrid control eventually branched out to biomimetic hands. Roccella and Zecca *et al.* developed a nine (DoF) 3 finger artificial hand with two motors, which was controlled with information from the three dimensional force (F_x, F_y, F_z) sensors on the fingertip of the hand and encoder position of the motors [30] [31].

There have been different grasping approaches based on various algorithms to achieve a successful grasp, in either virtual or physical space. Machine Learning (ML) algorithms have been useful for the prediction of values when a model of the system is not available or hard to develop. In the broad field of ML algorithms, the

main similarity is that it involves a *learning* process in which a predictive method is improved by reducing the value of a cost function or reducing the error between predicted and benchmark values.

Li *et al.* investigated grasping with the 3-finger Barrett hand and the 4-finger hand under object shape uncertainty [32]. They developed a system where the system takes into account the uncertainty of the object shape during the grasp planning and execution stage. The process by which uncertainty is considered in the grasping algorithm is that the shape uncertainty is parameterized using Gaussian processes and incorporated as a constraint. Their algorithm was demonstrated in hardware on a 4-finger Allegro hand to grasp objects such as a cylinder, spray bottle and a jug.

Fok *et al.* developed Recurrent Neural Networks (RNNs) for grasping force optimization of multi-fingered (3-finger) hands [33]. The force optimization problem is formulated as nonlinear and the cost function is convex and subject to linear equality constraints. Their algorithm was shown to converge to a force vector that remained positive towards the normal direction of the object. Xia *et al.* considered the problem of grasping with a multifingered robot as an optimization problem where a cost (objective) function would need to be minimized considered a set of assumed constraints such as object balance (external wrench) and finger force distribution [34]. They used a RNN to solve their linear and nonlinear constraint equations and they demonstrated that their solutions converged to an optimal grasping force. They further demonstrated their results in simulation by successfully grasping polyhedral objects of known weights with a 3-finger gripper.

The recent trend in grasping research has been in the area of coupling a vision system to a robot with a simple (2 or 3-finger) gripper as discussed by [35, 36, 37, 38, 39, 40, 41, 42, 43, 44] among others. This type of research focused on tripodal

or 2-finger pinch grasp of various objects in a controlled area, such as a bin, box or conveyor belt.

Using the Baxter robot, Lenz *et al.* researched the application of deep learning using a 2-finger gripper and a vision system [45]. Images of the objects were acquired and used in a two stage method to predict potential rectangles superimposed over the object to be grasped. The rectangles would give the general gripper approach orientation and location to the object for a successful grasp. They were able to structure regularization in the deep neural net in order to learn features from the images gathered. Similarly, Redmon *et al.* expanded on the bounding box method by using five layers of Convolutional Neural Networks (CNNs) into the first five layers of the prediction algorithm architecture [46]. Among one of their results is that their approach performs classification so that in a single step it recognizes the object and predicts a good grasp rectangle. Later, Kumra *et al.* further expanded the method of prediction grasp rectangles for a parallel (2-finger) gripper using deep CNNs to extract object features from an image [47]. Kerzel *et al.* researched the use of the NICO (3-finger gripper) robot in a self-learning process where the input to the system was a RGB image and after two convolutional layers of 16 filters and then two dense layers of 900 neurons, the resulting output was the joint configuration of the robotic arm (6 DoF) [48]. Guo *et al.* also employed the use of five CNNs, using max pooling as a filter, and then used two fully connected layers with the two final outputs, one for object discovery and the other for grasp representation [49]. Guo used the 4 DoF Barrett Hand (3-finger gripper) and a Microsoft Kinect camera for hardware verification. They were able to discover and successfully grasp a desired target object from a cluster of 15 or more objects and grasp it successfully.

What has been described till now has been methodologies where grasping was performed in simulation or if in hardware, performed by a 2 or 3-finger gripper with

low or basic finger complexity. However, researchers have also shown interest to investigate grasping using biomimetic artificial hands. For a biomimetic hand, one of the most important digits is the thumb since it plays a major role in object grasping and manipulation, as described by Cotugno and Nanayakkara [50] [51]. Regoli *et al.* used the iCub humanoid robot to implement a two level controller with the addition of Gaussian Mixture Model (GMM) to predict the joint reference values and object position [52]. Vicente *et al.* also researched manipulation using a method based on position and visual feedback to demonstrate their importance in grasping tasks [53]. Only few researchers have investigated similar research with reference to Ku *et al.* who used the Robonaut-2 hand (5-fingers) and Huang *et al.* who used a Dreamer Mekahand (5-finger underactuated) with the use of different Artificial Neural Networks for grasp point predictions [54] [55]. The University of Bologna (UB) hand has also been a platform for anthropomorphic grasping research and has made major contributions in demonstrating effective grasps in hardware with Principal Component Analysis of 36 hand configurations [56, 57, 58, 59]. Ficuciello *et al.* was able to consider three predominant postural synergies which yields the entire reference set of grasps defined for the 5-finger 12 DoF UB hand [57].

In summary, the proliferation of robots in society, industry, military, entertainment, medicine, and education require efficient communication for interaction and task execution. In the area of anthropomorphic grasping, this research will focus on researching machine learning algorithms in an effort to more reliably grasp objects with an anthropomorphic hand.

This dissertation document is organized in the following way. In Chapter 2, initial research in the development of the hardware platform to be used in research is discussed. Furthermore, initial research in grasp pattern prediction is elucidated as a proof of concept, based on the hardware platforms developed (biomimetic hand

H1 and H2) as well as their advantages pertaining to grasping research. In Chapter 3, the methodology, results, and conclusions of grasping investigation are discussed. Afterwards, some of the limitations found with H1 are addressed with the second iteration hand H2 in Chapter 4. To further address issues that arose with both H1 and H2, an event based controller was investigated and incorporated. The results and discussion are described in Chapter 5. Then a summary all the investigations and results are presented in Chapter 6. It is important to note that there is some supplemental information such as tables and figures located in the Appendix.

CHAPTER 2

Pre-Comprehensive Research

This chapter discusses research that has been conducted since the start of the graduate career. The majority of research conducted builds upon the central area of HRI, however, some research topics have gone above and beyond the current focus of research. Consequently, a patent is being pursued in conjunction with the UT Southwestern, regarding a medical diagnostic device developed. Research ranges from vision feedback control to a rolling robot to the development of a HRI platform using an artificial hand. The purpose of this chapter is to provide scholarship up to this moment and quality of research and development of skills in software and hardware needed for the proposed research.

2.1 Single Finger Platform

A paper was published in the ASME IMECE 2015 conference that goes into more depth regarding the development, interaction and accomplishments with a single finger platform, Abrego [60].

French designer, Gaël Langevin has developed a full body humanoid robot which is open source so that hobbyists and researchers can download the stereo lithography (stl) files and 3D print on their own [61]. A single finger was printed, along with a holder for the motor and microcontrol board, as a proof of concept, presented in Figure 2.1. The tendons of the finger were actuated with a servo motor, TowerPro MG995, via NI LabVIEW software. A tendon, attached to the motor, runs through the finger and is anchored at the tip of the finger. As a result, when the motor turns,

the tendon is tensioned, resulting in the finger joints rotating. It is important to note that the system is underactuated and the rotation of some joints depend on the full rotation of previous joints. This is a result of having one actuator rotate 3 rotational joints in series.

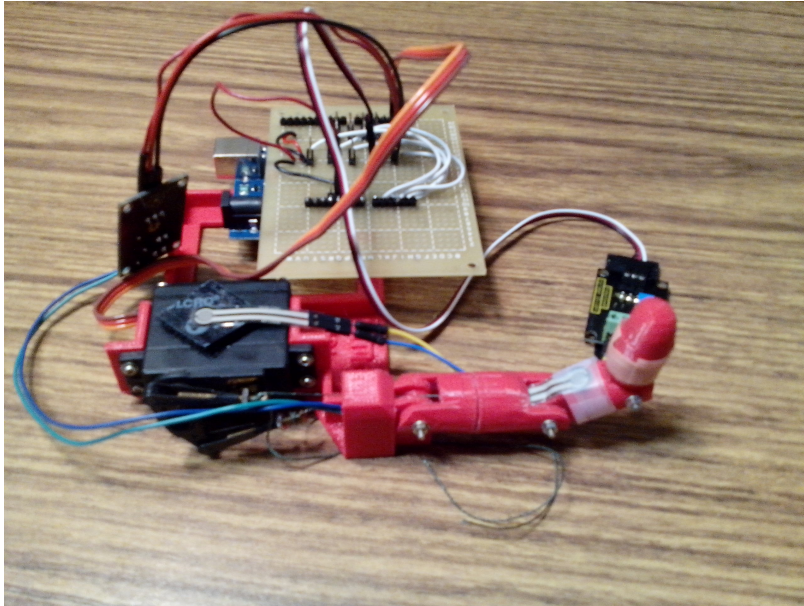


Figure 2.1: Single mechanical finger with force sensor on the second joint.

For the scope of this research, certain modalities were investigated, using a single finger as a testbed, such as force control, holding action and voice control. A single finger was employed to prove algorithmic concepts, however the end goal of this research is to have a full hand (multiple fingers) perform grasping and manipulation tasks.

As mentioned, the software used to interact with the single finger was National Instruments LabVIEW. The user interacts with the front panel, which is a graphical user interface. The front panel of the program is found in Figure 2.2. There is a pull down menu where the user can select the desired interaction mode. There is a

vertical slider where the user can control the angular motion of the motor. The user can also observe the force experienced on the force sensors. The user can also observe the voice command that the system recognized.

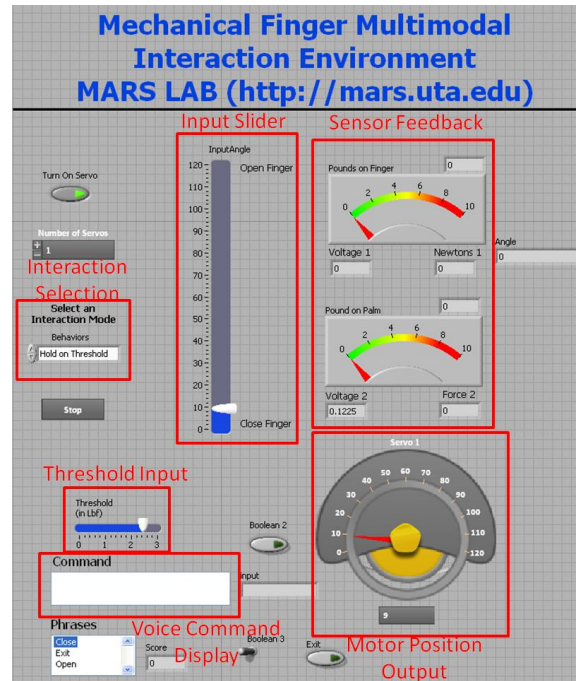


Figure 2.2: Front panel of the system in which the user interacts directly to control the system.

To accomplish force control, the artificial finger was outfitted with a force sensor, FSR 400 series from Interlink Electronics. The force sensor would provide an indication of the force experienced by the finger. As the force on the sensor increases, the resistance decreases, which in turn changes the voltage output of the sensor. Initially, an Arduino control board was used for data acquisition and control signal execution¹. However, a programming transition was then made from the Arduino control board to the National Instruments myRIO. Decision making algorithms based

¹Even though the Arduino board was utilized, no programming was completed in the native Arduino language. All programming was completed in LabVIEW.

on the force sensed were developed in the LabVIEW programming environment, so that if a certain amount of force was experienced, then the motor would stop closing the finger. For this, finite state machine algorithms were developed and programmed in LabVIEW in order to perform the action of position or state holding at a desired applied force.

Finite state machines can be explained as distinct and independent states and their relation with each other. It is claimed that each motion of the artificial hand can be decomposed to a combination of simple prime states of the hand. Static and dynamic states of the artificial hand were categorized and established. For the artificial hand to move from one state to another state, an action would need to be executed. This helps visualize sequential actions that the hand must take in order to complete a desired action from the user, which in turn will help the operator provide coherent inputs to accomplish tasks.

Additionally, voice commands to interact with the single finger were explored. Three commands were pre-programmed to be recognized in the LabVIEW program; close hand, open hand, half hand. *Close hand* had the motor action command to rotate the motor output shaft about 120° , *half hand* would rotate the output shaft about 60° and *open hand* would rotate the output shaft to 0° . The software program used for speech recognition was Windows Speech Recognition. Previously, there was a program created to call windows speech recognition, capture the voice, then interpret the utterance, all in LabVIEW [62]. This LabVIEW sub-program was used to actuate the finger with voice commands.

The voice recognition system was first trained using Christopher Abrego's voice². The training was completed by reading out-loud a series of short stories. Voice would be captured with a desktop microphone and then correlated to the required text to

²In terms of demographics, Christopher is a Hispanic male who grew up in Texas.

read. The voice control modality was able to successfully recognize and match the utterance of the user with a specified motor action. To further test this modality, 14 people were asked to utter 21 commands³, in order to test the reliability of the voice recognition program and command execution. The participants that were asked to interact with the platform ranged from male and female as well as White, Black, Hispanic, Arab, and Indian backgrounds, so a variety of accents were tested.

The participants were given a microphone to capture their voice and a series of commands to recite ⁴. The number of times (i.e. the number of repeats) it took the system to respond⁵ to the voice command were counted. The 21 command test was performed in a quiet environment and also in noisy environment. The noisy environment is characterized by playing background music. In total there were 588 commands spoken (14 participants asked to give 42 commands). The number of times that it took the system to respond appropriately to the spoken command was collected. For instance, if the system responded appropriately the first time the command was issued, then 1 was inputted for that command, likewise, if the user had to repeat the command three times, then 3 was inputted to the command. Therefore, at the end of a test session with a participant, a 42 by 1 table had the number of times it took for the system to recognize the command and execute motor functions correctly.

Among the results, intuitively, the system responded better when there is no noise compared to with noise. It was observed that the voice command *half hand* was the most misinterpreted command and, on average, it required the most repetitions.

³21 commands were spoken, consisting of *close hand*, *half hand*, and *open hand*. The voice commands were ordered such that the successive commands would not repeat.

⁴Every participant was given the same number of commands in the same order

⁵By responding, the system had to accept and correctly interpret the command and perform the appropriate mechanical motion.

The reason behind this is could be because of the pronunciation of the word *half*. It was noticed that some people pronounced it “haff” and others “halef”⁶ and others “have”.

Moreover, it was observed that system was not sensitive to males or females, nor accents based on backgrounds. It should be noted that the average number of attempts for each trial in the case with no noise imposed was 1.241 and 1.439 for the case where noise was imposed. Not only this, but among the 588 voice commands spoken, only once was there a false command executed. This is important to point out because in a practical application of this platform, executing a false command is riskier than having to repeat a voice command.

Having to repeat a command assumes, in practice, that the finger is currently in a acceptable state but needs to transition to a new state to achieve a certain purpose, such as being in a open state and needing to go to the close state in order to hold an object. If the system does not respond the first time to the voice command, the user would have to repeat this voice command until the system responds appropriately. From the tests conducted, on the conservative side, the user would have to repeat the word 1.439 times (2 times) for the system to respond appropriately. Having to repeat a voice command without it falsely executing a different command is more assuring and reliable, than if the system executed a false command instead.

The developed platform was able to successfully stop the motors from moving once a certain force threshold⁷ was achieved. Moreover, the investigation of voice commands show promise to further incorporate this modality in future HRI platforms and combine it with desired applied force thresholds.

⁶More emphasis on the “l”

⁷The force threshold is set by the user to be whatever desired value. This is beneficial since it allows for the threshold to be modular, based on the object being grasped.

2.2 Artificial Hand

Based on the success of the single finger platform, the idea of a single finger expanded to an entire 5 finger artificial hand. Once again, stl files were downloaded from inmoov.fr to print the complete hand, along with the forearm, presented in Figure 2.3. Each finger is actuated with a single servo motor and all 5 motors are packaged in the forearm. The control program in LabVIEW was expanded by duplicating the actions required to move the single finger, 5 times in order to move each finger. Once again, as mentioned in Abrego [60], the cable mechanism that mimics muscle tendons, is employed to rotate the joints in the finger.

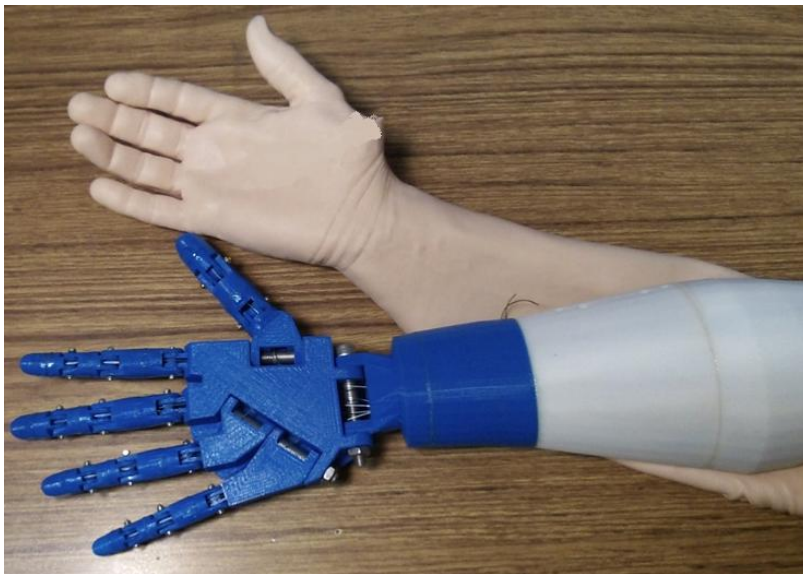


Figure 2.3: Image of the artificial hand with a artificial skin sleeve used to dress manikins.

The tensioning mechanism for the mechanical hand has been an issue in the past, such as when initial investigations were performed on the single finger. This is due to the fact that the pull length required for finger actuation is not equal to slack provided for rotation. After rotating, there is unnecessary slack in the tendon, which

delays reverse motion. To ameliorate this issue, springs were attached to the base of the servo holder hub. This ensures that the finger is maintained in tension and allows for a smooth retraction of the finger.

The spring is anchored on a nut which revolves around a bolt which is fixed to the servo carriage, presented in Figure 2.4. This allows for the tension on the cables to be modular, i.e. increase or decrease the tension as desired. As the motor rotates and pulls the cable, the reverse direction cable is maintained in tension by the spring. When the motor rotates the opposite direction (direction required to open⁸ the finger), the spring pulls on the reverse cable so that the finger opens.

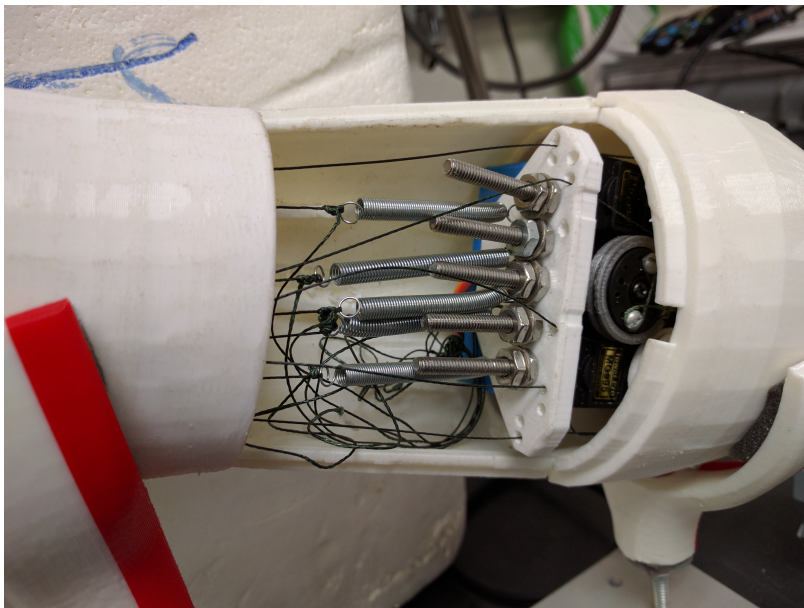


Figure 2.4: Springs used to retract the fingers back to their original open position.

After the mechanical and electrical modifications were incorporated to the artificial hand, focus was switched to investigate grasping. The research topic of grasping

⁸Open is the term that is used to indicate that all the joints have not been rotated from their zero position.

is popular when developing artificial hands. Among one of the pioneers in this research, Cutkosky [63], researched the classification of different grasps by observing how operators in a machine shop would grasp tools. This provided a taxonomy for grasping patterns and helped initiate the growing field of grasping for mechanical hands.

Mattar [11] reviewed biomimetic hands to further show the importance of not only the use of biomimetic hands, but also the study of grasping. A non-exhaustive list of pioneering mechanical hands include the the UTAH/M.I.T. hand [64], the Stanford/JPL hand [65], Belgrade hand [66], the famous DLR hand [67] [68] [69] [70], the University of Bologna hand [71] [72] [73] [58], the Gifu hand [74] [75] [76] [77], and the CyberHand [78] [25] among the most recognized mechanical hands developed and used for research.

The question of grasping has focused on different aspects such as dexterity [79], hand synergies [13], hand postures [3], grip control [80], and slip detection [81]. It is evident that the development of a mechanical hand has created many branches of research that incorporate different technologies and controls based on the applications.

As shown in Figure 2.5, the space allotted by the geometry of the InMoov hand allows for objects no bigger than 79mm in diameter, about the size of a bottle of water, and no smaller than 28mm to be grasped. Any object with diameter beyond this would not be able to be securely grasped using a cylindrical grasp [1]. Therefore, at this point of research, investigations regarding grasping were limited to objects that fit in that range. These pitfalls will be further addressed in later sections as well as a solution to the lack of dexterity.

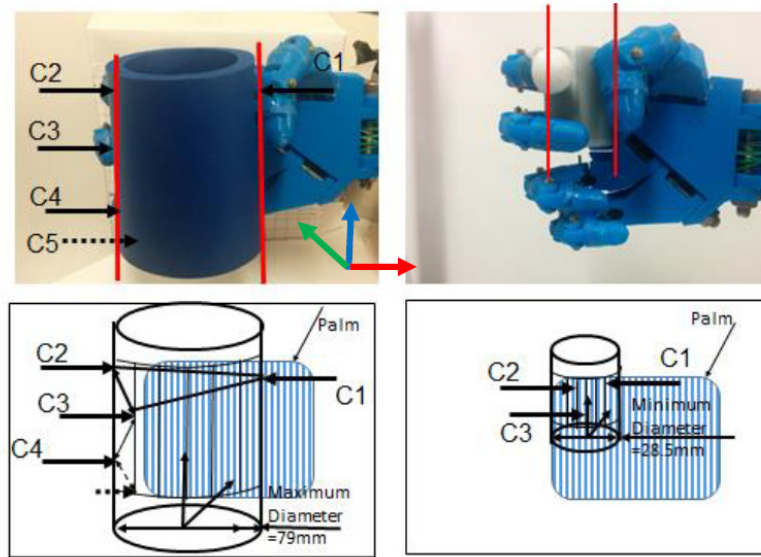


Figure 2.5: Grasp limits based on the geometry of the artificial hand. On the left, the maximum diameter cylinder that can be grasped is about 79mm. On the right, the minimum diameter cylinder is about 28.5mm.

2.2.1 Grasp Learning

Grasp learning in the artificial hand platform is another important focus in the field of developing biomimetic hands. The goal of this line of research is to use machine learning algorithms in order to automate the artificial hand to appropriately grasp objects of different dimensions, without requiring the user to input a large amount of data. As mentioned earlier, at this point in research, objects such as cylinders were used to investigate grasping, shown in Figure 2.6.

In order to “learn” the objects, an Artificial Neural Network (ANN) was programmed in LabVIEW. In this algorithm, only cylinders were investigated and not spheres. The ANN was composed of 3 layers; input, hidden and output layer. The input layer was composed of 3 neurons which were the physical characteristics of the cylinder such as the diameter, length, and height relative to the hand. The hidden layer of the neural network contained 24 neurons with sigmoid activation functions.

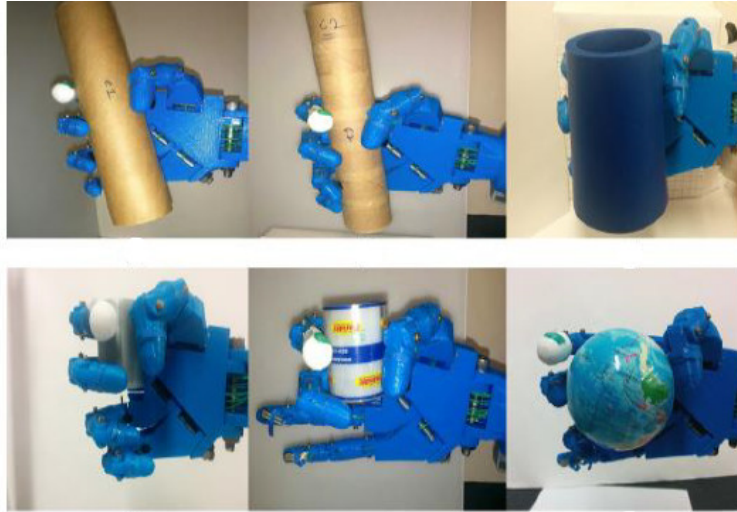


Figure 2.6: Mechanical hand demonstrating grasping simple objects such as cylinders of different sizes and a sphere.

The output layer consists of 5 neurons, each representing each motor for the fingers. Figure 2.7 shows a visual representation of the ANN and its connectivity between layers.

For “learning” to take place, the ANN needs training data. The gradient decent method minimizes error between the actual and calculated outputs and the back propagation algorithm updates the values of the weights. To perform learning 28 training patterns were developed for the algorithm. This was accomplished by measuring the diameter, length and location of the object relative to the hand and having this information as input. Afterwards, the fingers were manually closed onto the object until there was a secure grip⁹.

The input to the ANN is the object data set and the output is the required motor angles for grasping. The object data set consisted in object dimensions and relative position to the hand. Output of the ANN are motor angles required to grasp

⁹The observer would judge when the grip would be secure based on how well the object could be held in the artificial hand without it falling.

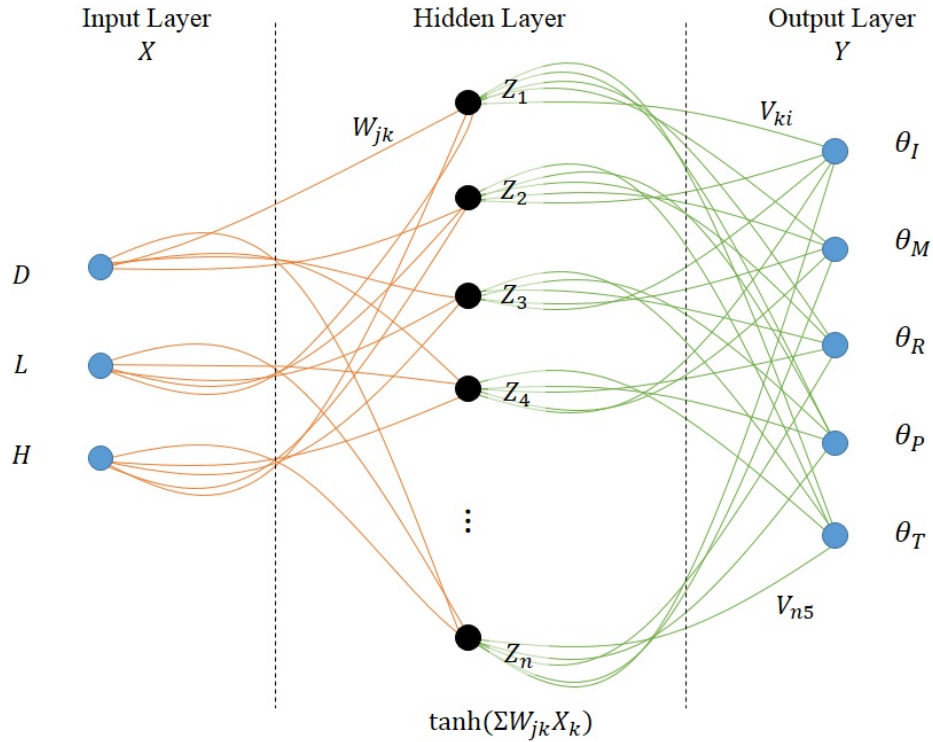


Figure 2.7: Visual of ANN. Note that the layers are fully connected and there is no direct communication between input and output layer.

the object. This set of motor angles is what is termed *grasping pattern*, the state of the hand which relates to a secure grasp over a desired object. As can be imagined, objects of different shapes and sizes would require different grasping patterns [63] [13].

There are a number of observations which are common when training ANNs. As the number of iterations for train the ANN increased, the resulting Mean Square Error (MSE) decreased. Likewise, as the iteration count increases, the algorithm will be able to, in theory, find a set of values that will minimize error for the next step. Likewise, another observation made was that as the number of neurons in the hidden layer increased, so did the MSE. The reasoning behind this is that theoretically, neurons help connect patterns in data, since every neuron is associated with at least two

weights. Therefore, if there are many neurons, in theory, there are more variables that the back propagation algorithm, in conjunction with the gradient descent method, is able to fine tune for optimization.

Another observation is the number of neurons in the hidden layer. As mentioned, there are 3 input neurons because of the object characteristics and there are 5 output neurons because there are 5 motors that move the fingers. However, selecting how many neurons are in the hidden layer is not as simple as the input and output layers. Every scenario calls for a different number of neurons in the hidden layer, and there are no suggested number of neurons. However, a key is to start off with a small number of neurons and increase to minimize the resulting MSE. At the same time, it is key not to increase the number of neurons *ad infinitum* because there are risks of long computation time, redundant neurons and memorization. In this analysis, the number of neurons began at 5 and increased till 24 to minimize MSE and at the same time not strain the computation time which takes about 3 seconds for computing the resulting output motor angles.

To test the efficacy of the ANN, two objects that were previously used to train with were provided as training data. Therefore, the user knew what the motor angles should be in order to securely grasp the object. What this demonstrated is that indeed, the ANN is able to give accurate predictions based on object characteristics. As shown in Table 2.1, for the two cylinders, C1 and C2¹⁰, the motor angles, per finger,

Table 2.1: Comparison between ANN output motors angles and actual motor angles, [1].

Object	Actual Output (degrees)					Predicted Output (degrees)				
	θ_I	θ_M	θ_R	θ_P	θ_T	θ_I	θ_M	θ_R	θ_P	θ_T
C1	53	105	85	30	70	56	112	81	33	76
C2	81	110	80	38	56	83	109	79	37	59

from the ANN and the experimental results are compared. The maximum angular difference for cylinder C1 is 7 degrees and the maximum difference for cylinder C2 is 3 degrees. Taking into account that servo motors used, neither of these maximum differences are significant¹¹. Moreover, in both cases, the artificial hand was able to securely grasp both objects.

A more convincing test of the ANN was to provided the algorithm with an unknown object which had not been introduced during training. This would demonstrate that unknown objects, of similar shape, are able to be handled by the algorithm to find the appropriate motor angles for each finger. Therefore, a new object was tested with a diameter of 61mm, length of 147mm and a relative height of 165mm.



Figure 2.8: Time lapse images of the artificial hand grasping the unknown object based on input characteristics, [1].

As observed in Figure 2.8, the ANN was able to generate the appropriate motor angles to each finger so that the *unknown* object may be grasped securely. Nevertheless, it is important to mention that the input data works best when the data is within the range of the training data, i.e. interpolation as opposed to extrapolation.

¹⁰C1 has a diameter of 51.5mm, length of 197mm and a relative height of 139.7mm. Similarly, C2 has a diameter of 40.5mm, length of 202mm and a relative height of 139.7mm

¹¹Servo motors control is crude, therefore, anything less than 4 degrees of rotation is not distinguishable.

For instance, if the object were to have a diameter beyond the grasp capabilities of the hand, then a resulting non-achievable angle might be sent to the motors, which could result in damage or instability of the system.

2.3 Improved Biomimetic Thumb

With the assistance of a senior design team¹², a more dexterous thumb was developed. While working with the Inmoov artificial hand, it was observed that grasping space was limited. For instance, when the hand would close, the thumb would only be able to contact the index finger. The actual human thumb is able to touch the tip of every other finger, which this ability is advantageous when it comes to manipulating and grasping. Therefore to improve the dexterity of the thumb, a new thumb was designed, fabricated and assembled.

Figure 2.9 shows the improved thumb with more grasping range and control. The human thumb is critical in grasping operations, therefore the improvement will help in controlling the grasping process. The thumb represents 40% of the functionality of the human hand [82], hence the importance of the thumb. The base joint of the thumb is called the carpometacarpal joint (CMC) and the next joint is called the metacarpophalangeal joint (MCP) and then the interphalangeal joint (IP). The CMC joint has two degrees of freedom and the MCP and IP joints have a single degree of freedom [83]. The ability of the thumb to oppose the other fingers in motion is a major characteristic of the thumb for grasping, which if it were not available, grasping could not be accomplished as easily. Therefore, the design of the thumb aimed to mimic the joint lengths and rotational ranges.

¹²The 2016/17 team consisted of senior students in mechanical engineering: chief engineer - Eddie Ordonez, engineering specialist - Aaron Romero, engineering quality control - Samantha Meeks, engineering R&D - Britton Sanders

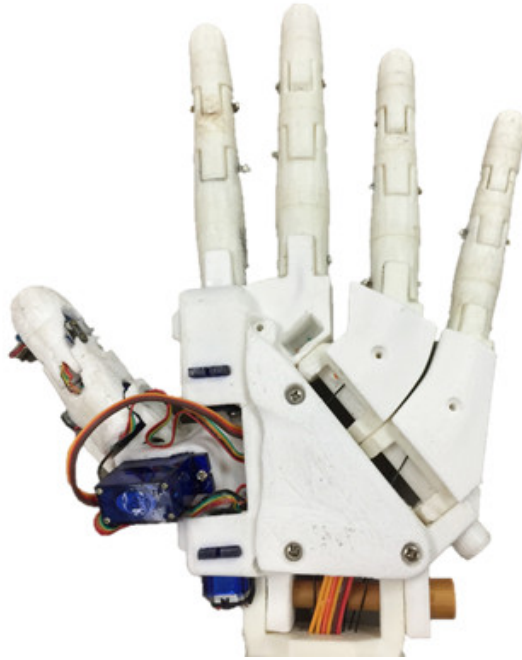


Figure 2.9: Improved design of the mechanical thumb (view of the back of the hand).

The previous artificial hand was able to have a maximum grasp of about 79mm, but with the improved thumb, the maximum grasp is about 88mm, a size improvement by about 11%. The previous thumb was under-actuated, see Section 2.1, in other words, one tendon cable was employed to move 3 joints. Moreover, the control of the individual joints of the thumb was not possible. With the pull of the cable, from the motor, each joint is actuated when the previous joint reaches its mechanical limit¹³. However, the improved thumb has the ability to move each joint individually by commanding of separate motors, since each joint has an associated motor which controls its angular rotation.

¹³Relative to the palm of the artificial hand, the furthest finger joint bends until it reaches its geometric limit. The distal joint bends until it reaches its mechanical limit, then the intermediate joint bends until it reaches its limit and finally the proximal joint bends until it reaches its mechanical limit.

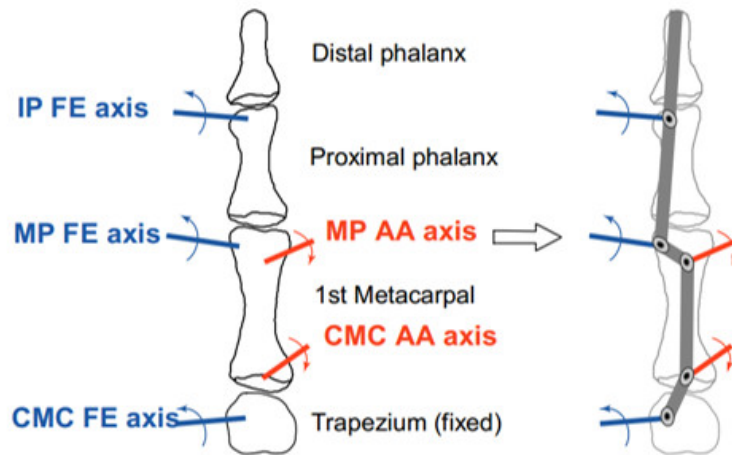


Figure 2.10: Thumb joint kinematics [2].

Figure 4.1 shows the naming convention of each motor in the thumb. Due to the fact that the motors are not used to pull cables to tension the finger, they have to be incorporated near the joint location to use the body as a rotation link.

It is important to mention that θ_{T1} consists of two coupled motors since it was assumed that this joint would have to provide the highest torque for grasping. After the motors were embedded into the finger joints, free weights were used to determine how much output torque the motors could give. By supplying 7V of power and about 31mA of current, the stall torque was 1.7kg/cm.

With the addition of the new thumb, control as well as dexterity, were improved for grasping. The new thumb allows for precise grasp, spherical grasps, cylindrical and planar grasps.

2.4 Machine Learning Object Recognition Sans Vision

Another front of research was the use of machine learning algorithms to improve grasping understanding. This research was mentored by Christopher Abrego and the

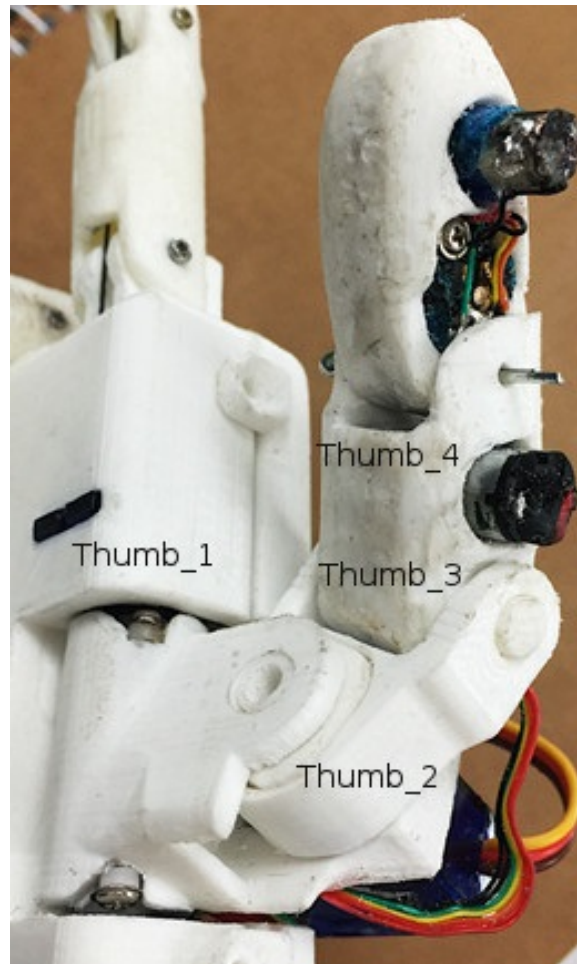


Figure 2.11: Thumb with improved mobility. Naming convention for each motor on the thumb starting from the base of the thumb, moving toward the tip of the thumb.

mentee was Saul Gutierrez¹⁴, funded by the McNair Scholar program at UT Arlington. Research in this section was performed with the original thumb and not the improved thumb.

Tele-manipulation is an interesting and important modality of HRI research. The glove allows the user to perform tasks in remote locations without having to be in the location of the operation. One of the motivating questions in this line of

¹⁴Saul G. is an undergraduate senior in the Computer Science Engineering department at UT Arlington.

research was object identification in the absence of a vision system. This challenge was approached by using a tele-manipulation glove, presented in Figure 2.12, and various machine learning algorithms.



Figure 2.12: Image of the user grasping a sphere while using the interaction glove.

The objective was to have the user grasp an object while wearing the glove and have the system correctly identify the object and its dimensions. Figure 2.13 demonstrates that the algorithm should first distinguish between spheres and cylinders and based on that, decide which algorithm would be used to predict the dimension of the object (either sphere or cylindrical). As a proof-of-concept, simple objects such as spheres and cylinders were used for test grasping. Then different machine learning (ML) algorithms were investigated such as Support Vector Machines (SVM), Artificial Neural Networks (ANN), and K -Nearest Neighbors (KNN). The software platform used was LabVIEW, where these ML algorithms were programmed.

Among one of the questions of interest was grasping methods that could be mimicked by the artificial hand developed. The artificial hand could not match the dexterity of the human hand. It was chosen, for the purpose of the research to constrain the human hand to better match the performance of the artificial hand.

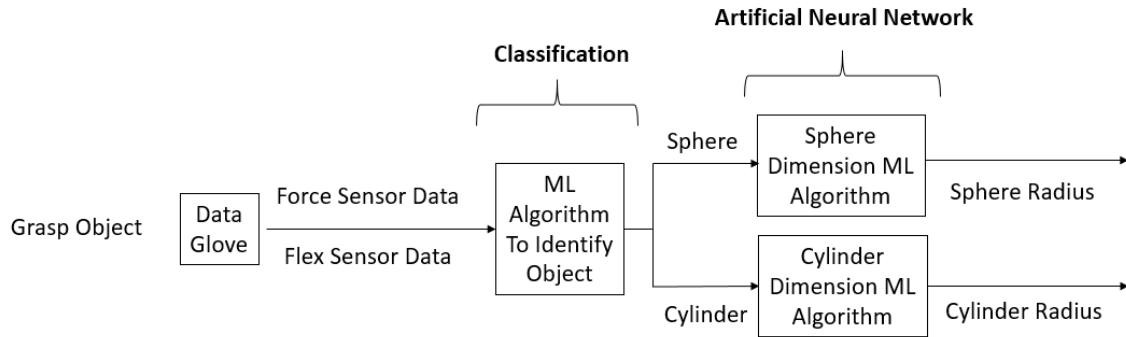


Figure 2.13: Algorithm flow used to identify objects.

Different people can take different approaches at grasping objects, so 3 apprehension methods were identified in order to standardize and be able to get repeatable results. One approach, *Method 1* was not to have the object come in contact with the palm of the hand, instead, focus mainly on the finger tips for contact. *Method 2* consisted in specific placement of the fingertips around the object¹⁵. *Method 3* considered the user to first approach the object and make contact with the palm of the hand. Then, naturally wrap fingers around the object to get a grip. *Method 4* tried to mimic the dexterity of the robotic hand by preventing abduction and adduction¹⁶ of the base joints of the fingers.

The voltage from each flex sensors was captured with the NI myRIO and analyzed to help distinguish between spheres and cylinders using different machine learning algorithms. A statistical *t*-test was performed to determine the similarity of the grasping methods. However, visual and experiential analysis was also used to determine which grasp method was most repeatable and comfortable for the user. Based

¹⁵For spheres, place fingers around the largest cross sectional area plane. For cylinders, place fingers on a longitudinal plane.

¹⁶Abduction is the motion of a limb away from the body, adduction is the motion of a limb toward the body.

on how repeatable the method and how well the artificial hand could reproduce the motions, *Method 4* was chosen to carry out the experimentation.

Once the grasping method was determined, training data was collected. The user was asked to grasp various objects and the resulting voltages from the flex sensors were manually recorded in a spreadsheet, later to be used as inputs to the machine learning algorithms. For classification, cylinders were given the classification ID as 0 and spheres were given the classification ID as 1. Therefore, data patterns look like the following example, [0.5, 0.6, 1.2, 0.3, 0.5, 1]. Where the first 5 entries are the voltage values from the flex sensors from each finger and the 6th entry is the appropriate shape classification. The total set of data patterns collected was 58. Ideally, it is beneficial to have as many training patterns as possible, as a rule of thumb over 100 patterns [84] [85], however there are only a limited amount of objects that would be graspable which would result in significantly different data¹⁷.

As mentioned in the beginning, the machine learning algorithms explored were ANN, KNN, and SVM. All three algorithms were programmed in LabVIEW to accept a spread sheet with training data, formatted as previously mentioned. The output of the algorithms were to indicate the shape of the object, in other words give the shape classification (0 or 1). The efficiency of the algorithms was determined by the ability to classify a object correctly or not.

Artificial Neural Networks is an algorithm attempting to mimic the biological functions of the brain. The algorithm is provided with training data and the neurons of the network. Weights in the network are trained so that the provided outputs can be calculated by the inputs. Optimization techniques, such as back propagation and gradient decent, are used in the process of minimizing error between the actual

¹⁷The difference in output voltage between grasping a cylinder that is 33mm in diameter and a cylinder that is 35mm in diameter is not significant.

output and the training data output. The parameters that the user defines when creating an artificial neural network are the number of layers, hidden layer nodes, node connections and iteration number.

Support vector machines (SVM) are algorithms that find linear models to separate classes of patterns [86]. The goal of the SVM algorithm is to design a hyperplane that separates data into *independent* and distinguishable sets. As a result of this separation, the hyperplanes calculated help distinguish distinct sets of data. The data that has similar patterns form into groups. Therefore, SVM was chosen to help classify objects.

K -Nearest Neighbors (KNN), similar to SVM, seeks to group similar patterns. Given a new pattern, the algorithm takes into consideration the distance between the new pattern and its k nearest neighbors, where k is defined by the user [87] [88]. Based on this distance (Euclidean, Mahalanobis, Weighted, etc) either closest neighbor or majority vote decides on the new patterns classification.

In order to identify which method would perform best, training and testing data were developed from the number of available training patterns gathered. There were 3 training sets as well as 3 testing sets¹⁸. Using the KNN algorithm, tests using closest neighbor method and majority vote method (3, 5, and 7 votes) was carried out, see Table 2.2. Likewise, test results of for ANN and SVM as shown in Table 2.3.

To determine which algorithm was best suited for classification, the following observations were made. Using the ANN, the resulting output would not be binary (0 or 1), but because of the summation technique that the ANN uses, the output was a number between -0.25 and 1.25. It was decided, based on observation, that the numbers that fell between -0.25 and 0.25, would be rounded to 0. Similarly, number

¹⁸Test set 1 contained 14 spheres and 12 cylinders (26 total), Test set 2 contained 7 spheres and 6 cylinders (13 total) and Test set 3 contained 17 spheres and 15 cylinders (32 total).

Table 2.2: K-Nearest Neighbor test results. Percent accuracy based on method Closest or Majority Vote (MV).

Test Set Number	Closest	MV 3	MV 5	MV 7
1	100	96.2	96.2	96.2
2	92.3	92.3	100	100
3	90.6	84.4	87.5	84.4

Table 2.3: Test result for the ANN and SVM algorithms. Percent accuracy based on both methods.

Test Set Number	ANN	SVM
1	100	100
2	100	100
3	91	91

that would fall between 0.75 and 1.25 would be rounded to 1. This decision is difficult to justify, so using ANN as a classifier was discarded. It is also important to note that ANN are more suited to give variable values, as opposed to binary values.

Both KNN and SVM are naturally classification algorithms, therefore, they are more appropriate for this application. Based on the accuracy in Tables 2.2 and 2.3, it can be concluded that the SVM algorithm has a better ability to accurately distinguish between cylinders and spheres. Therefore, SVM was chosen to be the algorithm for the first part of item identification.

After an object is classified, the next step is for the algorithm to determine the diameter of the object. For this step, the ANN machine learning algorithm is chosen, since both the SVM and KNN algorithms are naturally more focused on classification. The ANN developed had 1 hidden neuron, used sigmoidal activation functions, and 6000 iterations were used. The average diameter difference between the actual and predicted is about 5%, which puts confidence in the ANN developed.

Improvements in this system could be performed by increasing the number of iteration and number of hidden nodes, however, there would be more time spent

on computation. Another general improvement would be to improve the interaction glove by potentially incorporating more sensors. At this point, only flex sensors that capture the bending of the fingers were used. However, it can be conceived that including force sensors as well can help determine which fingers are exploited when grasping spheres in cylinders.

CHAPTER 3

Grasp Analysis and Performance of First Iteration Hand, H1

3.1 Grasping Objects

In this chapter, research in the investigation of grasping with H1, the first iteration hand developed, is discussed. Furthermore, the training/testing process and the procedure for predicting object dependent grasp patterns is discussed. The grasping algorithm is testing in both software and hardware to determine the success of the predicted grasp pattern. The experimental procedure for grasping is outlined in Figure 3.1.

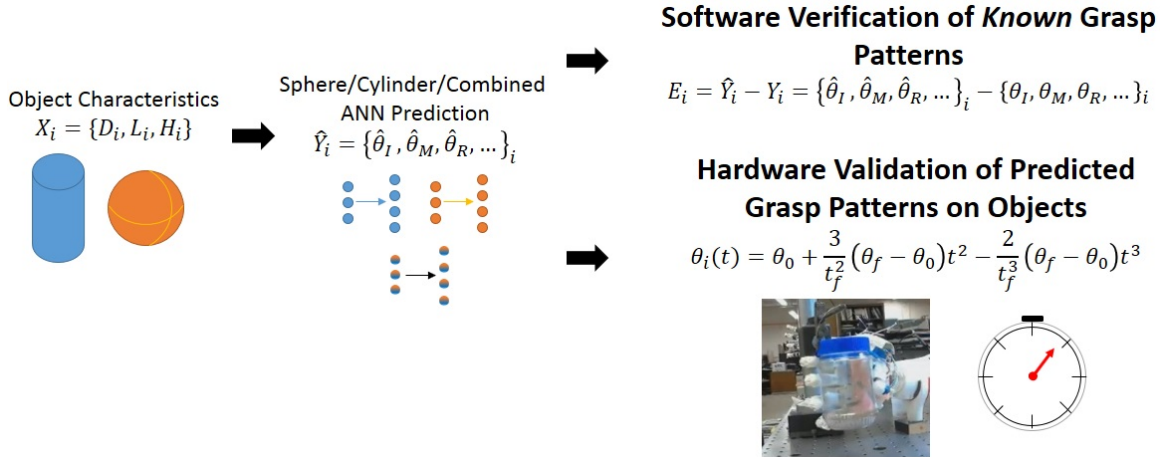


Figure 3.1: Procedure for grasping considering both software analysis and hardware validation.

The first step in the procedure is to gather the object characteristics such as the object diameter, length, and shape category (sphere, cylinder or neither) as well as the relative height of the object with respect to the hand. The next step is to perform

a grasp pattern prediction based on the appropriately trained ANN. Afterwards, if the grasping pattern is known, in our software analysis, the prediction will be compared to the actual grasp pattern to evaluate the accuracy of the prediction. Then, during hardware verification, the developed hand (H1) is controllably commanded to move to the predicted grasp pattern onto the object and observed if the grasp is successful or unsuccessful. Therefore, there are two validation processes to evaluate the performance of the prediction algorithms.

A variety of household convex and assumed rigid objects were used to grasp by H1, among which were spheres and cylinders. In general, the objects used to grasp are rigid and some, to a degree, are slightly compliant (such as the insulation koozie, object C3). A total of 21 cylindrical objects and 7 spherical objects used to investigate grasping. Table 1 in the Appendix presents all the objects and dimensions (L_o and D_o) along with normalized dimensions (G_r and L_r , see Equations 3.1 and 3.2) with respect to the hand. There are two dimensions used to characterize every object, which are a vertical distance length, L_o , and a horizontal distance, D_o . The normalized ratios (G_r and L_r) are important because they are scalable to different size hands and objects, hence results can be generalized to similar structured hands. Note that the denominators of Equations 3.1 and 3.2 are dependent of the hand used for experimentation. In this case, D_h is 3.7" which is the largest diameter object that can be held with the hand. Moreover, L_h is the vertical length of the palm, which is 3.75".

$$G_r = \frac{D_o}{D_h} \tag{3.1}$$

$$L_r = \frac{L_o}{L_h} \tag{3.2}$$

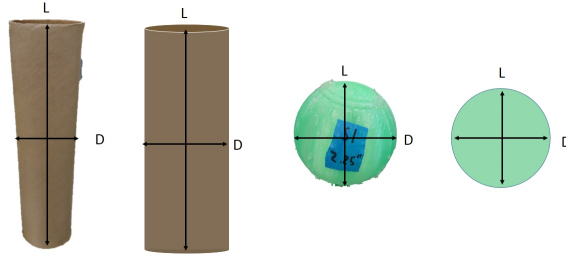


Figure 3.2: Image of objects with their dimensions marked which are used for the prediction algorithm.

Equation 3.1 normalizes the diameter of the object, D_o , with the total grasp diameter of the hand. This term is called the grasp ratio, G_r , and all objects in the study will have a grasp ratio less than 1.0. If a grasp ratio were larger than 1.0, then the object would not be grasped by the hand because its diameter has exceeded the grasp potential of the hand. Similarly, Equation 3.2 normalizes the object length, L_o , with the span of the palm of the hand. This term is called the length ratio, L_r . When the length ratio is less than 1.0, the length of the object is smaller than the palm span of the hand (see Figure 3.3). If the length ratio is greater than 1, then the object will be longer than the palm span of the hand. Both G_r and L_r are important since they determine how the hand will grasp the object and what fingers will be used to grasp it.

The 28 objects were grasped at 4 different locations relative to the hand height. Note that the objects were grasped twice. Figure 3.4 presents cylinder and sphere dimensions (Figure 3.4 a) and the relative object dimensions compared to the hand (Figure 3.4 b). It is noticeable that there are no objects with diameters greater than 3.7” because the hand is not possible to grasp these. It is also observed (Figure 3.4 a) that the object with the smallest length is about half the length of the palm and the thinnest object is a little less than a third of the grasp diameter of the artificial hand H1.

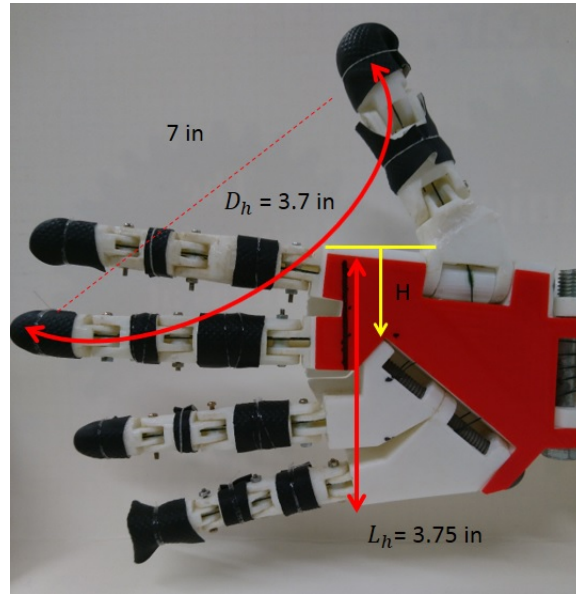
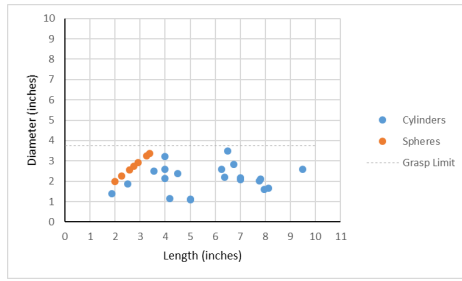


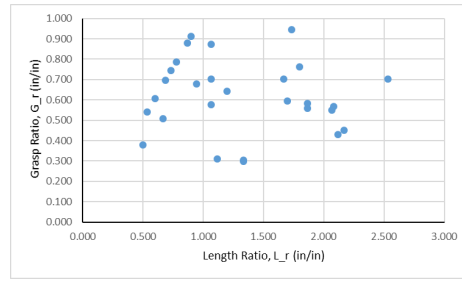
Figure 3.3: Dimensions of the artificial hand including the palm height and the arc grasp. Note, that the largest object that the hand can grasp has a diameter no larger than 3.70". This was found through experimentation. The yellow arrow indicates the positive direction of the height.

Every object grasp was recorded as well as the front panel of the program (Figure 3.5) used to actuate the motors. Each grasp video was on average about 30 seconds long, therefore in total 112 minutes of grasp recordings. The angular motor values necessary to have the fingers grasp the object were noted and entered into a database.

Among some of the preliminary observations of the grasp data is that objects can be grasped with a variety of fingers with different poses. For instance, an object can be grasped with 3, 4, or 5 fingers (including the thumb) depending on the relative height of the object with respect to the hand and depending on object size. Notation that is used to distinguish different fingers used are: T - thumb, I - index, M - middle, R - ring, P - pinky. So, for instance, "TRM" indicates that the object was grasped



(a) Survey of all the objects used for grasping.



(b) Survey of the objects that have the relative heights to the dimensions of the artificial hand.

Figure 3.4: Spheres and cylinders that are used for grasping.

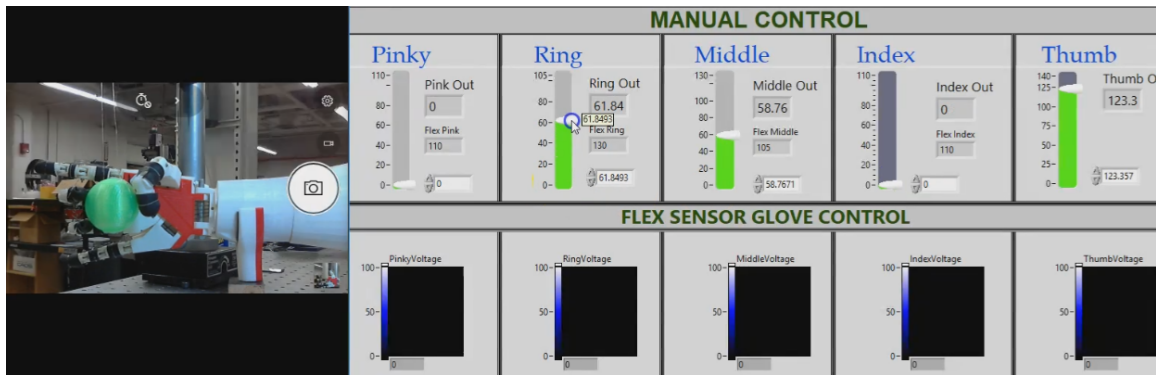
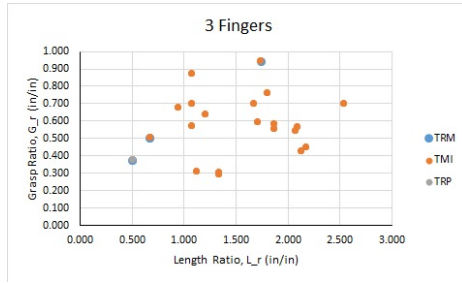


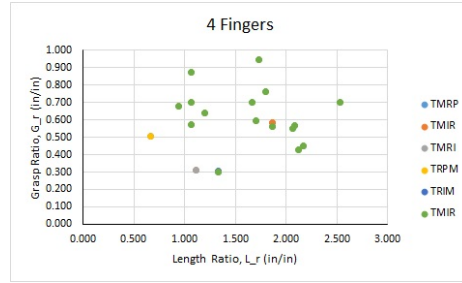
Figure 3.5: Image of the computer with the real-time video feed of the artificial hand on the left hand side, and the front panel of the LabVIEW VI used to control the actuation of the servo motors for grasping.

with a thumb-ring-middle finger combination. For cylinders, Figure 3.6 indicates how objects with different dimensional ratios are grasped with various finger combinations.

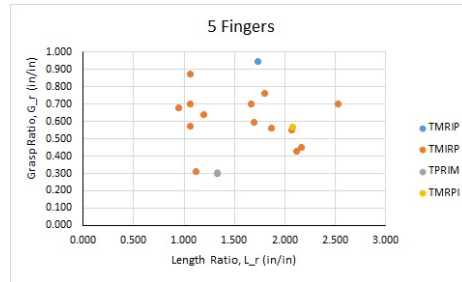
It can be observed from Figure 3.6 that all objects were grasped with 3 fingers, particularly using the combination of TMI, TRM and TRP. Note that the TRP grasps were done on the objects that had low L_r and G_r , which makes sense since this combination of fingers would be used to grasp small objects. Moreover, 5 fingers were used to grasp objects, except for two occasions, particularly because the L_r value was less than 0.8. The reason for this is that the object has a length less than the palm span of the hand ($L_r < 1$), then not all 5 fingers will be required to grasp it.



(a) Grasps made with 3 fingers.



(b) Grasps made with 4 fingers.



(c) Grasps make with 5 fingers.

Figure 3.6: Distribution of grasps of cylinders with with different finger combinations respect to their relative size, G_r and L_r .

In Figure 3.7, various grasp combinations with different number of fingers are presented. The objects grasped with TRP and TRM combination are on the lower side of the G_r and L_r ratios, which corresponds to small objects. One thing to note at this point is that the height location of the object has not been mentioned yet, however, it will be addressed in further sections. Considering 4 finger grasps, objects that have L_r less than about 0.75 do not appear. The reasoning behind this is that smaller objects require less fingers to grasp objects.

The type of finger and number of fingers required for grasping is also dependent on the relative height of the object with respect to the hand. This is important because depending on the object dimensions and height, different fingers and at different motor angles will be used. It can be observed in Figure 3.8 the patterns of the objects grasped at different heights. Note that all objects were able to be grasped at all heights, but

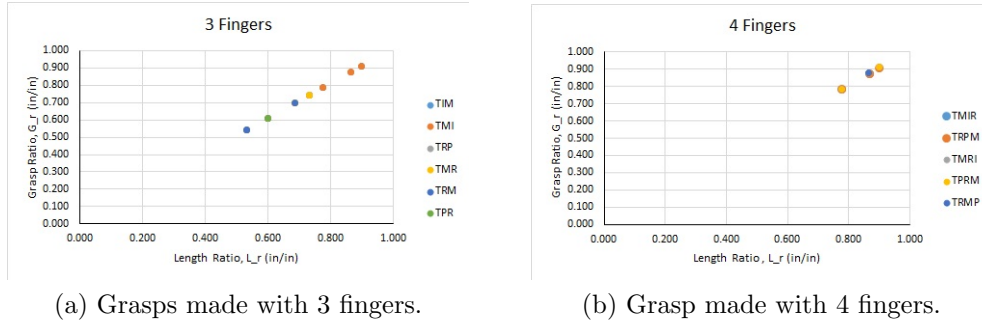


Figure 3.7: Distribution of grasping spheres with different finger combinations with respect to their relative size, G_r and L_r .

not all grasps were performed at all heights. Therefore the same data points are found on every plot in Figure 3.8, but sometimes with different fingers for grasping.

It is observed that all objects are grasped at 1" height and only 3 fingers were used. This reasonable since at 1" height, only the thumb index and middle finger can be used for grasping. Similarly, at a height of 1.5", only 3 fingers are used to perform all the grasping. However, once the height increases to 2.5", the middle finger is employed to grasp objects. At this height, 3 fingers are still used, however, it can be noted that for cylinders that have $L_r < 0.75$ and $G_r < 0.5$, indicating that the cylinders are relatively small, which can be visualized in Figure 3.9. The distribution of grasps of cylinders at a height of 3.5" can be observed. Once again, relatively small objects have grasps that can be made with 3 or 4 fingers, where as the majority of the grasps are made with all 5 fingers. In Figure 3.10, the comparison of the objects grasped with different number of fingers is shown for different heights. As the height increases, from 1.5" to 2.5", the number of fingers used for grasping changes from 3 to 4, this being because more fingers are "required" to grasp. However, when the height increases from 2.5" to 3.5", the majority of the grasps are now performed with 5 fingers, however, there are some grasps that are performed with 3 and 4 fingers. The

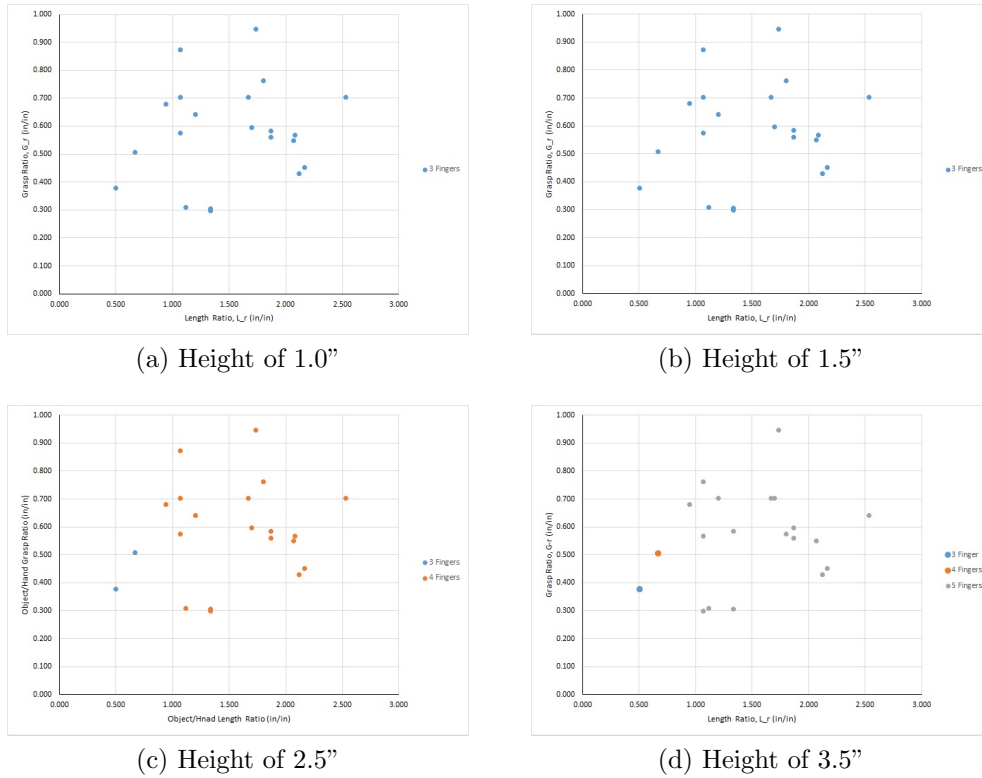


Figure 3.8: Different cylinders grasped at various heights.

reason behind this is that some of the objects have a small height, L_r in the range of 0.5 - 0.7, thus not requiring all 5 fingers to grasp.

Now that an analysis of the different grasps required to hold cylinders, with respect to relative dimensions and height, the same analysis is realized for the spheres. The distribution of grasps at different heights is presented in Figure 3.11. It can be noticed that all the objects appear throughout the different heights. At height 1" and 1.5" all objects are grasped with 3 fingers. Then, at 2.5" and 3.5", larger objects with L_r and G_r values greater than 0.75, are grasped with 4 fingers and smaller objects with G_r and L_r less than 0.75, are grasped with 3 fingers. Figure 3.12 indicates the percent distribution of the number of fingers used for grasping spheres are different

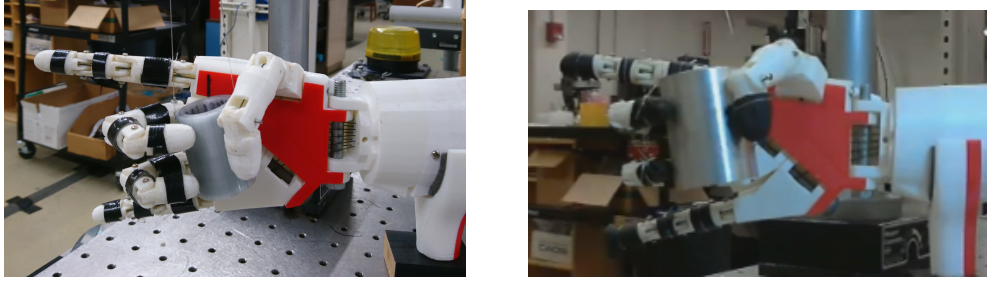


Figure 3.9: Cylinders that are grasped with 3 fingers considering them at a height of 2.5 inches. Note that the objects grasped are small and only require 3 fingers for grasping.

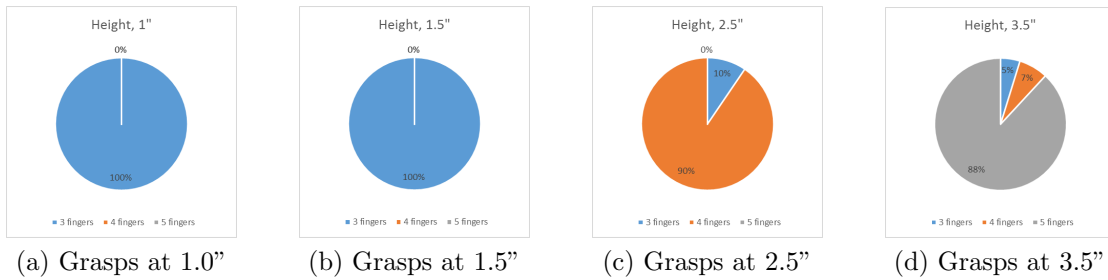
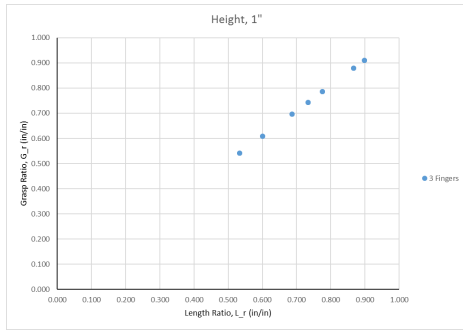


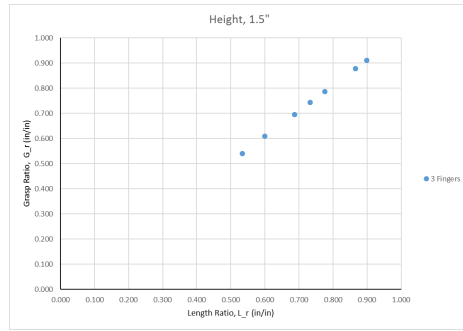
Figure 3.10: Pie graphs of the total fingers used to grasp cylinders at different heights.

heights, indicating that 3 fingers is the most used posture for the hand. This could also be because $4/7$ spheres used to experiment have G_r and L_r less than 0.75.

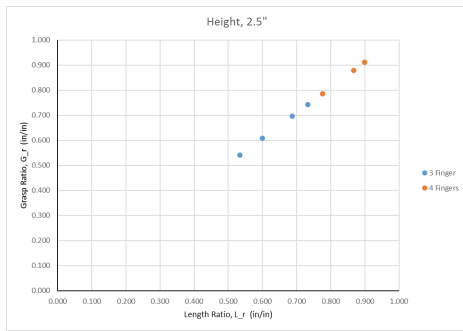
Some general observations can be made based on both grasping spheres and cylinders. Relative height of the object being grasped is important, this is especially true if the L_r and G_r are below a certain range, namely around 0.75. The order in which the appropriate fingers close is also notable, namely, all grasps require the thumb to close to a certain position before other fingers actuate. The second finger to close is usually the middle finger which pushes the object against the fixed thumb. Some noticeable differences between grasping cylinders and spheres is the dimensional ratios with respect to the finger count for grasping. Since the sphere will have L_r and G_r relatively close in value, and G_r cannot exceed 1, therefore, spheres are not grasped with 5 fingers. For instance, spheres closer to $G_r = 1$ will, in theory, be grasped with



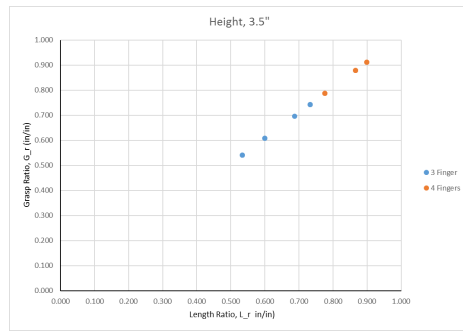
(a) Spheres that grasped at 1.0”.



(b) Spheres that grasped at 1.5”.



(c) Spheres that grasped at 2.5”.



(d) Spheres that grasped at 3.5”.

Figure 3.11: Number for fingers used to grasp different spheres at various heights.

all 5 fingers, however, spheres that approach $G_r = 1$ will not be grasped and as a result not all fingers will be able to be used.

This analysis is performed prior to developing an Artificial Neural Network (ANN). Trends and patterns observed will help determine relationships between object and grasp which will then be used to reinforce predictions and forecast which fingers are used to grasp various objects at different heights. For instance, objects can be categorized by their G_r and L_r , therefore not depending strickly if the object is concretely a sphere or cylinder. If the general shape of the object closely conforms to a sphere, then the G_r/L_r ratio will approximate 1. This is because the G_r and L_r will be very similar. The following, is a list of observations for quick reference:

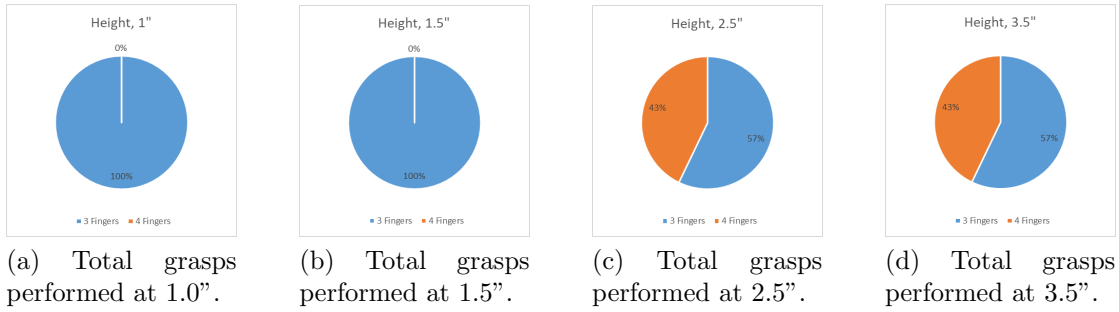


Figure 3.12: Pie graphs of the total finger used to grasp spheres at different heights.

- Objects that have G_r and L_r similar to one another ($G_r/L_r \approx 1$) can be considered spheres.
- At height 2.5" and 3.5", if a sphere has a G_r or L_r value less than 0.75, will only require a 3 finger grasp.
- (*The converse of the previous point*), at a height of 2.5" and 3.5", if a sphere has a G_r or L_r value greater than 0.75, will require a 4 finger grasp.
- No spheres can be grasped with 5 fingers.
- Cylinders with L_r less than 0.7 cannot be grasped with 5 fingers at height 3.5". Similarly, at the height of 2.5" cylinders cannot be grasped with 4 fingers (only grasp with 3 fingers).

Other benefits from this analysis is decision making after ANN prediction¹. Take for instance the grasp prediction of a object with both G_r and L_r less than 0.7. Based on what is observed, if $G_r/L_r = 1$ then the object can be closely considered a sphere. Furthermore, if G_r and L_r are less than 0.7, then 3 fingers will likely be the amount of fingers used to grasp the object. Now, if the output of the ANN gives a finger count of 4 or 5, then we know that some fingers will not be necessary. A decision will then be made on which 3 fingers will move and which will remain

¹The grasp needed for the object will be determined by the prediction of the Artificial Neural Network

stationary. This decision will be made based on the trends observed with respect to the height. As noted previously, at heights 2.5” and 3.5”, only 3 fingers are used, and only the thumb, middle, ring, and pink fingers are used, eliminating the use of an index finger. Observations from patterns like the one just mentioned will be incorporated to the post-processing of the ANN outputs for data conditioning.

3.2 Grasp Observations

Another observation is that grasping was improved with the addition of a high friction surface placed at various points of contact that the artificial hand makes with objects. These locations are at the finger tips and on the middle of all the finger joints. The material used for grip was a standard glove worn by people to wash dishes. Another observation is that the tip of the thumb does not come into contact with the objects. Instead, the last joint of the thumb contacts the object. Hence, even if the tip of the thumb were to be covered with grip material, grip of the thumb on the object would not improve. Therefore, the joint was covered with grip material and contact with the grip surface was easily made with the object.

Repeatability is low and the area of contact is not always the same, even for the same object. The reason is that the fingers are under-actuated to start with, where one motor controls the actuation of 3 joints. Moreover, the actual motion is not as repeatable as if there were direct control over all the joints in the finger. In theory, when actuating, the next joint will begin to actuate when the last joint has reached its mechanical limit of motion. The actuator will continue to pull the actuation tendon and this will require the next joint to rotate. In theory, this is true for all the joints with the assumption that there is no interference friction on all the joints and no friction between the actuation tendon and the cavities of the finger, and so forth. For





	T_n	I_n	M_n	R_n	P_n
	71.33	43.59	57.87	46.23	34.44
	70.28	48.97	67.56	45.20	0.00
	61.89	45.74	65.51	0.00	0.00
	65.03	50.93	60.41	0.00	0.00

Figure 3.13: Image of the same cylinder (C_1 , $L_r = 2.069$ and $G_r = 0.549$) grasped at different heights. On the right, the normalized close percent values at the respective height. The top image shows the cylinder being grasped at height 3.5” and progresses down to 1” at the bottom image.

these reasons, the motion of the finger cannot be predicted to a degree that will result in a desired contact surface.

Moreover, the finger has limited dexterity and the desired contact surface cannot always be met. The human hand has the ability to flex and extend about multiple axes due to the fact that some joints would be modeled as spherical joints. On the other hand (pun intended) all the joints in the fingers are modeled as purely rotational joints. Therefore, the human hand has more possible poses that could result in a specific contact area. The grasp of the artificial hand does not have the ability to grasp an object with a desired contact surface. Figure 3.14 shows the different contact areas observed based on the grasp ratio (G_r) of the object. When the value of the object starts low, $G_r < 0.5$, the finger tips are used the most. The reason for this

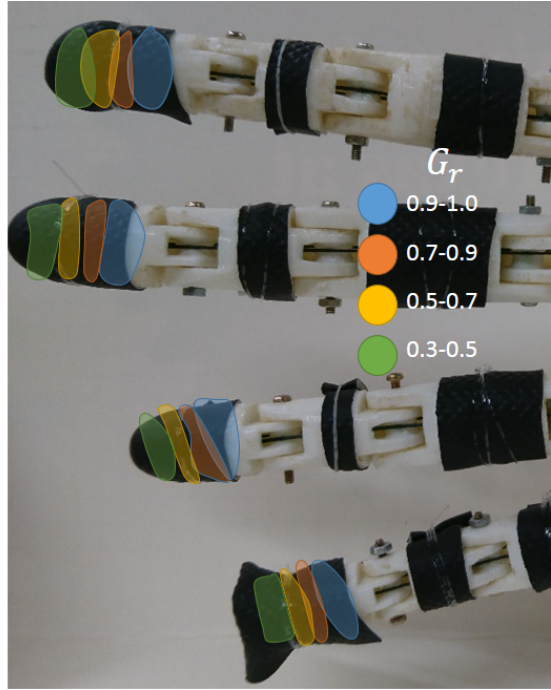


Figure 3.14: Areas of contact on the fingertips with respect to the object G_r . The different colors on the fingertips represent the contact area on the finger for an object with a certain G_r range.

is that the last two joints have completely closed, so that the resulting contact with the object will be made with the tip or near tip of the finger. Note that objects with $G_r < 0.5$ are relatively thin, requiring the fingers to close the most to contact the object. For instance, object C_{11} has a $G_r = 0.310$ and the resulting close percentages at a height of 3.5" are $T_n = 96.43\%$, $I_n = 59.29\%$, $M_n = 63.57\%$, $R_n = 63.57\%$, and $P_n = 62.14\%$. On the other extreme of the objects, those that are grasped with a $G_r > 0.9$ are grasped with an area closer to the base of the joint. The reason for this is that the fingers do not need to close as much in order to contact the object. This is observed in Figure C_{13} which has a $G_r = 0.946$ and the resulting close percentages of the fingers at height 3.5" are $T_n = 58.57\%$, $I_n = 27.14\%$, $M_n = 28.57\%$, $R_n = 35.00\%$ and $P_n = 34.29\%$.

Based on the expressed observations, it can be concluded that object grasping is not repeatable to the degree that the contact area can be accurately predicted on the fingers. The reason for mentioning this is for the outlook of force control for grasping. Force control will incorporate the force experienced on the hand to determine when the hand will stop closing or when the fingers have made contact with the object. To embed a sensor would require the sensor of the finger contact location is guaranteed. However, this does not seem possible, given the low repeatability of contact area on the fingers.

3.3 Artificial Neural Network Analysis

After grasping all the objects, the next step is to implement a machine learning algorithm to predict the desired grasp for both the cylinders and the spheres. Artificial Neural Networks (ANNs) will be utilized to predict the appropriate grasp. The inputs of the ANN will be the diameter of the object (D_o), length (L_o), and relative height (H_o). The outputs of the ANN will be the motor angles; index (θ_I), middle (θ_M), ring (θ_R), pinky (θ_P), and thumb (θ_T).

In terms of software, a three layer ANN will be used for prediction and Back Propagation will be the algorithm used for optimizing weights, input (W_{jk}) and output (V_{ki}). The back propagation procedure updates the weights using the error gradient with respect to the individual weight and a learning factor, η , as shown in Equations 3.3 and 3.4 respectively.

$$W_{jk} \leftarrow W_{jk} - \eta \frac{\partial E}{\partial W_{jk}} \quad (3.3)$$

$$V_{ki} \leftarrow V_{ki} - \eta \frac{\partial E}{\partial V_{ki}} \quad (3.4)$$

The input, X , for the ANN is the object geometry, which is the object horizontal distance (D_o , diameter), vertical dimension (L_o , length). The relative height of the

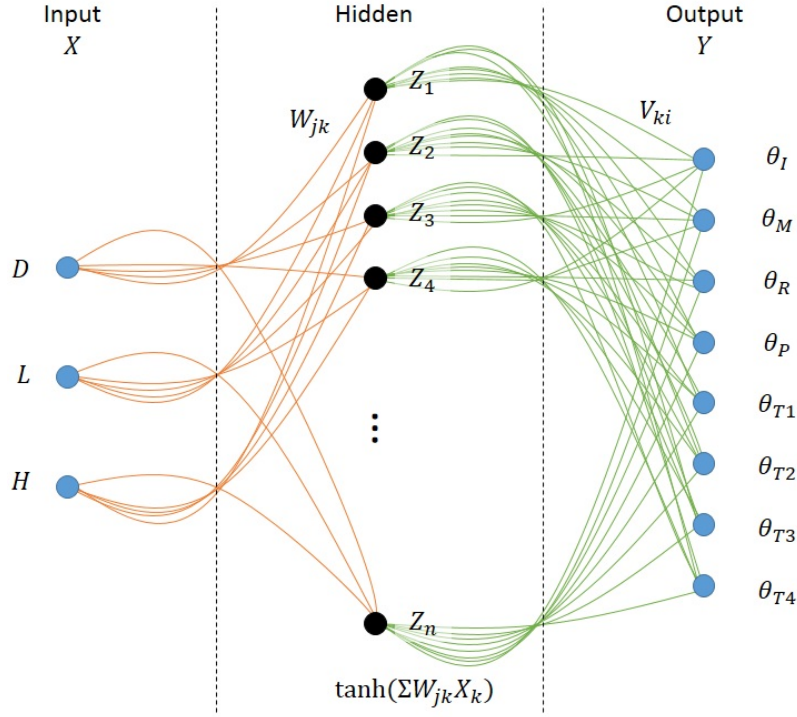


Figure 3.15: Image of the general architecture of ANNs. Inputs are the diameter (D_o), length (L_o), and relative height (H_o) of the object. The outputs are the motor angles index (θ_I), middle (θ_M), ring (θ_R), pinky (θ_P), and thumb (θ_T).

hand (H , height) starts at zero at the index metacarpal location and is positive in the downward direction. This information (D , L , H) is gathered by direct measurement of the object, Equation 3.5.

$$X_i = \{L_i, D_i, H_i\}_{i=\{1,n\}} \quad (3.5)$$

$$Y_i = \{\theta_I, \theta_M, \theta_R, \theta_P, \theta_{T1}, \theta_{T2}, \theta_{T3}, \theta_{T4}\}_{i=\{1,n\}} \quad (3.6)$$

For the training procedure, the object was placed near the hand, and the user controlled the servo motors to a position where the fingers were grasping the object. The object dimensions (and relative height) and the motor angles for all the fingers were recorded in a database as inputs and outputs for each training set. Then, utilizing National Instruments LabVIEW, the grasp data was used to train two separate

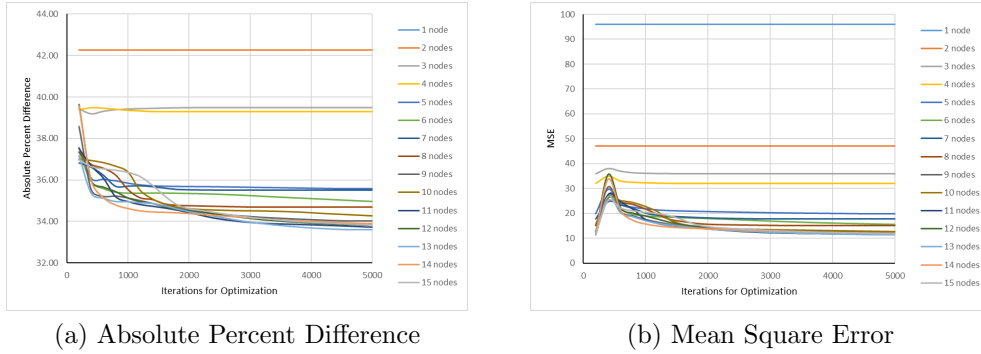


Figure 3.16: The Absolute Percent Difference and Mean Square Error of the resulting ANN architectures when the number of iterations for optimization is varied as functions of number of nodes in the hidden layer. This training was performed with 27 data sets.

ANNs, each consisting of 3 layers, see Figure 3.15, which used back propagation and gradient descent method to optimize the weights, W_{jk} and V_{ki} , between the layers, [89]. The activation function used for the hidden layer is a hyperbolic tangent which compresses the outputs into the range of -1 to 1, Equation 3.7.

$$\tanh(z) = \frac{e^{2z} - 1}{e^{2z} + 1} \quad (3.7)$$

Two parameters needed to be investigated to find the appropriate architecture. The number of iterations for optimization and the number of nodes in the hidden layer are the two parameters that the user can tune in order to reduce the mean square error (MSE) of the resulting ANN. A two layer loop was developed which iterated the number of iterations and the number of nodes in the hidden layer. As shown in Equations 3.8 3.9 3.10, the MSE, absolute percent difference (APD) and the (AD) absolute difference for the overall architecture for the ANN are functions of the number of iterations for optimization (N_I) and the number of nodes in the hidden layer (N_{HN}). These are the main contributors to the resulting metrics of error, and

the major contributors are N_I and N_{HN} , which are also the direct parameters that change the architecture of the ANN.

$$MSE = f(N_I, N_{HN}) \quad (3.8)$$

$$APD = f(N_I, N_{HN}) \quad (3.9)$$

$$AD = f(N_I, N_{HN}) \quad (3.10)$$

3.3.1 Cylinder Prediction

Figure 3.16 gives the results of the different ANN architectures based on the change of number of iterations for optimization and the number of nodes in the hidden layer. On the left hand side of Figure 3.16 shows the resulting absolute percent difference and the graph on the right shows the MSE. In terms of the Percent Difference, the majority values of absolute percent difference begin around the same range (35-38). Then, after the number of iterations increases to 500-1000, the absolute percent difference becomes more sparse among the various node lines. After about 2000 iterations, the absolute percent difference becomes steadily monotonic and some eventually plateau (5-6 nodes in the hidden layer). Eventually, the lowest point of absolute percent difference occurs when 4600 iterations are used for optimization with 13 nodes in the hidden layer.

The MSE of the resulting ANN based on the architecture parameters of the number of iterations for optimization and the number of nodes in the hidden layer. At the beginning, 200-600 iterations, the majority of the node lines increases in MSE. Afterwards, 600 and onward, there is a steady monotonic decrease in MSE value. Just as was observed for the absolute percent difference, the architecture that yielded the lowest MSE was 13 nodes in the hidden layer and about 4600 iterations for optimization.

A question that comes to mind is why the absolute percent difference is too high. Absolute percent difference is calculated the following way,

$$100 \times \frac{|predicted - actual|}{predicted} \quad (3.11)$$

To get a metric of how the ANN is performing holistically, the absolute percent difference is performed on the individual sets. Then, all the absolute percent differences are averaged to give a benchmark of the performance of the architecture based on the number of nodes in the hidden layer and the number of iterations for optimization. However, the average is skewed when the actual value of an angle is 0° . For instance, for object C1 grasped at height 2.5", the motor angle of the pinky is 0. However, the ANN predicted is 0.969° for the pinky². Therefore, using Equation 3.11, the resulting absolute percent difference is,

$$\begin{aligned} 100 \times \frac{|0.969 - 0|}{0.969} &= 100 \times \frac{0.969}{0.969} \\ &= 100 \end{aligned}$$

The resulting absolute percent difference is 100, even though the magnitude difference was off by 0.969° . Therefore, an alternative understanding of the performance of the ANN architectures is viewing the absolute value difference between the ANN prediction and the actual value, as a function of iterations for optimization and number of nodes in the hidden layer. In Figure 2, the absolute angular difference begins around 9-13 degrees in general. Then, after about 2000 iterations, the absolute difference decreases monotonically. Similar to what was found in Figure 3.16, the lowest absolute difference is found with an architecture of $N_{HN} = 13$ with $N_I = 4600$, Cy-13/46.

Given the results from Figures 3.16 and 2, the overall consensus is that an architecture of $N_{HN} = 13$ and $N_I = 4600$ for the ANN will yield the lowest absolute

²The prediction was performed with $N_{HN} = 6$ and $N_I = 3400$.

percent difference, MSE, and absolute difference. Nevertheless, another aspect to consider is the time for computation. In the scenario envisioned, the system will identify an object, then identify the dimensions, and then send those dimensions to the designed ANN. The ANN will then predict, based on the object data, what the final motor angles should be to appropriately grasp the object.

If the ANN were to take a long period of time, for instance 2 minutes or more, to calculate what the final motor angles should be, then the whole grasping process is stalled for a period of time. Case in point, if the number of iterations for optimization or the number of nodes in the hidden layer, requires a large amount of computation (≈ 1 min), then the grasping process will be hindered for a period. Keeping two objectives in mind, process time and error (MSE,APD,AD), the needs to be a balance given that both extremes may not be achieved simultaneously. In this study, optimization was only taken at a maximum of 5000 iterations and 15 nodes in the hidden layer. Additionally, for 9-15 nodes in the hidden later, the difference between 4600 and 2200 iterations is relatively small³.

Apart from the question of optimization of the objectives of time and error, the control algorithm also needs to be considered. The control algorithm will be two-stage, and will consider the grasping process of *approach* and *apprehend*. The fingers first approach the object (*approach* stage) and once contact is made, the fingers embrace the object with a certain amount of force (*apprehend* stage). Therefore, in the first stage, purely kinematic or position information will need to be considered and in the later part, purely force information will contribute to the actualization of the grasp. Hence, related to the topic of deciding the architecture of the ANN, it is proposed that the positional error can be slightly compromised since the next process

³10% change in AD, 2% change in APD and 16% change in MSE

(force control in the *apprehend* stage) will switch to force error control as opposed to position error control.

Keeping both optimization objectives of error and time, it is decided that ANN-9/2600 will be investigated. This architecture should, in theory, be faster to perform computations, compared to 15 nodes in the hidden layer and 4200 iterations for optimization. The difference in errors between Cy-9/2600 and CY-13/4600 is relatively low, as already mentioned.

3.3.2 Sphere Prediction

Similar to the previous section, the same analysis on the training data was performed, however this time to the spheres. There were a total of 7 spheres used for collecting data, which resulting in 28 data sets gathered. Out of the data gathered, 24 of the 28 were used for training and 4 out of the 28 were used for testing. Figure 3.17 indicates the resulting APD and the MSE of the resulting ANN. The value of N_I was varied from 200 - 5000 in increments of 200 and N_{HN} ranged from 1 - 15 nodes in the hidden layer.

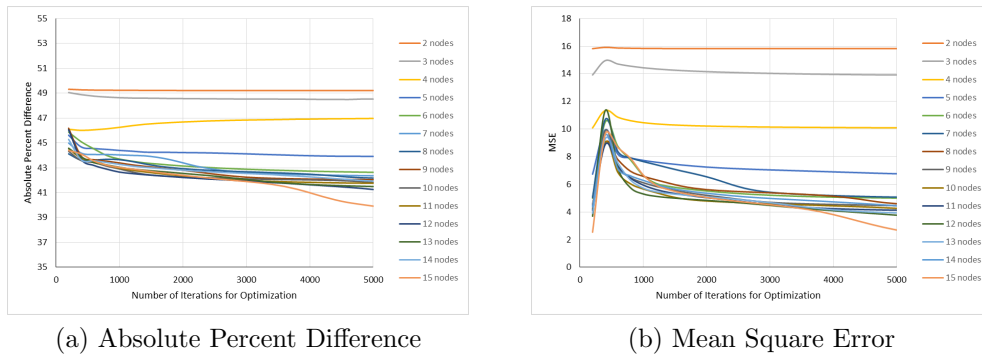


Figure 3.17: The resulting Absolute Percent Difference and MSE of the various ANN architectures. The training was performed with 24 training data sets.

In both the APD and MSE (Figure 3.17) the architecture that yields the lowest error is Sp-15/5000. The majority of the lines of constant N_{HN} decreases monotonically and the change in error flattens out. Figure 3, in the Appendix, shows the average AD for each individual ANN architecture. Likewise, the architecture that results in the lowest error is Sp-15/5000. Note that the error range between the APD and AD (Figure 3). Therefore, there is a clear consensus between all 3 metrics, that the architecture that will yield the lowest error is Sp-15/5000. For comparison purposes, a relatively smaller ANN was selected to compare its predictive accuracy based on similar APD and MSE. Therefore, it is proposed to use Sp-9/3600, where the relative error difference between architectures is small.

3.4 Artificial Neural Network Prediction

3.4.1 Software Implementation

There will be two ANNs used for testing. One ANN will be dedicated to predict the grasping patterns for cylinders, and the other will be dedicated to predict grasping patterns for spheres. It is important to note that the number of training sets for cylinders and spheres are not equal, since there are more cylinders available to grasp, as opposed to spheres. As a result, the predictions made for cylinders may be more accurate as opposed to spheres because of the different training data set size. In the following subsection the results of testing in software are discussed.

3.4.1.1 Cylinders

After training the ANN with 76 data sets, the next part is to test to benchmark the predictive performance of the ANN. There were 4 data sets⁴ used to test the

⁴The four data sets are noted as Test1, Test2, Test3, and Test4. The test is performed on an object, C22, with $D = 2.129''$, $L = 4''$ and Height of $H = \{3.5, 2.5, 1.5, 1\}$ respectively.

ANN. In Figure 3.18 indicates the angular errors between the actual and predicted

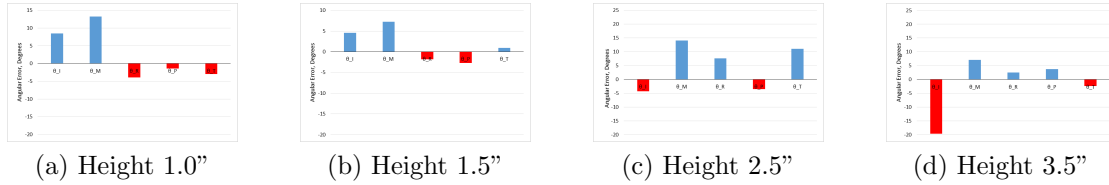


Figure 3.18: Angular errors between the actual and predicted angles for grasping cylinders. The predictions were performed with Cy-9/2600 trained with 76 data sets. Black bars indicate *overgrasping* and red bars indicate *undergrasping*.

angles. From a conservative standpoint, it is preferred that the object be squeeze rather than not contacted. Considering the control architecture, the output result from the approach stage, should leave the fingers in position that has already made contact with the object. The second stage of the controller will focus on the force control of the object, hence the system will begin in a non-null state.

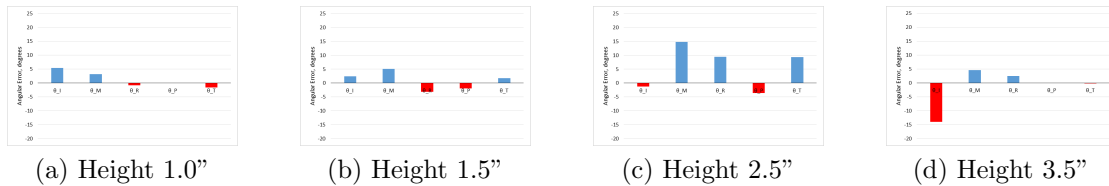


Figure 3.19: Angular errors between the actual and predicted angles for grasping cylinders. The predictions were performed with Cy-13/4600 trained with 76 data sets. Black bars indicate *overgrasping* and red bars indicate *undergrasping*.

For comparison, the ANN was trained with the architecture that would yield the lowest error. Figure 3.19 presents the error between the actual and predictions of motor angles. In general, the characteristics for the angular errors are similar between both architectures, *i.e.* the fingers that *overgrasp* for the first architecture also *overgrasp* in the second architectures, etc. Another observation is that the angular

errors are lower, in general, for the second architecture (Cy-13/4600) compared to the first architecture (Cy-9/2600).

3.4.1.2 Spheres

Training was performed with 24 data sets, for two different architectures; Sp-9/3600 as well as Sp-15/5000. There are 4 data sets which are utilized, Test 1, Test 2, Test 3, and Test 4. As seen in Figure 3.20, the angular errors between predicted and actual based on Sp-15/5000. Test 1 the thumb *undergrasped* and the index, middle, ring and pinky *overgrasp*. In Test 2 the thumb *undergrasped* and the index, middle, ring and pinky *overgrasp*. Then, in Test 3 the thumb, index and pinky *undergrasps* and the middle and ring finger *overgrasp*. In Test 4, the thumb and ring *overgrasp*, while the index, middle and pinky *undergrasp*.

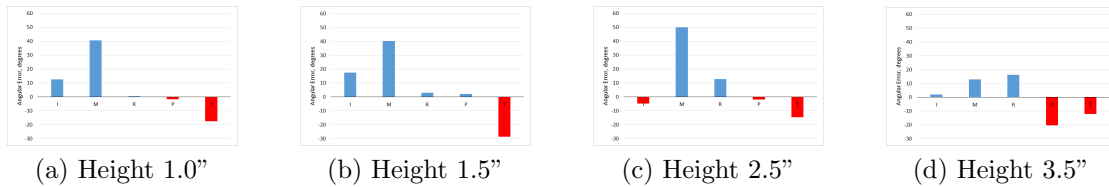


Figure 3.20: Angular error between the actual and ANN predicted angles for grasping sphere S1. The predictions were performed with Sp-15/5000 with 24 data sets. Blue bars indicate *overgrasping* and red bars indicate *undergrasping*.

The “under” and “over” grasping of the fingers based on the predictions of Sp-9/3600 can be seen in Figure 3.21. For Test 1, the index, middle, and ring *overgrasp* the object, while the pinky and thumb *undergrasp*. Test 2, all the fingers, except the thumb, *overgrasp*. In Figure Test 3, the middle and the ring finger *overgrasp*, while the index, pinky, and thumb *undergrasp*. Finally, in Figure Test 4, the index, middle, and ring *overgrasp*, while the pinky and thumb *undergrasp*. Another observation

to make is that in general, the predictions made by Sp-9/3600 yielded lower error compared to Sp-15/5000.

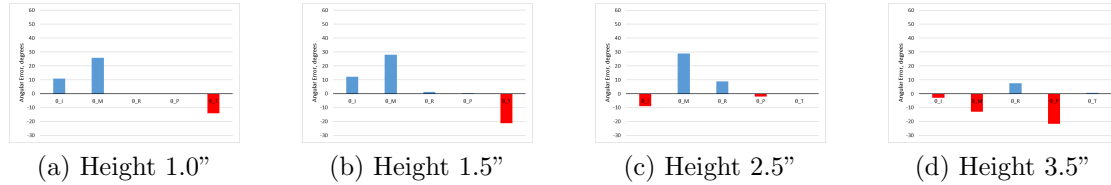


Figure 3.21: Angular error between the actual and ANN predicted angles for grasping spheres. The predictions were performed with Sp-9/3600 trained with 24 data sets. Black bars indicate *overgrasping* and red bars indicate *undergrasping*.

After software validation for grasp predictions for both spheres and cylinders, it was observed that the errors were relatively small, which provided confidence to move forward to hardware validation, which is found in the following subsection.

3.4.2 Hardware Implementation - Grasping Known Objects

To validate the performance of the ANNs, hardware validation will be performed to determine successful or unsuccessful grasps. The motor angles predicted for Test1, Test2, Test3, Test4 are sent to the actuators to grasp cylinders and spheres. Note that the motion of the fingers follow the order of moving first the thumb, then the middle finger, then the index and ring finger at the same time and then finally the pinky. This is true for the fingers required for a grasp. There is also a motion delay artificial. The motion of the fingers run off the same clock, however, the fingers are delayed. For instance, the thumb (θ_T) runs off of time τ , while the middle (θ_M) finger runs off time $\tau-4$. Then the index (θ_I) and ring (θ_R) finger runs off time $\tau-5$, while the pinky (θ_P) runs off time $\tau-6$. All fingers start from their initial position ($\theta = 0^\circ$)

and then end at their final position ($\theta = \theta_f$) following a cubic polynomial profile. Hence, the equations of motion for all the fingers are the following:

$$\theta_T(t) = a_0 + a_1t + a_2t^2 + a_3t^3 \quad (3.12)$$

$$\theta_I(t - 5) = a_0 + a_1(t - 5) + a_2(t - 5)^2 + a_3(t - 5)^3 \quad (3.13)$$

$$\theta_M(t - 4) = a_0 + a_1(t - 4) + a_2(t - 4)^2 + a_3(t - 4)^3 \quad (3.14)$$

$$\theta_R(t - 5) = a_0 + a_1(t - 5) + a_2(t - 5)^2 + a_3(t - 5)^3 \quad (3.15)$$

$$\theta_P(t - 6) = a_0 + a_1(t - 6) + a_2(t - 6)^2 + a_3(t - 6)^3 \quad (3.16)$$

Where the constants of the equations are evaluated based on desired end position, velocity and acceleration of the motors, hence,

$$a_0 = \theta_0, \quad a_1 = 0, \quad a_2 = \frac{3}{t_f^2}(\theta_f - \theta_0), \quad a_3 = -\frac{2}{t_f^3}(\theta_f - \theta_0)$$

The value of t_f for all fingers is set to 5 seconds, so that upon motion, it takes all the individual fingers 5 seconds to reach their final positions. As for the value of θ_f , this is the output of the ANN and is different for each finger. Since each experiment starts at the hand completely open, the value of θ_0 is set to zero.

3.4.2.1 Grasping Cylinders

Figure 3.22 shows the final positions of the fingers based on the output predictions of Cy-9/2600. It is observed that the fingers adequately grasp the object at all heights except at 3.5". On the other hand, at relative height of 3.5", the index does not come into contact with the object. The reason for this is that the object is assumed uniformly cylindrical, from the point of view of the ANN. However, in reality, the object is a pill container, composed of a body and a cap (the body and cap have similar diameters, but the cap has a slightly smaller diameter). Therefore the object is not grasped with the index finger.

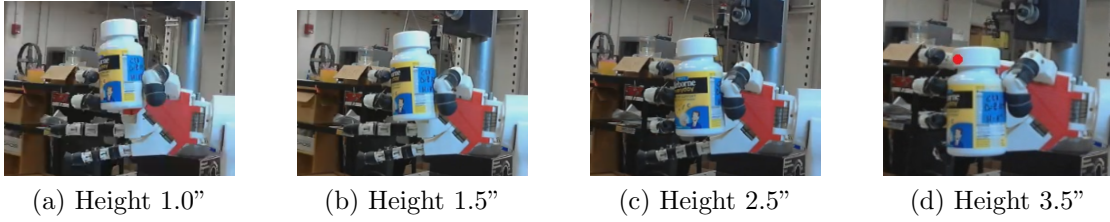


Figure 3.22: Images of the final grasp from the resulting prediction of Cy-9/2600 trained with 76 data sets.

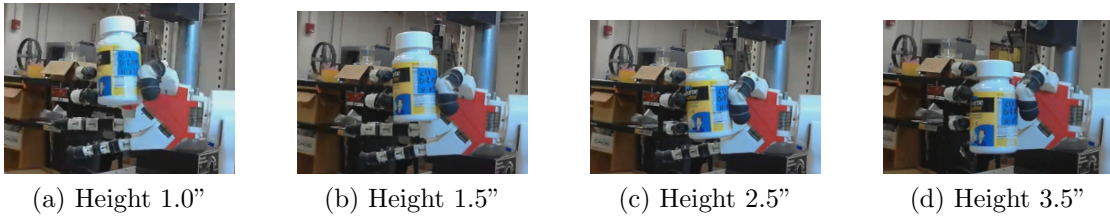


Figure 3.23: Images of the final grasp from the resulting prediction of the Cy-12/4600 trained with 76 data sets.

Using Cy-12/4600, Figure 3.23 shows the resulting grasps. As observed in all the Subfigures, all the fingers come into contact with the object. As opposed to the previous architecture, the object is grasped throughout all the relative heights. The prediction made by the ANN resulted in the *overgrasp* of the index finger, and as a result, contact is made on the object.

For the majority of the tests, in both architectures, the ANN prediction results in *overgrasping* for the majority of the fingers. This puts confidence in the prediction of the ANN, given that the finger positions will ensure the grasp of the object. It is observed, in both Figures 3.18 and 3.19, that the finger that always results in the highest *overgrasp*, in general, is the middle finger, θ_M . This is important to note, since for the majority of the grasps, the order by which the fingers close on to the object is first the thumb (θ_T) and then the middle finger (θ_M), see Figure 3.6. If the middle finger *overgrasps* the object, then the apprehension of the object is at least

guaranteed from a two contact-point⁵ perspective. Then, the rest of the appropriate fingers will grasp the object, however, an initial first grasp is made with the thumb and middle fingers.

Up till now, the tests performed have been on objects for which we know what the grasp patterns should be at. For Test1, Test2, Test3, and Test4, the final finger positions were already investigated, that is how the error was calculated, where the difference was taken between the actual (measured) and predicted motor angles. The next stage of testing is to introduce new objects for which the grasp pattern is unknown.

3.4.2.2 Grasping Spheres

After testing the ANN prediction in software, the next step, as it was for the cylinders, is to test out the algorithm in hardware. As shown in Figure 3.24, the final grasping positions for the fingers are noted for Sp-9/3600. One of the observations made of how the artificial hand grasps objects, made with ANN predictions, is the inability to grasp the object at height of 1.5". The thumb is *undergrasping*, even if the index and the middle finger are *overgrasping*. Moreover, the *undergrasping* angular error is quantified as a little more than 20°. Due to this, the sphere was not able to be grasped.

The same test was performed for ANN-15/500 on the object. The final grasps can be observed in Figure 3.25. The grasps between both Figures look similar, and the same fingers were incorporated to grasp the sphere at a specific height. The thumb actually performs worse with the ANN architecture that resulted in the lowest MSE

⁵In general, a contact or grasp using two finger, is termed a *precise grasp*. This is given since two fingers will not apply a relatively large amount of grasp force and delicate objects which require precise handling, are performed with two fingers.

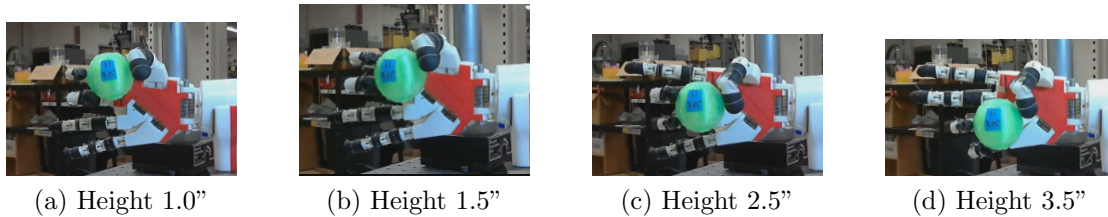


Figure 3.24: Images of the final position of the fingers in grasping based on the Sp-9/3600. The object grasped is a Sphere S1.

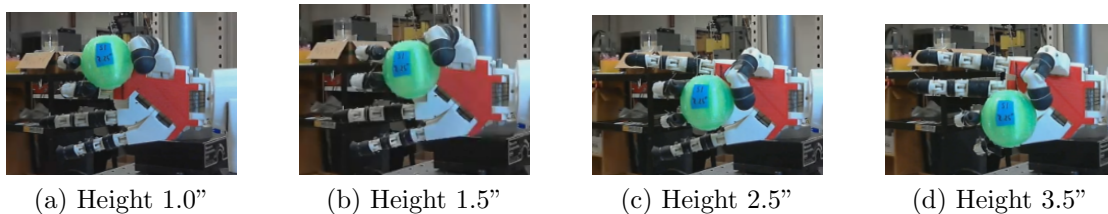


Figure 3.25: Images of the final position of the fingers in grasping based on the Sp-15/5000 predictions. The object grasped is a Sphere S1.

and APD (Sp-15/5000). It can be observed in Figure 3.20 that the angular error is about 30° , more than the angular error observed in Figure 3.21 with Sp-9/3600.

Both of the different architectures (Sp-15/5000 and Sp-10/3600) were able to successfully grasp the object 75% of the time. On the occurrence that the object was not grasped successfully, it was due to the thumb *undergrasping* its required position. Once again, at this stage of research, only position control is considered as opposed to force control. In this environment, there is no awareness of contact or the degree of contact if contact is made. However, it is envisioned that when force control is implemented, the issue of not making contact will be eliminated with a force control algorithm for grasping.

3.4.3 Hardware Implementation - Grasping *Unknown* Objects

There were 3 objects, two cylinders and one sphere, that were not used in the training process. These objects are considered *unknown*, since they were not incorporated in the optimization of the ANN, hence, their “actual” grasping pattern is not known. The following subsections demonstrated the hardware implementation of grasping *unknown* objects.

3.4.3.1 Cylinders

To test the robustness of the ANN for predicting grasp patterns, two cylinders that were not used for the training of the ANN were utilized to evaluate prediction. Both of the objects that were used for unknown cylinders for grasping are found in Figure 3.26. Their dimensions are tabulated in Table 3.1. Note, based on G_r and L_r , there is an object that was used in training that was similar to UC1 which is object C20. Similarly for object UC2, object C7 has similar G_r and L_r ratios⁶.

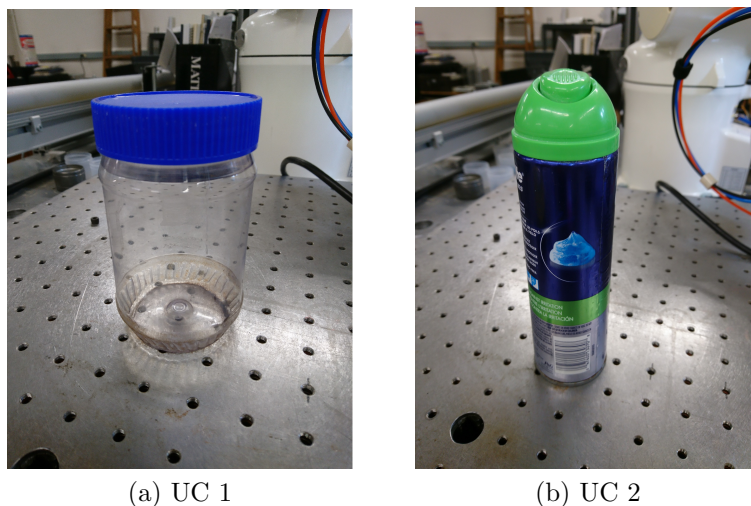


Figure 3.26: Objects used to evaluate the prediction performance of the ANN.

⁶See Figure 1 found in the Appendix to note the similarities

Table 3.1: Table of the geometric properties of the objects used to test the ANN prediction abilities.

Object	Diameter, D_o (in)	Length, L_o (in)	G_r (in/in)	L_r (in/in)
UC 1	2.90	4.91	0.78	1.31
UC 2	2.08	7.00	0.56	1.87

Table 3.2: Final finger positions for grasping object UC 1. The units for the fingers are in degrees. The first 4 rows are results of Cy-9/2600, while the last 4 rows are results from Cy-12/4600.

ANN Architecture	Height (in)	θ_T	θ_I	θ_M	θ_R	θ_P
Cy-9/2600	1.0	92	45	54	0	0
	1.5	98	51	56	0	0
	2.5	106	54	61	55	0
	3.5	113	61	62	52	52
Cy-12/4600	1.0	89	47	51	0	0
	1.5	96	54	57	0	0
	2.5	107	56	58	53	0
	3.5	110	63	65	49	52

Table 3.2 shows the tabulated results of the grasp patterns of the mechanical hand at all the standard heights (1.0, 1.5, 2.5, 3.5) for both proposed architectures Cy-9/2600 and Cy-12/4600. It is noticed that the angles between both architectures. Moreover, the images of the final position of the hands grasping object UC 1 for Cy-9/2600 as well as Cy-12/4600 can be observed in Figure 5 (Appendix) and 3.27 respectively. It is important to note that all the objects were able to be grasped at all heights for both architectures, and all the finger tips made contact with the surface of the cylinder.

In Figure 6, in the Appendix, the grasping of the unknown object, UC 2, can be observed at all the standard heights (1.0, 1.5, 2.5, 3.5). Throughout all heights, the artificial hand was successful at grasping with Cy-9/2600. The final angular positions

Table 3.3: Final finger positions for grasping object UC 2. The units for the fingers are in degrees. The first 4 rows are results of Cy-9/2600, while the last 4 rows are results from Cy-12/4600.

ANN Architecture	Height (in)	θ_T	θ_I	θ_M	θ_R	θ_P
Cy-9/2600	1.0	100	66	76	0	0
	1.5	105	70	78	0	0
	2.5	112	65	78	65	0
	3.5	121	69	74	64	57
Cy-12/4600	1.0	98	70	77	0	0
	1.5	105	70	81	0	0
	2.5	114	67	77	63	0
	3.5	119	65	73	64	58

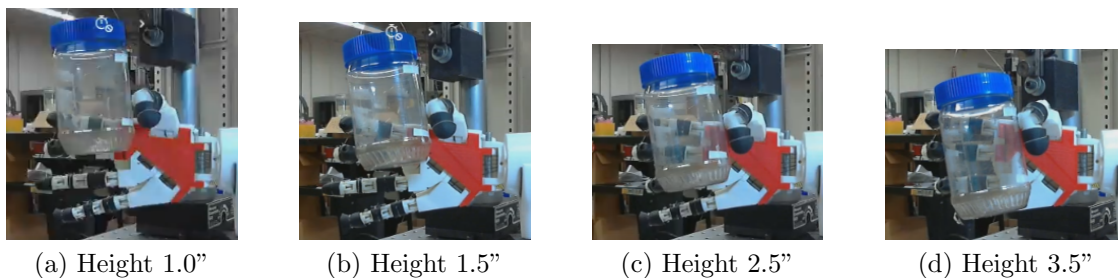


Figure 3.27: Final grasp positions of the fingers for object UC 1. The ANN was trained with Cy-12/4600.

of the motors for each individual finger can be observed in the first four rows of Table 3.3. Similarly, in Figure 3.28, the final grasps of the fingers on cylinder UC 2 are observed to successfully grasp the object, this time with Cy-12/2600. The last four rows of Table 3.3 indicate the final angles for grasping with this architecture. For all heights that the object UC 2 was placed, the final angles for both architectures are similar, and result in a successful grasp.

3.4.3.2 Spheres

To test the predictive performance of the ANN in hardware, a sphere that was provided to the ANN so that it could predict the grasp pattern. The sphere has a

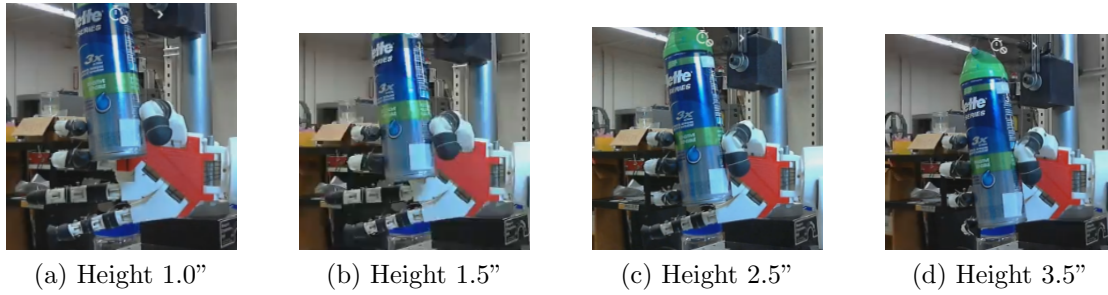


Figure 3.28: Final grasp positions of the fingers for object UC2 with Cy-12/4600.

diameter of 3" and can be seen in Figure 3.29. The final finger angles used for grasping sphere US1 are tabulated in Table 3.4. The images of the final finger positions for grasping the object at different heights are presented in Figures 7 (Appendix) and 3.30 for Sp-9/3600 and Sp-15/5000 respectively.

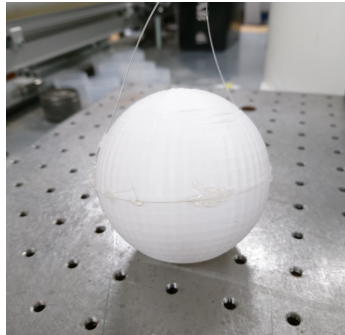


Figure 3.29: Object used as an unknown sphere to grasp.

Using object US 1, testing was performed with ANN-10/3600 and then the motor angles for the fingers were sent to the on-board motors. It is observed from Figure 7 that the object, US 1, is grasped at all heights, except at 2.5" the index finger does not make contact with the sphere. At height 2.5", the index finger does not make contact with the object. However, it is important to note that the object is firmly maintained with the thumb, middle and ring finger grasp. Afterwards, the object is grasped with ANN-15/5000. It is assumed that the grasp predictions will

Table 3.4: Final finger positions for grasping object US 1. The units for the fingers are in degrees. The first 4 rows are results of Sp-9/3600, while the last 4 rows are results from Sp-15/5000.

ANN Architecture	Height (in)	θ_T	θ_I	θ_M	θ_R	θ_P
Sp-9/3600	1.0	94	56	60	0	0
	1.5	94	56	60	0	0
	2.5	104	42	63	57	0
	3.5	121	0	64	59	50
Sp-15/5000	1.0	84	54	67	0	0
	1.5	97	55	60	0	0
	2.5	105	42	60	58	0
	3.5	119	0	64	60	50

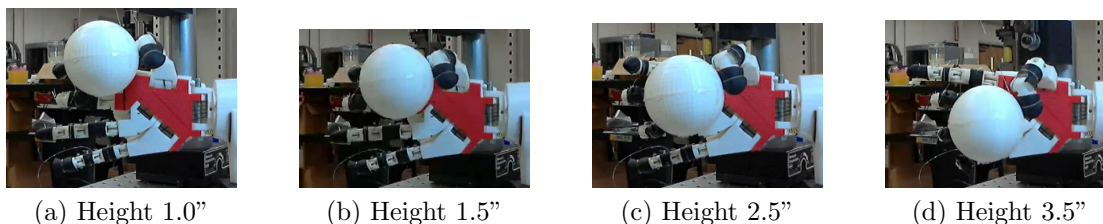


Figure 3.30: Final grasp positions of the fingers for object US1 with Sp-15/5000.

improve, however, it is observed in Table 3.4, the angles only slightly differ between both architectures. Moreover, the same issue that was present at height 2.5" occurred again with the larger ANN architecture. In both cases, the index finger does not move enough to make contact with the object, hence does not participate in the holistic grasp of the object, nevertheless, the object is still secure in the grasp of the hand with the other fingers (thumb, middle and ring).

This is a issue at the moment, because it is assumed that the object will be firmly held by the had with all available fingers. Nevertheless, at this point in research, only position is the main component that will be considered for control. It is envisioned that when the two stage controller is implemented, the force control will ensure a firm contact is made with all the fingers, thus ameliorating this current issue.

3.4.4 Conclusion - Hardware Implementation

In conclusion from both scenarios, grasping spheres and cylinders, it can be seen that algorithm performed adequately considering only position control. One of the recommendations would be to grasp more objects, particularly spheres. There were only 7 spheres used to train the ANN, compared to the cylinders, in which there were 21 cylinders. This could be the reason why the ANN predicted an *undergrasp* twice for the unknown object US1. Nevertheless, the predictive system based on the training structure for spheres and cylinders are promising for future control architecture and hardware grasp implementation. Below is a summary of the ANN architectures that were established as benchmarks for testing.

- Cylinders
 - Cy-9/2600
 - Cy-12/4600 - Architecture yielding lowest MSE
- Spheres
 - Sp-9/3600
 - Sp-15/5000 - Architecture yielding lowest MSE

3.5 Non Discriminatory Artificial Neural Network

So far, an ANN has been isolated to predict grasp patterns for specific objects; spheres and cylinders. The inputs to both ANNs were the object dimensions of length and diameter⁷ as well as the height at which the object will be grasped relative to the location of the hand. This resulted in two customized ANNs that would be able to provide the user, specifically the system, a prediction of the necessary motor angles for grasping either the sphere or cylinder. In this section, we will investigate the

⁷For spheres, the diameter and the length are equal.

development of a ANN that will take as input, not length and diameter, but G_r and L_r (Equations 3.1 and 3.2). The reason for this is two fold:

- In one sense, this will avoid the user having to strictly use the terms of sphere and cylinder, but would allow shapes that are not perfectly cylindrical or spherical.
- Second, it allows the training set to expand to combine both what have already been categorized as spheres and cylinders into one group. This will result in a larger size for training sets, which in theory will make more accurate predictions of grasp patterns.

In the context of this research with objects of distinct geometric features, cylinders and spheres, a 5-step process for identifying grasp patterns is developed as shown in Figure 3.31. In order to predict a grasp pattern, it is required to first discriminate or identify/define the object, as a cylinder or sphere, and its geometric characteristics. Then, two ANNs would need to be trained with distinct object specific training data. Once both ANNs are trained, the testing process would required that the category of whatever object to be grasped need to be identified (sphere or cylinder). Finally, the shape specific ANN would be used to predict the grasp pattern. On the contrary, as presented in the bottom of Figure 3.31, the entire process can be shortened if a non-discriminatory ANN was trained to predict the grasp pattern of the shape without *a priori* specific knowledge on the object category (sphere or cylinder). This approach would reduce the grasp pattern prediction procedure from five to two steps. As such, training occurs with the entire object population and then the resulting trained ANN is used to predict the grasp pattern of any object. This is a simpler approach of attaining the final outcome of a grasp pattern to be executed in hardware.

Therefore, all the dimensions of the objects (diameter and length) were normalized to the dimensions of the artificial hand, hence L_r and G_r . The inputs to the

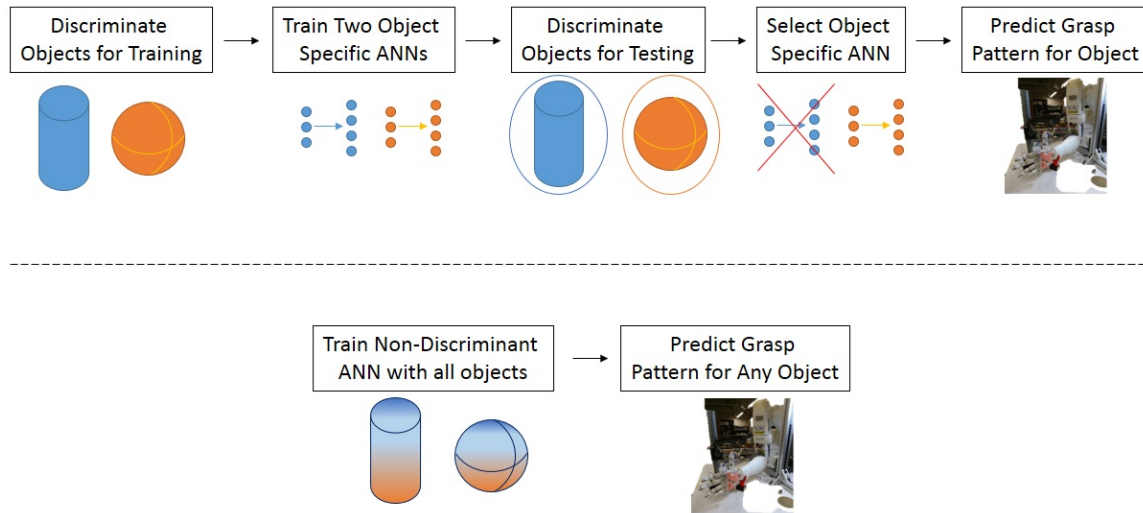


Figure 3.31: Schematic of grasp planning with discriminating objects, top, and non-discriminating objects, bottom.

ANN would now be L_r , G_r , and relative height. The next task was to determine the architecture of the ANN for training and testing purposes.

3.5.1 Sphere and Cylinder Predictions

There are a total of 6 spheres and 19 cylinders, so there are a total of 25 objects that were used for training. Each object had 4 heights at which it was grasped, thus there are a total of 100 sets. The last 4 will be removed from the set and will be used for testing⁸, hence there will be 100 training sets and 4 test sets.

In Figure 3.32, the APD and MSE can be seen with respect to the different ANN architectures. The minimum Absolute Percent Difference begins around the same range of about 40-45%. After about 600 iterations, the different architecture lines become more sparse and the architectures that yield the lowest error become

⁸Which are grasping object C22 grasped at height 3.5” and 2.5” and object S6 grasped at height 1” and 1.5”

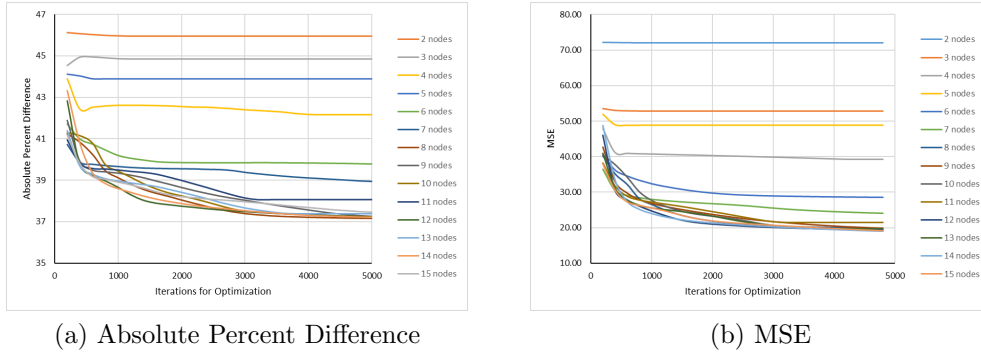


Figure 3.32: Performance of the ANN based on the APD and MSE. The ANN was trained with 100 training sets.

more pronounced. The architecture that yields the smallest APD is $N_I = 5000$ and $N_{HN} = 8$. This is also observed in MSE and in AD (Figure 8 found in the Appendix).

Another observation made from Figure 3.32 and 8 is that starting from 3000 iterations, the majority of the architectures yield similar errors. This is advantageous when considering computation time, given that the architecture that yield the lowest error is also the largest architecture will require the most computations. Due to this, it would be beneficial to identify a smaller architecture, yet would yield a relatively small error. Based on the Figures, it was identified that CySp-10/3600 would be smaller than CySp-8/5000 and yet still yield a relatively small error. Therefore, two architectures will be used to investigate grasp pattern predictions, CySp-10/3000 and CySp-8/5000.

3.5.2 Software Implementation

Now that the ANN architectures have been identified, the next step is to test the ANN. The resulting difference will determine the accuracy of the prediction relative to the actual/measured grasp pattern.

3.5.2.1 Spheres and Cylinders

Using objects C22 and S6 at all heights, the ANN architectures are tested by comparing the actual and predicted grasp pattern. Figure 3.33 presents the angular errors for the predictions made with CySp-10/3000 are presented. Note that for all four test, the only finger to consistently *overgrasp* is the middle finger. This could be a positive characteristic, as mentioned before, *overgrasping* could result in a guarantee that the hand will make contact with the object. The most accurate prediction, in terms of fingers throughout all test, is the pinky, which is observed by the its low magnitude. Moreover, comparing Test 1 and 2 to Test 3 and 4, the “over” and “under” grasping is more pronounced. This indicates, at least for these four test, that grasping cylinders will result in least accurate predictions compared to grasping spheres.

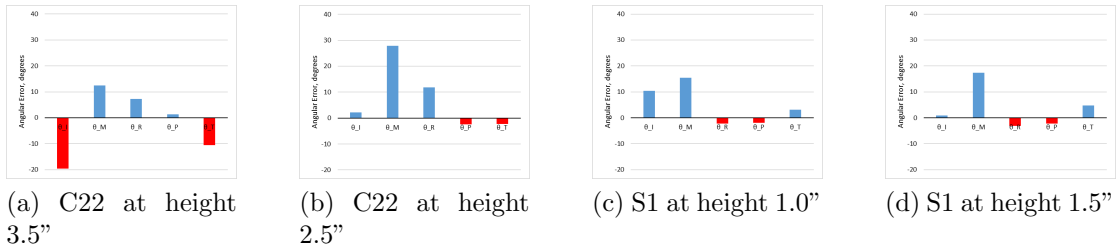


Figure 3.33: Angular error between the actual and ANN predicted angles for grasping sphere. The predictions were performed with CySp-10/3000 trained with 100 data sets. Blue bars indicate *overgrasping* and red bars indicate *undergrasping*.

Figure 3.34 presents the angular error between the prediction and the actual grasp patterns with CySp-8/5000. In general, the accuracy has improved slightly, which is observed by noting the magnitudes of the errors between Figure 3.33 and 3.34. Just as in the case for CySp-10/3000, the most accurate finger is the pinky and

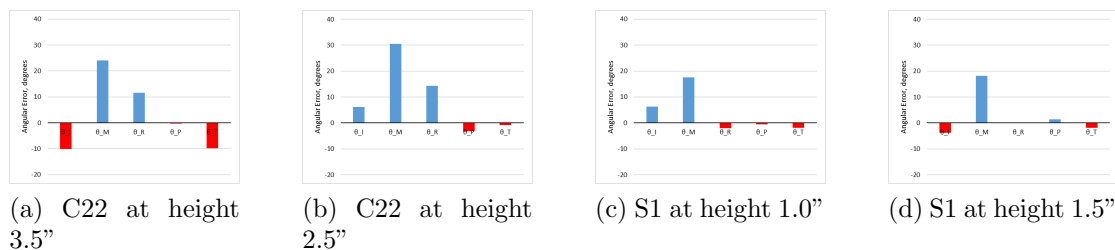


Figure 3.34: Angular error between the actual and ANN predicted angles for grasping. The predictions were performed with CySp-8/5000 trained with 100 data sets. Blue bars indicate *overgrasping* and red bars indicate *undergrasping*.

the least accurate is the middle finger. From both ANN architectures, two items can be concluded:

- Predicting grasps for cylinders is less accurate than for spheres.
- The middle finger and then pinky are the least and most accurate of the fingers.

3.6 ANN Architecture Comparison

There have been 3 independent ANN architectures investigated so far; one for spheres (Sp), cylinders (Cy) and a combined (CySp). The predictive performances will be compared between the ANN trained for spheres and combined objects (Sp vs. CySp) and then the ANN trained for cylinders and combined objects (Cy vs. CySp) will be compared. In the combined test, a cylinder and sphere were utilized, particularly, cylinder C22 and sphere S1. Therefore, the predictions made for C22 at height 2.5" and 3.5" will be compared (Cy vs. CySp) and the predictions for S1 at height 1" and 1.5" will be compared (Sp vs. CySp).

3.6.1 Cy and CySp: Comparison of Predictive Performance

In Figure 3.35, the angular errors as a result of the predictions can be observed for grasping C22 between both categories of ANN. At height 2.5", considering

the index finger, CySp-8/5000 resulted in the smallest error and Cy-9/2600 has the highest error, and all architectures result in *undergrasping* of the index finger. For the middle finder, Cy-13/4600 results in the lowest error while CySp-8/5000 had the highest error and all fingers are *overgrasping*. For the ring finger is Cy-13/4600 and the highest is CySp-8/5000, note that all the fingers result in *overgrasping*. For the pinky, Cy-13/4600 has the lowest error and Cy-9/2600 has the highest error. Note that for the prediction of the pink results in the lowest error for all the ANN architectures. For the thumb, lowest error is a result of Cy-13/4600 and the highest is from CySp-10/3000.

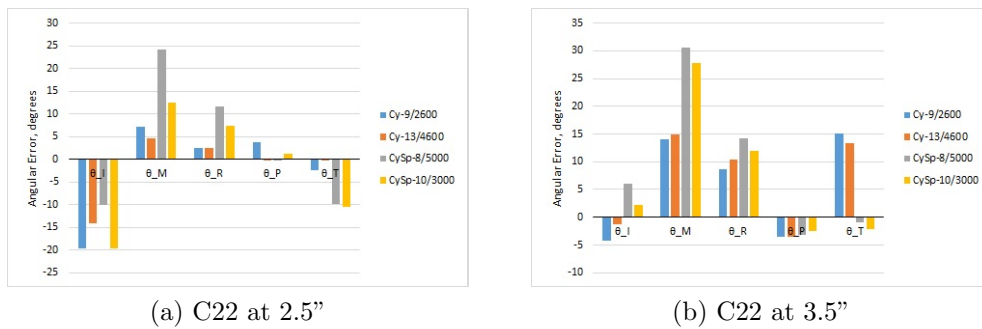


Figure 3.35: Comparison of the angular errors between the actual grasp and the predictions of the various ANN architectures. This is for grasping object C22 at height 2.5" 3.5".

At height 3.5", the lowest error for the index finger is provided by the architecture Cy-13/4600 and the highest is provided by CySp-8/5000. For the middle finger, the lowest error is given by Cy-9/2600 and the highest error is given by CySp-8/5000, and all fingers result in the finger *overgrasping*. The ring finger has the lowest error is given by Cy-9/2600 and the highest error is given by CySp-10/3000. The lowest error for the pinky is given by CySp-10/3000 and the highest is given by Cy-13/4600.

And for the thumb, CySp-8/5000 results in the lowest error and the highest error is from Cy-9/2600.

In Table 3.5, the average absolute errors for each architecture are presented. It is observed that for height 2.5", Cy-13/4600 results in the smallest error and the largest absolute error by CySp-8/5000. Then, for the height of 3.5", the lowest error is from Cy-13/4600 and the largest error is from CySp-8/5000. It can be concluded from Table 3.5, the architecture that yields the lowest error is Cy-13/4600 and the highest error by CySp-8/5000. Moreover, it can be concluded that the ANNs specifically trained for cylinders resulted in more accurate predictions compared to the combined ANN.

Table 3.5: Summary of the average absolute error for all the architectures across all the fingers.

Height	Cy-9/2600	Cy-13/4600	CySp-8/5000	CySp-10/3000
2.5"	7.086	4.252	11.153	10.226
3.5"	9.125	8.681	10.996	9.319

3.6.2 Sp and CySp: Comparison of Predictive Performance

It is also interesting to analyze the angular error in the predictions by different architectures, now predicting for object S1, presented in Figure 3.36. For the predictions made for the index finger, the lowest error is provided by CySp-8/5000, and the highest by Sp-15/5000, all of which resulted in *overgrasping* of about 10°. The largest errors occurred when predicting the middle finger position. The smallest prediction error is provided by CySp-10/3000 and the largest by Sp-15/5000. For the ring finger, the largest error was made by CySp-10/3000 and the smallest error is by Sp-9/3600. The pinky finger angle prediction resulted in the smallest error across all

architectures, smallest by Sp-9/3600 and the largest by Sp-15/5000. Then, for the thumb, the largest error was by Sp-15/5000 and the smallest by CySp-8/5000.

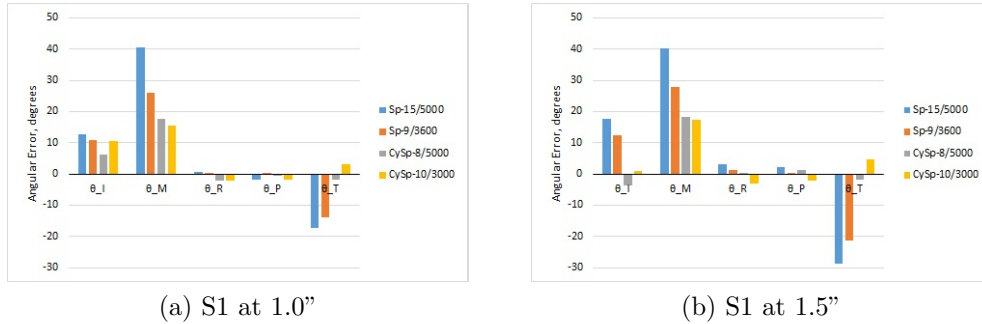


Figure 3.36: Comparison of the angular error between the actual grasp and the predictions of the various ANN architectures. This is grasping object S1 at height 1.0'' and 1.5''.

Table 3.6: Summary of the average absolute error for all the architectures across all the fingers.

Height	Sp-9/3600	Sp-15/5000	CySp-8/5000	CySp-10/3000
1.0''	10.162	14.543	5.663	6.635
1.5''	12.630	18.311	5.031	5.671

In Table 3.6, the absolute error was averaged for all the fingers and tabulated for each architecture at the different heights. It is observed that the architecture that resulted in the lowest error was CySp-8/5000 where Sp-15/5000 yielded the highest error. It can be concluded that the combined ANN, CySp-8/5000, resulted in more accurate predictions of grasping objects and the least accurate was the ANN trained with only spheres, Sp-15/5000. Moreover, the prediction of the pinky resulted in the most accurate and the prediction of the middle finger resulted in the least accurate, for all architectures.

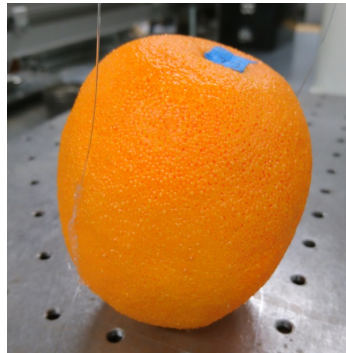
From the results of comparing the combined ANN with both the sphere and cylinder ANN, it was observed that the combined ANN performed more accurately when compared with the sphere ANN. The reason for this might be attributed to the fact that the ANN for combined objects has 27 objects compared to 7 spheres and as a result, the more training sets, the more accurate the expected prediction. However, the cylinder ANN performed more accurately than the combined ANN. The possible reason behind this is that the spheres introduced to the ANN, created more variance to the data set and resulted in less accurate predictions. Nevertheless, all three ANN architectures result in a firm grasp of the objects.

3.6.3 Grasping Near-Cylinders and Near-Spheres

Another aspect of research is grasping objects that are not strictly spheres or cylinders. The research question is how do the three architectures compare in predicting the grasp of the object. In Figure 3.37, the two objects used for unidentified grasping are a decorative apple, UO1, and a decorative orange, UO2, which are neither spherical nor cylindrical. The outer dimensions of these objects are found in Table 3.7.



(a) UO1 - decorative apple



(b) UO2 - decorative orange

Figure 3.37: Two objects that are neither completely spherical or cylindrical.

Table 3.7: Dimensions of objects UO1 and UO2.

Object	Horizontal Width	Vertical Length
UO1	2.972	3.001
UO2	2.933	3.382

After grasping the both objects with all the selected ANNs, Table 3.8 has all the resulting motor angle predictions tabulated for grasping UO1. It is observed that there were instances where the object was grasped but some fingers did make contact with the object. For grasp UO1 at height 1.0” from Sp-9/3600, the grasp resulted weak and when tapped, the object slipped out. For grasp UO1 at height 1.0” from Cy-9/2600, the object slipped out of the hand and was not grasped. For grasp at height 2.5” with CySp-8/5000, Cy-13/4600, Sp-9/3600 and Sp-15/5000 result in a good grasp, however, the index is *undergrasped* and does not make contact with the object. For grasp UO1 at height 3.5” for CySp-10/3000, the hand is able to grasp the object, however, the middle finger does not make contact with the object. For grasp UO1 at height 3.5” for CySp-8/5000, the hand is able to grasp the object, however, the middle and pinky finger do not make contact the object. For OU1 grasped at height 3.5” for Cy-9/2600, the object is grasped, however, the index and pinky finger do not make contact with the object. For OU1 grasped at height 3.5” with Cy-13/4600, the object is grasped, however, the index and the pinky do not make contact with the object. For both architectures, Cy-9/2600 and Cy-13/4600 predict that the index finger should move and contact the object, nevertheless, the index finger does not make contact. For grasping with Sp-9/3600 and Sp-15/5000, the object is grasped but pinky does not make contact with the object.

Moreover, Table 3.9 tabulates the predictions by the different ANN architectures for grasping UO2. It is important to note that some of the fingers did not make

Table 3.8: ANN predictions by various architectures on unidentified object UO1. The values in bold indicate when the finger has moved however, no contact was made with the object.

Height	Architecture	Thumb	Index	Middle	Ring	Pinky
1	CySp-10/3000	84	53	63	0	0
	CySp-8/5000	83	57	64	0	0
	Cy-9/2600	86	60	65	0	0
	Cy-13/4600	79	58	70	0	0
	Sp-9/3600	90	49	63	0	0
	Sp-15/5000	83	54	66	0	0
1.5	CySp-10/3000	93	60	67	0	0
	CySp-8/5000	93	55	63	0	0
	Cy-9/2600	93	61	67	0	0
	Cy-13/4600	91	57	74	0	0
	Sp-9/3600	93	56	60	0	0
	Sp-15/5000	97	55	60	0	0
2.5	CySp-10/3000	109	34	64	58	0
	CySp-8/5000	109	31	58	57	0
	Cy-9/2600	104	48	69	63	0
	Cy-13/4600	110	42	69	63	0
	Sp-9/3600	103	44	62	57	0
	Sp-15/5000	104	44	59	58	0
3.5	CySp-10/3000	123	0	38	59	60
	CySp-8/5000	121	0	39	58	59
	Cy-9/2600	112	42	65	61	60
	Cy-13/4600	119	40	71	58	59
	Sp-9/3600	120	0	66	60	50
	Sp-15/5000	119	0	66	60	50

contact with the object, these are indicated with bold numbers. When grasping UO2, at height 2.5” with CySp-10/3000, the object is grasped, however, the index finger does not make contact. Grasping OU2 at height 2.5” with CySp-8/5000, the object is grasped however, the index finger does not make contact. For grasping UO2 at height 2.5” with Cy-9/2600, the object is grasped however, the index and the ring finger barely come into contact. Grasping at height 3.5” with CySp-8/5000, UO2 is grasped however, the index finger moves a little (26°) and the middle finger

Table 3.9: ANN predictions by various architectures on unidentified object UO2. The values in bold indicate when the finger has moved however, no contact was made with the object.

Height	Architecture	Thumb	Index	Middle	Ring	Pinky
1	CySp-10/3000	84	56	62	0	0
	CySp-8/5000	87	54	61	0	0
	Cy-9/2600	88	56	62	0	0
	Cy-13/4600	82	55	65	0	0
	Sp-9/3600	87	50	62	0	0
	Sp-15/5000	79	58	67	0	0
1.5	CySp-10/3000	93	61	66	0	0
	CySp-8/5000	96	57	61	0	0
	Cy-9/2600	95	59	65	0	0
	Cy-13/4600	93	57	71	0	0
	Sp-9/3600	91	55	59	0	0
	Sp-15/5000	93	58	64	0	0
2.5	CySp-10/3000	107	41	67	57	0
	CySp-8/5000	111	39	60	55	0
	Cy-9/2600	105	51	67	61	0
	Cy-13/4600	110	48	68	61	0
	Sp-9/3600	100	51	50	58	0
	Sp-15/5000	99	54	55	59	0
3.5	CySp-10/3000	120	23	47	58	57
	CySp-8/5000	122	26	47	56	56
	Cy-9/2600	113	48	64	58	57
	Cy-13/4600	118	48	71	55	56
	Sp-9/3600	119	0	60	66	60
	Sp-15/5000	118	0	73	64	52

makes contact. For grasping at height 3.5” with CySp-10/3000, UO2 is grasped, however, the index finger moves slightly but does not make contact. Grasping object at height 3.5” with Sp-9/3600, the object is held however the pinky moves but does not make contact with the object. At height 3.5” with Sp-15/5000, the object is grasped however, the pinky does not make contact with the object. At height 3.5” with Cy-9/2600, the object is held by the index moves slightly and does not come close to contacting the object, and the pinky moves up to the object but does not

make contact. At height 3.5" with Cy13/4600, UO2 is grasped, however, the index finger does not come into contact with the object and the pinky moves toward the object but does not make contact with it.

From both Tables 3.8 and 3.9, it can be concluded that once the object is held at heights 2.5" and 3.5", there are instances where some of the fingers do not come into contact with the object. Moreover, the finger that does not contact with the object at height 3.5" is the pinky. The pinky moves toward but does not make contact with the object. This is observed for both case of UO1 and UO2.

Another observation is that the ANN trained with only cylinders (Cy), when grasping at height 3.5", predicts that the index should move, slightly, but does not result in a contact. For instance, at height 3.5", in Table 3.8, the combined (CySp) and the sphere (Sp) ANNs predict zero (0) motion for the index, nevertheless, the cylinder (Cy) ANN predicts small motion, (42° and 40°). Similar predictions can be observed in Table 3.9 at height 3.5", where the index finger is predicted to move, however, it does not come into contact with the object. At 3.5", the combined (CySp) and the cylinder (Cy) ANNs predict that the index finger should move, but no contact is made. The combined (CySp) ANN has a smaller prediction of the index finger motion compared to the cylinder (Cy)⁹.

Both observations¹⁰ result in the lack of contact from the active fingers, nevertheless, the objects are grasped successfully. Another observation is that the thumb makes contact with objects in an unorthodox contact surface, thumb nail. Even though there are successful grasps considering the thumb awkward position, it is of

⁹The cylinder ANNs predicted about 48° for the index to move and the combined ANNs predicted about 25° .

¹⁰1) It is not very contact for the pinky to make common with the object at 3.5", and 2) the cylinder ANNs do not predict the position of the index finger, where instead the finger should not move at all.

interest to optimize thumb positioning based on the shape of the object as well as its relative height to the hand to improve grasping. It is also observed that the fingers have a lack of contact after height 3.5". Issues of *undergrasping* or no contact could be possibly be resolved with additional sensor that could monitor contact.

CHAPTER 4

Grasping Analysis and Verification with Improved Hand H2

A second generation artificial hand with 8 degrees of freedom was utilized to investigate its grasping ability as well as the ability for the ANN to predict the position of the joints or their state for secure and successful grasping. The new thumb incorporated into the second generation hand for grasp investigation is presented in Figure 4.1 and among its different features found in the previous hand is a 4 degree of freedom thumb. Therefore, in total, the user has control of 4 fingers (θ_P , θ_R , θ_M , and θ_I) and 4 joints in the thumb, namely θ_{T1} , θ_{T2} , θ_{T3} , and θ_{T4} .

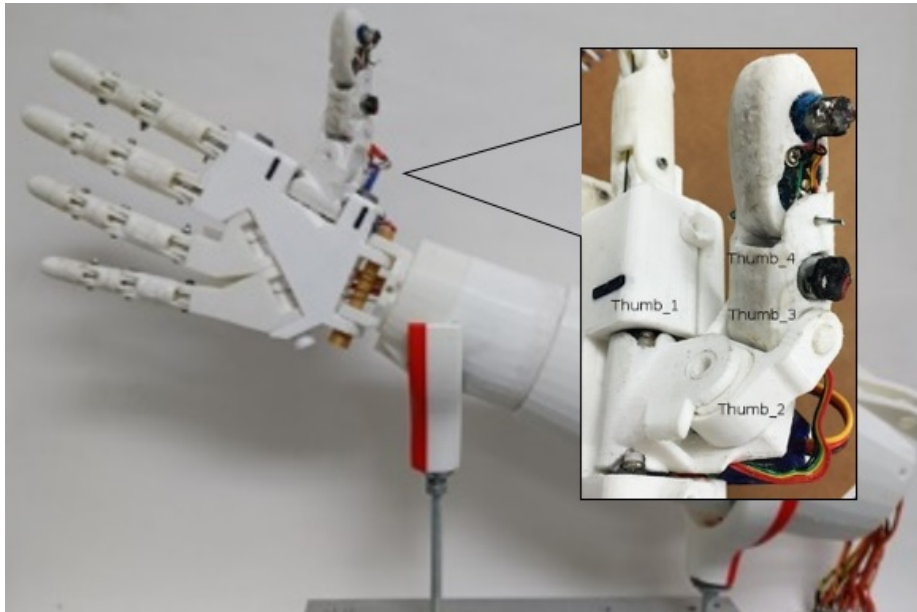


Figure 4.1: Thumb on the second generation hand with individually control joints.

The first joint of the thumb is θ_{T1} , which provides the thumb with flexion and extension movement. The next joint is θ_{T2} , which provides abduction and adduction. The next two joints, θ_{T3} and θ_{T4} provide additional flexion to the thumb. The motion of flexion/extension and abduction/adduction are presented in Figure 4.2 with a natural palm/thumb. The addition of these joints in the thumb provided the user with the advantage of having the ability to control the joints independently and provide fine tuned movement, both features which were not available for the first generation hand. However, there are some disadvantages to the thumb, such as less torque output due to smaller motors and more complexity in motion. These advantages and disadvantages will be observed when grasping objects as well as when predicting the grasp pattern by the ANN.

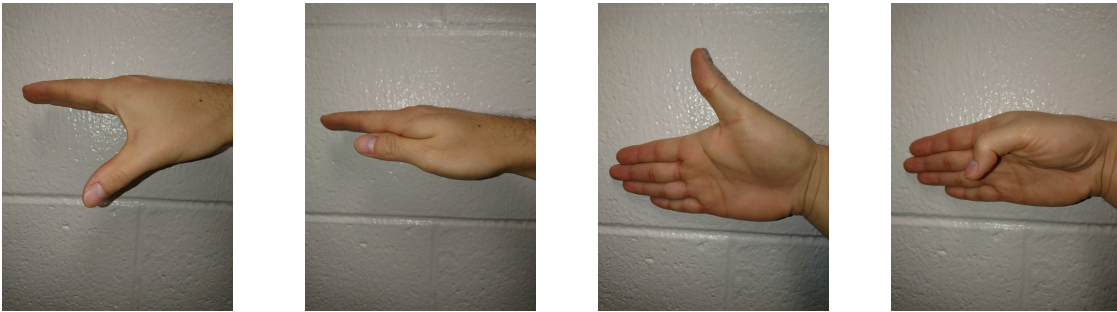


Figure 4.2: From left to right, abduction, adduction, extension, flexion.

4.1 ANN Training

As presented in Chapter 3, the hand was utilized to collect data for training purposes, where different objects were presented to it, grasps were performed and the final angles of all the joint positions were recorded. The information of the object properties such as length, diameter and height, were used as inputs to the ANN, and

the outputs were the final joint angles for grasping the object. Previously, the outputs of the ANN would predict 5 joint angles, now the ANN will have to predict 8 joint angles.

4.1.1 Cylinder Prediction

The cylinder ANN was trained with 76 data sets. Figure 4.3 presents the APD between the ANN predicted angles and the actual joint angles. Along with the APD, the MSE of different ANN architectures is presented.

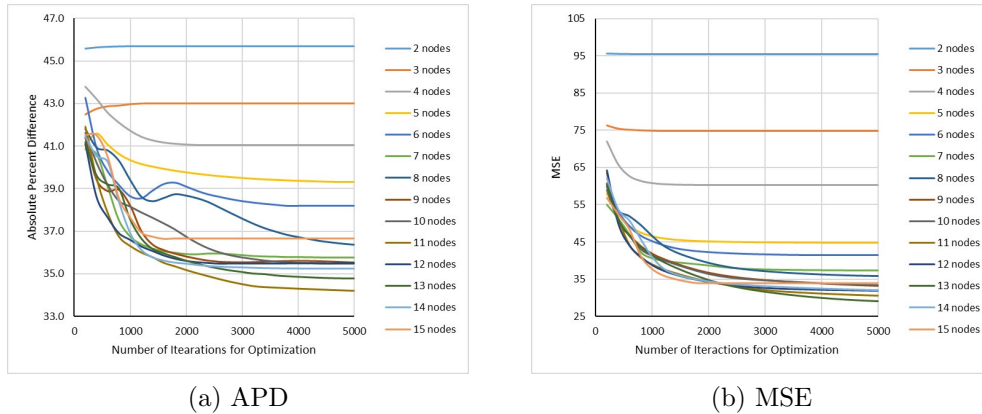


Figure 4.3: The APD and MSE of the resulting ANN architectures when the number of iterations for optimization is varied as functions of number of nodes in the hidden layer. This training was performed with 76 data sets.

At the beginning of the APD, the majority of lines of constant node number begin around the range of 40-44%. Afterwards, at about 100 iterations, the lines become more sparse and distinguishable. At about 4000 iterations, there are two lines that provided low APD, 13 and 11 hidden nodes, the latter yielding the lowest number. Similarly, the same pattern of line separation can be observed, however, at 4000 iterations, 13 nodes yields the lowest MSE. In Figure 9, in the Appendix, the average absolute difference between the predicted and actual angles is presented.

Similar to the results from the MSE, the number of hidden nodes that yields the lowest difference is 13, closely followed by 11 nodes. Therefore, based on all 3 performance metrics, between the APD, MSE, and AD, the architecture to use for testing will be $N_{HN} = 13$ and $N_I = 4600$ (Cy-13/4600). For comparison, another ANN architecture used for testing is $N_{HN} = 11$ and $N_I = 2400$ (Cy-11/2400).

4.1.2 Sphere Prediction

There are a total of 24 data sets utilized for training the ANN. The resulting APD and MSE based on the different architectures are presented in Figure 4.4. Considering the APD, the majority of the architectures result in APD of about 37-39% at 200-400 iterations. Afterwards, at 800 iterations, the resulting APD for the architectures become sparse and distinguishable. The architecture that yields the lowest APD is $N_{HN} = 12$ and $N_I = 5000$ (Sp-12/5000). This combination of number of hidden nodes and iterations for optimization also yields the lowest MSE. In the Appendix, the average absolute difference between the predicted and actual grasp pattern is plotted against the the number of iterations for optimization, Figure 10, where the lowest yielding AD has the architecture of $N_{HN} = 11$ and $N_I = 5000$ (Sp-11/5000). Nevertheless, the architecture that will be utilized as a benchmark for testing will be $N_{HN} = 12$ and $N_I = 5000$ (Sp-12/5000). For comparison, another architecture that yields similar error (in both APD and MSE) is $N_{HN} = 11$ and $N_I = 3000$ (Sp-11/3000). This architecture will also be used to experiment and compare with the first choice architecture.

4.2 Artificial Neural Network Prediction

After the investigation of appropriate ANN architectures for both cylinders and spheres, it was concluded that:

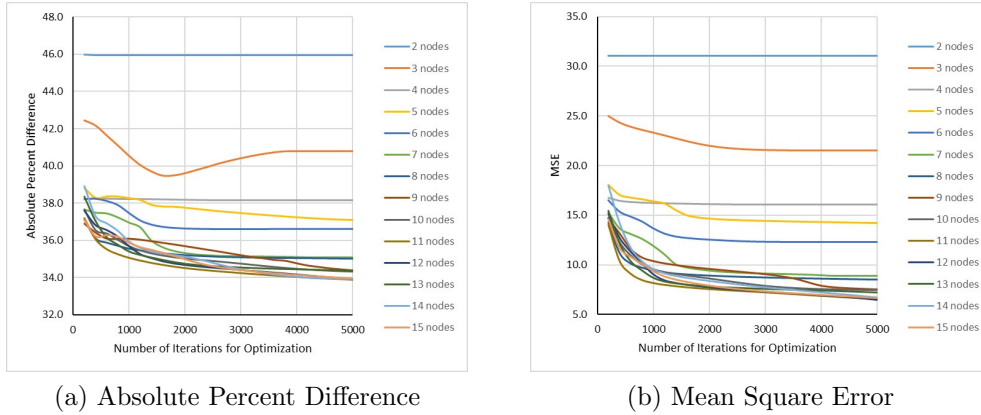


Figure 4.4: The Absolute Percent Difference and Mean Square Error of the resulting ANN architectures when the number of iterations for optimization is varied as functions of number of nodes in the hidden layer. This training was performed with 24 data sets.

- For cylinders, the first choice is Cy-13/4600 and second choice will be Cy-11/2400.
- For spheres, the first choice is Sp-12/5000 and the second choice will be Sp-11/3000.

These architectures will be used to compare the actual grasp pattern with the predicted motor angles. This comparison will be analyzed in both software and hardware for verification.

4.2.1 Software Implementation

There were 4 patterns, in both the cylinder and sphere category, that were not used for training. These patterns will be used to test the predictive ability of the four architectures (two architectures for cylinder prediction and two for sphere prediction) in software.

4.2.1.1 Cylinders

As was performed in Subsubsection 3.4.1.1 with the first generation hand, training was performed with 76 data sets and testing would be performed on 4 data sets. In particular, the four data sets are noted as Test1, Test2, Test3, and Test4, on object C22, with $D = 2.129''$, $L = 4''$ and relative height $H = \{3.5, 2.5, 1.5, 1\}''$ respectively. It is important to note that the same data sets were used for testing in both versions of the hand.

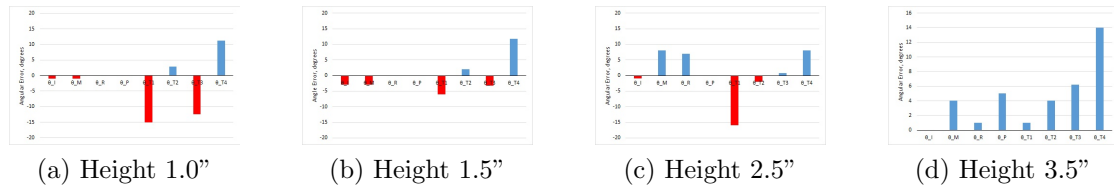


Figure 4.5: Angular error between the actual and ANN predicted angles for grasping cylinder C22. The predictions were performed with Cy-11/2400 trained with 76 data sets. Blue bars indicate *overgrasping* and red bars indicate *undergrasping*.

Considering the resulting angular error of Cy-11/2400, as presented in Figure 4.5, there are some noticeable observations. For instance, height 1.5" resulted in the lowest overall angular error while height 3.5" resulted in the highest overall angular error. The joint that resulted in the highest error throughout all tests was thumb 4, θ_{T4} . On the other hand, the joint that resulted in the lowest overall error is thumb 2, θ_{T2} . This is acceptable, since the thumb joints that will be providing the main motion for power grasps are θ_I , θ_M , and θ_{T2} . Note that θ_{T2} is the main joint in the thumb that provides abduction/adduction to perform power grasps, [63].

Considering the resulting angular error from Cy-13/4600, presented in Figure 4.6, similar observations were noted as with Cy-11/2400, such as predictions at height 1.5" resulting in the lowest overall angular error and at height 3.5" resulting in the

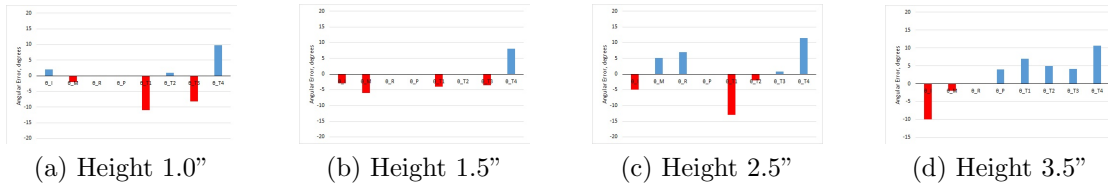


Figure 4.6: Angular error between the actual and ANN predicted angles for grasping cylinder C22. The predictions were performed with an Cy-13/4600 trained with 76 data sets. Blue bars indicate *overgrasping* and red bars indicate *undergrasping*.

highest overall angular error. In general, the angular errors are lower for the first ANN, in the majority of the joints for all four tests. Hence predictability improves with the first choice ANN (Cy-13/4600).

4.2.1.2 Spheres

As was performed with cylinders, the ability of the sphere ANNs to predict the sphere grasp pattern was examined. Training was performed on 24 data sets and testing on 4 data sets. In particular, the testing was on object S1 with $D = 2.25''$ at $H = \{1, 1.5, 2.5, 3.5\}''$.

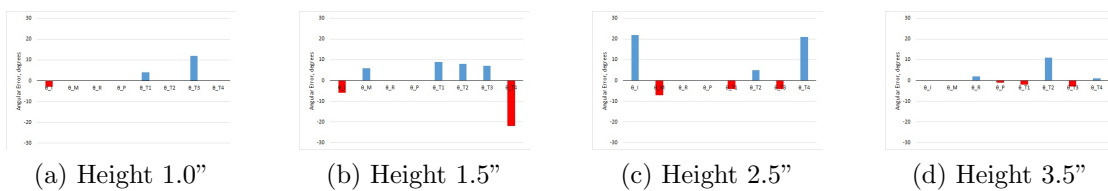


Figure 4.7: Angular error between the actual and ANN predicted angles for grasping sphere S1. The predictions were performed with Sp-11/3000 trained with 24 data sets. Blue bars indicate *overgrasping* and red bars indicate *undergrasping*.

Considering Sp-11/3000, presented in Figure 4.7, it was noted that the joint that consistently yields the highest angular error is θ_{T4} , where θ_P yields the lowest angular error. Another observation is that at height 2.5", θ_I is falsely predicted to

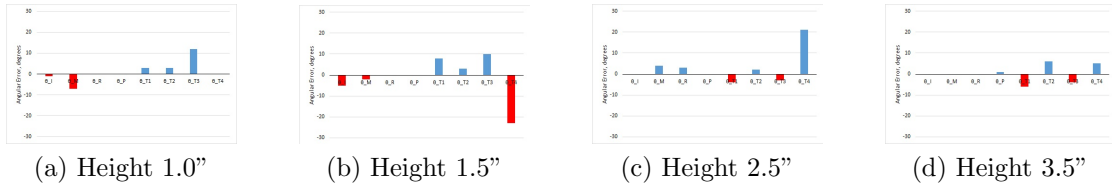


Figure 4.8: Angular error between the actual and ANN predicted angles for grasping sphere S1. The predictions were performed with Sp-12/5000 trained with 24 data sets. Blue bars indicate *overgrasping* and red bars indicate *undergrasping*.

activate by the ANN, which means that the actual value of θ_I is 0° is predicted to be $\tilde{\theta}_I = 22^\circ$, resulting in a relatively large error. The height that yields the most error is at 1.5" and the least error at 3.5" and 1". In a similar manner, some observations can be made with the angular error resulting from using predictions with Sp-12/5000, as presented in Figure 4.7. The thumb joints result in more angular error compared to the joints for the main fingers. However, unlike the performance comparison found in the previous section with cylinder grasp pattern prediction, here the first choice ANN does not significantly improve the overall prediction of the grasp, indicating that both architectures perform near-equally overall.

4.2.2 Hardware Implementation - Grasping *Known* Objects

After evaluating the performance of ANNs in software for predicting grasp patterns for cylinders and spheres, the next step is to validate their performance in predicted grasping patterns in hardware with the new artificial hand platform, H2. This was accomplished by sending signals to the actuators to move controllably to the ANN predicted motor angles, while an object is suspended by a string on the palm of the hand. All the motor angles would begin at their nominal position (fingers open exposing the palm of the hand) and gradually move to the predicted motor angle. After the fingers move to their predicted position, the string that is keeping

the object suspended is removed. If the object drops or slips out of the hand, the test is deemed *unsuccessful*. On the other hand, if the object is maintained in the grasp, then the grasp is deemed *successful*. In hardware, it is important to note that even if the grasp is deemed *successful*, there can still be *undergrasping* present for individual fingers. In physical space, *undergrasping* is the scenario of when an active finger reaches its final position, however, does not touch the object. Intuitively, at least two fingers are required, the thumb and another opposing finger, to *successfully* grasp an object, therefore having some fingers resulting in *undergrasping* does not necessarily deem the grasp *unsuccessful*. However, if a significant number of fingers result in *undergrasping*, then the object could potentially slip out resulting in being *unsuccessful*.

The timing of the finger motion when grasping the object is important, in order to result in a successful grasp. Based on the grasp study, it was observed that the thumb should be the first finger to reach its final state, since it acts as a structural ground, that resists/opposes the motion of object being grasped. After the thumb, the middle finger moves to its final position and then the ring and index, and finally the pinky. The controlled time dependent trajectory of each motor angle, from initial to final position was defined in a spline profile to reduce large changes in acceleration at the start and end of the motion, [90]. The motion profile for each finger with time delays incorporating the timing order of closing the fingers is presented in Equations 4.1-4.8.

$$\theta_{T1}(t) = a_0 + a_1t + a_2t^2 + a_3t^3 \quad (4.1)$$

$$\theta_{T2}(t - 2) = a_0 + a_1(t - 2) + a_2(t - 2)^2 + a_3(t - 2)^3 \quad (4.2)$$

$$\theta_{T3}(t - 2) = a_0 + a_1(t - 2) + a_2(t - 2)^2 + a_3(t - 2)^3 \quad (4.3)$$

$$\theta_{T4}(t - 2) = a_0 + a_1(t - 2) + a_2(t - 2)^2 + a_3(t - 2)^3 \quad (4.4)$$

$$\theta_M(t - 5) = a_0 + a_1(t - 5) + a_2(t - 5)^2 + a_3(t - 5)^3 \quad (4.5)$$

$$\theta_I(t - 6) = a_0 + a_1(t - 6) + a_2(t - 6)^2 + a_3(t - 6)^3 \quad (4.6)$$

$$\theta_R(t - 6) = a_0 + a_1(t - 6) + a_2(t - 6)^2 + a_3(t - 6)^3 \quad (4.7)$$

$$\theta_P(t - 7) = a_0 + a_1(t - 7) + a_2(t - 7)^2 + a_3(t - 7)^3 \quad (4.8)$$

Where the constants (a_0, a_1, a_2, a_3) are based on the desired end position and time for the motion and evaluated according to Equations 4.9-4.12,

$$a_0 = \theta_0 \quad (4.9)$$

$$a_1 = 0 \quad (4.10)$$

$$a_2 = \frac{3}{t_f^2}(\theta_f - \theta_0) \quad (4.11)$$

$$a_3 = -\frac{2}{t_f^3}(\theta_f - \theta_0) \quad (4.12)$$

4.2.2.1 Cylinders

The grasping predictions from Cy-11/2400 on for object C22 are presented in Figure 4.9. At every height, the object is grasped successfully, in other words, does not slip/fall out of the hand. However, it is observed that at height 3.5" (Figure 4.9 c) even though the object is being held securely, the pinky finger does not come into contact with the object. It is observed that C22 slightly tilted based on the contact by the other fingers. Note, the fingers close at specific times, and in this case, the thumb joints reach their final angular position first and then the pinky reaches

its final angular position last. Also, note that in Figures 4.5 and 4.6, the index and middle *undergrasp*. Even though the pinky *overgrasps* by 5 degrees, it is not sufficient to contact the object.

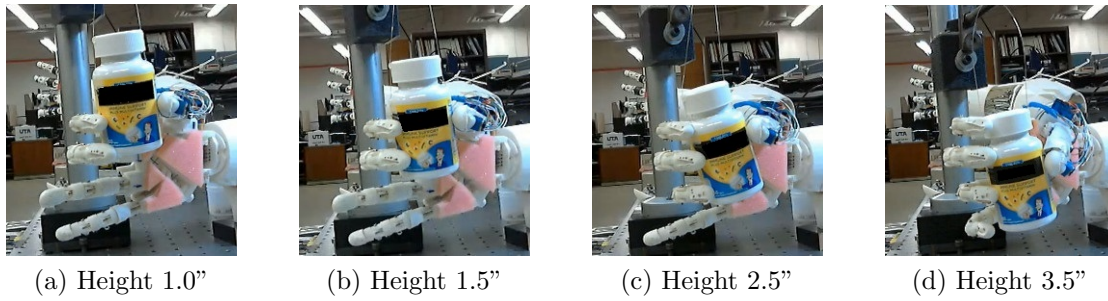


Figure 4.9: Images of the final grasp on object C22 from the resulting prediction of Cy-11/2400 trained with 76 data sets.

The test grasp was then performed with ANN-13/4600, with the images of the grasps presented in Figure 4.10. The grasps at all heights were successful, however, there were occasions where fingers did not come into contact with the object. At heights 1.0" and 1.5", all appropriate fingers required for grasping (thumb, index, and middle) made contact with the object, but at heights 2.5" and 3.5", some fingers did not come into contact with the object. It can be observed that at height 2.5" (Figure 4.10), the index finger comes close, but does not contact the object, this is reflected in the *undergrasping* prediction of the ANN as shown in Figures 4.7 and 4.8. Moreover, at height 3.5" in Figure 4.10, the index and the pinky under grasp and do not come into contact with the object. This is similar to what was observed in the second ANN grasp, however, the grasp prediction did not improve with the first choice ANN in hardware, hence Cy-13/4600 did not perform better than Cy-11/2400.

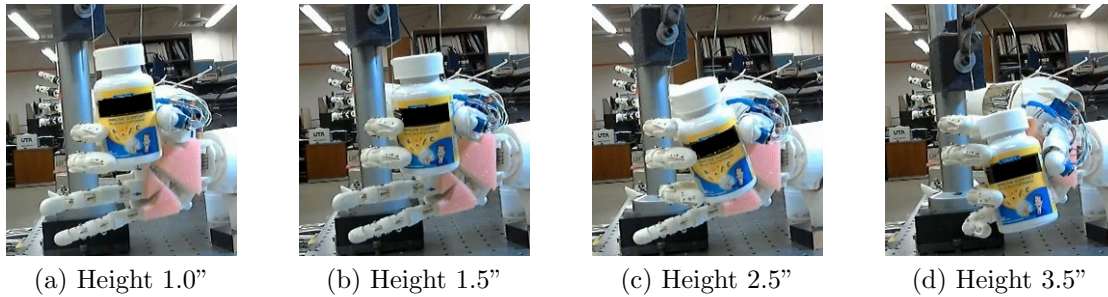


Figure 4.10: Images of the final grasp on object C22 from the resulting prediction of Cy-13/4600 trained with 76 data sets.

In conclusion, hardware testing resulted in successful grasps for all cases. These results coincide closely with the results observed in the software analysis. Even with all the test resulting *successful*, some of the fingers resulted in *undergrasping*.

4.2.2.2 Spheres

In Sp-12/5000 (first choice), the test was repeated in order to observe how well the prediction would result in a successful grasp of sphere S1. In Figure 4.11, the resulting grasps for Sp-12/5000 are presented. At both heights 1.0 and 1.5" (Figure 4.11) the index is activated and comes close to the sphere, but it does not make contact. The same is true with the pinky at height 3.5" (Figure 4.11), where the prediction is made to move a certain number of degrees, but does not come into contact with S1. The images of the resulting grasp with Sp-11/3000 (second choice ANN) are presented in Figure 4.12. An observation from the grasps is that the sphere was grasped at all heights and did not slip out of the hand. However, at height 1" (Figure 4.12), the index does not come into contact with S1. Again, at height 1.5" (Figure 4.12), the index finger does not contact the sphere. Additionally, at height 3.5" (Figure 4.12), even though the pinky finger is predicted to contact the object, it does not. In summary, for both ANN architectures, even though there was clear

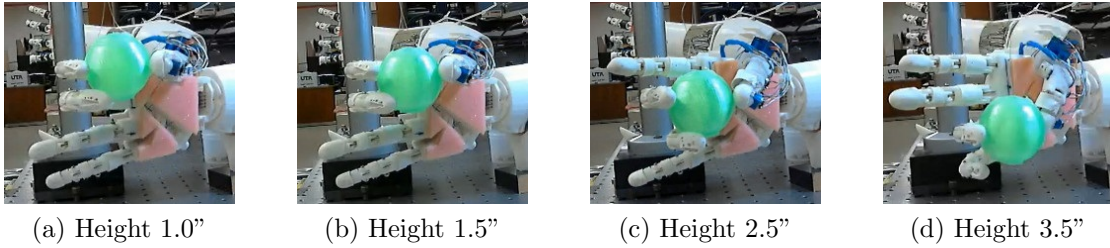


Figure 4.11: Images of the final grasp of object S1 from the resulting prediction of Sp-11/3000 trained with 24 data sets.

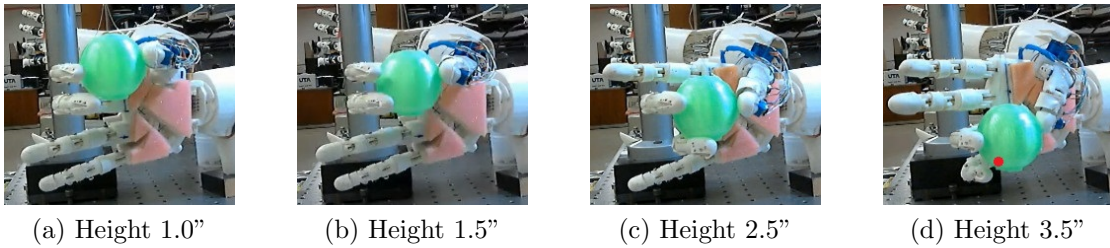


Figure 4.12: Images of the final grasp of object S1 from the resulting prediction of Sp-12/5000 trained with 24 data sets.

undergrasping present in the index and pinky, as was observed in the software analysis (Figure 4.7 and 4.8), the sphere was successfully grasped in all the tests. It was also observed that in hardware, there was no significant improvement in physical grasping between the more theoretically accurate Sp-12/5000 compared to Sp-11/3000.

4.2.3 Predicting Grasp Patterns of *Unknown* Objects

The predictive performance of the ANNs was evaluated with grasping of *unknown* objects not used in training or testing; objects for which the grasping pattern is not known *a priori*. There were 3 objects, two cylinders and one sphere, were used for testing the effectiveness of the ANNs, see Figure 3.37. The dimensions of the objects are tabulated in Table 3.1. The tests were performed at all heights $H = \{1.0, 1.5, 2.5, 3.5\}$.

4.2.3.1 Cylinders

In Figure 4.13, images of the final grasp with Cy-13/4600 for UC1 are presented. The predictions resulted in successfully grasping all the cylinder at all heights. However, at height 2.5” the index finger does not come into contact with the object. The images of the grasps performed with second choice ANN are presented in Figure 4.14. It is observed that at each height, the ANN predicted a grasp pattern that resulted in a successful grasp. It is important to note that at heights 2.5” and 3.5” the index finger *undergrasped* the object.

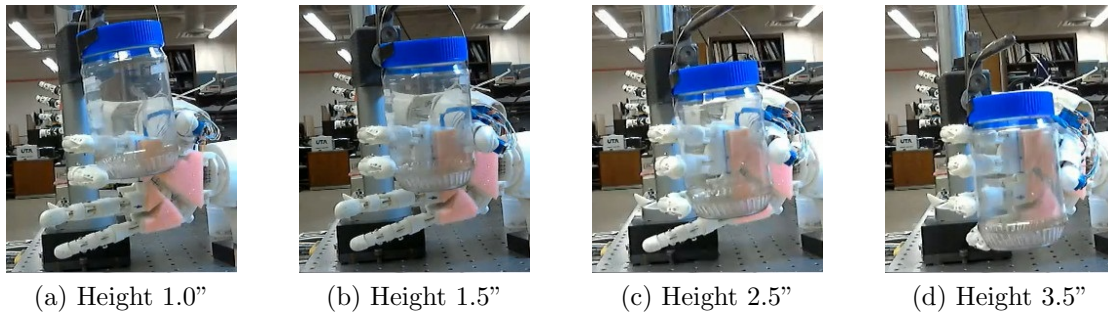


Figure 4.13: Final grasp positions of the fingers for object UC1. Cy-13/4600 was trained with 76 data sets.

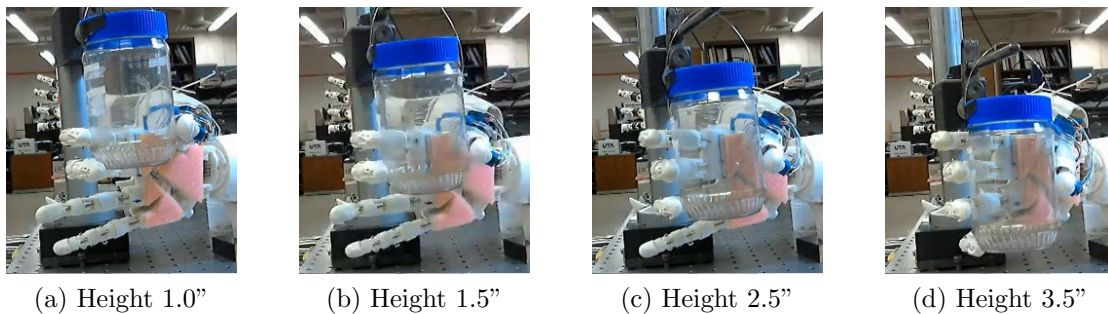


Figure 4.14: Final grasp positions of the fingers for object UC 1. Cy-11/2400 was trained with 76 data sets.

In Table 4.1, all the motor angles for each finger for all the grasps for cylinder UC1, for the two ANN architectures considered are tabulated. In the table, the underlined bold faced values indicate that even though the finger was activated, it did not make contact with the object.

Table 4.1: Final finger positions for grasping object UC1. The units for the fingers are in degrees.

ANN Architecture	Height (in)	θ_{T1}	θ_{T2}	θ_{T3}	θ_{T4}	θ_I	θ_M	θ_R	θ_P
Cy-11/2400	1.0	109	144	0	0	42	69	0	0
	1.5	120	140	0	0	42	70	0	0
	2.5	129	140	0	0	<u>33</u>	59	41	0
	3.5	136	133	0	0	<u>34</u>	57	41	46
Cy-13/4600	1.0	108	146	0	0	43	72	0	0
	1.5	115	139	0	0	45	73	0	0
	2.5	125	139	0	0	<u>39</u>	61	42	0
	3.5	135	134	0	0	40	59	40	48

For both sets of ANN architectures, the joint angles were similar with a low deviation. In the predictions, Cy-13/4600 had slightly higher values for joint angles. This is particularly apparent in the θ_I column, where the low joint angles predicted by Cy-11/2400 did not result in contact with the cylinder, whereas in Cy-13/4600 the index did make contact because the angles were slightly larger¹.

The images of grasping UC2 with Cy-11/2400 are presented in Figure 4.15. It is observed that at all heights, the UC2 was successfully grasped with all fingers coming into contact with the object. In Figure 4.16, the final images of the grasp of UC2 with Cy-13/4600 are presented. It is observed that at heights 1.0" and 1.5", the object slipped out of the hand. Even though the fingers contact the object, there was not enough force exerted on the object to prevent it from sliding out. Also at height 3.5",

¹For instance, at height of 3.5", Cy-11/2400 predicted 34° compared to Cy-13/4600 predicted 40°, which is slightly larger than the former prediction.

even though the pinky *undergrasps* and does not come into contact with the object, the object is successfully grasped and does not slide out of the hand.

For both ANN architectures, the motor angles were similar with a low deviation in value. In the predictions, Cy-13/4600 had slightly higher values for motor angles. This is particularly apparent in the θ_I column, where it was observed the low joint angles predicted by Cy-11/2400 did not contact the cylinder, whereas in Cy-13/4600 the index did make contact.

Table 4.2: Final finger positions for grasping object UC2. The units for the fingers are in degrees.

ANN Architecture	Height (in)	θ_{T1}	θ_{T2}	θ_{T3}	θ_{T4}	θ_I	θ_M	θ_R	θ_P
Cy-11/2400	1.0	109	151	0	0	51	86	0	0
	1.5	119	148	0	0	52	88	0	0
	2.5	129	151	0	0	46	81	53	0
	3.5	135	146	0	0	55	88	52	52
Cy-13/4600	1.0	110	152	0	0	48	83	0	0
	1.5	120	147	0	0	50	85	0	0
	2.5	130	151	0	0	47	80	51	0
	3.5	138	146	0	0	49	88	51	51

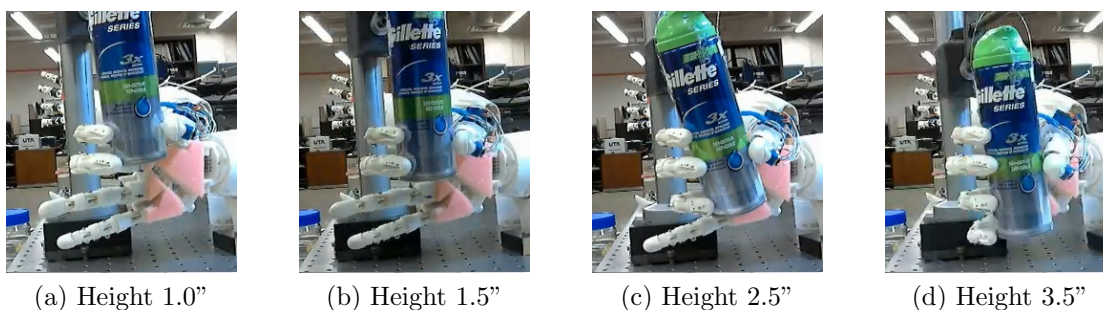


Figure 4.15: Final grasp positions of the fingers for object UC2 with Cy-11/2400, trained with 76 data sets.

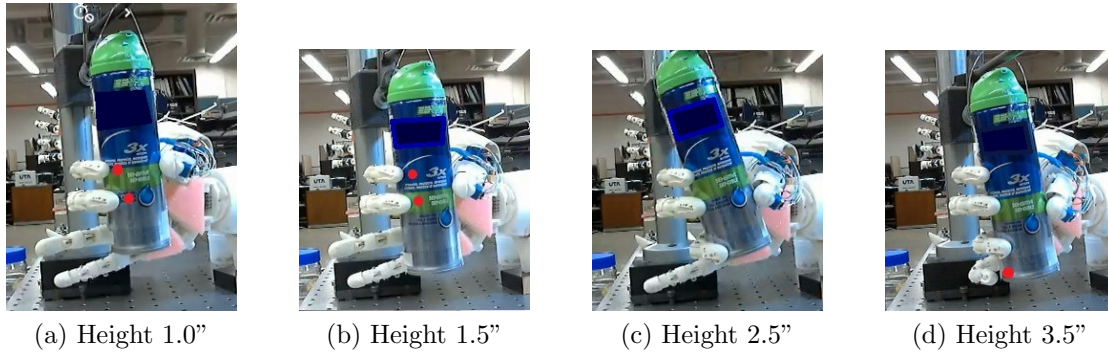


Figure 4.16: Final grasp positions of the fingers for object UC2 with Cy-13/4600 trained with 76 data sets.

In Table 4.2, all the motor angles predicted by both ANNs are tabulated for comparison. The entries in bold indicate that even if the motor was activated, the finger did not make contact. It is interesting to compare the predictions made by both ANNs at heights 1.0" and 1.5" for the index and middle fingers. For Cy-11/2400, the predictions were higher for θ_I and θ_M compared to the values predicted by Cy-13/4600. As a result of these low values predicted by Cy-13/4600, the object was *undergrasped* and slipped out of the hand.

Two distinct ANNs were tested for their ability to predict grasping patterns for *unknown* objects. Out of 16 test grasps, two were *unsuccessful* attempts at grasping. Another observation is that there were occasions where the hand was able to grasp the objects, however, some of the active fingers according to the prediction were not able to make contact with the object. It was also observed that the index had the most difficulty in coming into contact with the object. Nevertheless, *undergrasping* issues, could be addressed by incorporating additional sensors or other types of sensors, as will be discussed in Chapter 5.

4.2.3.2 Spheres

One unknown sphere, US1 was used to investigate the ability of the ANN to predict the grasp. In Figure 4.17, the images of the final grasps based on the predictions of Sp-11/3000 are presented. It is observed that at each height, the object is successfully grasped with the predictions made. However, at height 2.5", the index finger *undergrasps* and fails to contact the object. The resulting grasps for Sp-12/5000 are presented in Figure 4.18, and at all heights, US1 is successfully grasped. However, at height 2.5", the index finger is active but does not come in contact with US1.

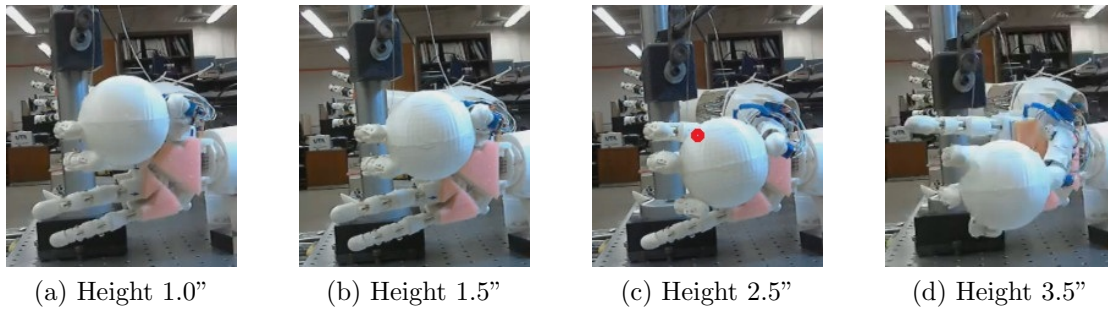


Figure 4.17: Final grasp positions of the fingers for object US1 with Sp-11/3000 trained with 24 data sets.

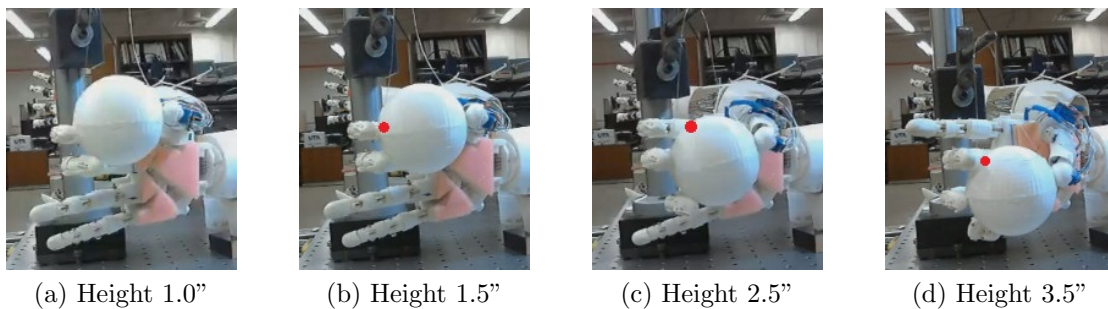


Figure 4.18: Final grasp positions of the fingers for object US1 with Sp-12/5000 trained with 24 data sets.

Table 4.3: Final finger positions for grasping object US1. The units for the fingers are in degrees.

ANN Architecture	Height (in)	θ_{T1}	θ_{T2}	θ_{T3}	θ_{T4}	θ_I	θ_M	θ_R	θ_P
Sp-11/3000	1.0	101	146	0	0	37	65	0	0
	1.5	109	147	0	0	41	69	0	0
	2.5	121	142	0	0	<u>37</u>	66	45	0
	3.5	139	133	0	0	0	<u>54</u>	36	57
Sp-12/5000	1.0	101	149	0	0	35	69	0	0
	1.5	108	144	0	0	41	65	0	0
	2.5	120	143	0	0	<u>38</u>	63	45	0
	3.5	140	134	0	0	0	<u>60</u>	38	55

In Table 4.3, the final angular positions for the two ANNs are tabulated. The numbers in underline bold indicate that the finger did not come into contact with the object or *undergrasping*. It is observed that for both architectures, the index *undergrasps* at height 2.5” and the middle finger *undergrasps* at height 3.5”. Nevertheless, the other fingers active in the grasp are able to “compensate” for the *undergrasp* of the finger and the object is held without falling or slipping, yielding a successful grasp.

In summary, both architectures are able to successfully predict grasp patterns for an *unknown* sphere. There are, however, instances where the object was not held by all the active fingers. As discussed earlier, this issue of *undergrasping* can be mitigated with the addition of other sensors, as will be discussed in Chapter 5.

4.3 Non Discriminatory ANN

Previously, ANNs were trained specifically with only spheres or cylinders, in order to later predict the angular positions of the joints to grasp a sphere or cylinder. This was performed by having the user first indicating to the system if the object is a sphere or cylinder, then proceed with providing the object’s dimensions and

relative height. It is of interest, however, to investigate an ANN that will not need the user to specifically provide the system whether the object is a sphere or cylinder, instead just provide the object dimensions and relative height. In this section, the investigation and selection of a ANN that will predict the appropriate grasp, based only on the object dimensions and relative height, without indicating whether the object is spherical or cylindrical in nature is introduced.

Given previous training data for grasping based on spheres and cylinders, a combined training set was developed resulting in 100 data sets for training. The inputs for the ANN are the object diameter (horizontal width) and length (vertical width) as well as it's relative height compared to the hand. The object dimensions are then scaled based on the dimensions of the hand, L_r and D_r , as mentioned in Section 3.5.

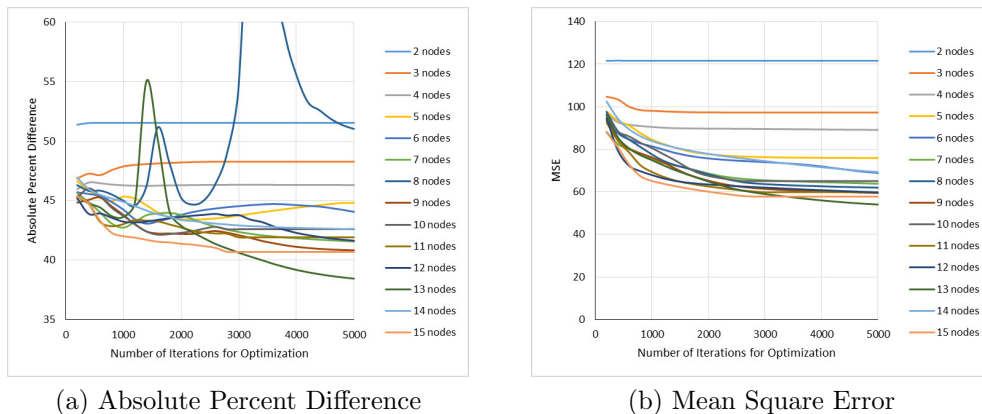


Figure 4.19: The Absolute Percent Difference and Mean Square Error of the resulting ANN architectures when the number of iterations for optimization is varied as functions of number of nodes in the hidden layer. This training was performed with 100 data sets.

After a training set was developed, the resulting MSE and APD were calculated for a variety of architectures. Figure 4.19 shows the resulting average APD and MSE

of the various ANN architectures. The APD begins around the same range of 43-48%. Afterwards, the values tend to decrease slightly, and after 3000 iterations, architectures are more spread out. The architecture yielding the lowest APD is with 13 hidden nodes and 5000 iterations for optimization. The resulting MSE based on the various architectures trained. At the beginning, the majority of the architectures have a MSE of about 90 to 110, and decrease uniformly to about 600-800 iterations. After 1400 iterations, there is a clear distinction between the architectures. The ANN resulting in the lowest MSE is 13 hidden nodes and 5000 iterations. Figure 11, in the Appendix, presents the average Absolute Difference resulting from the various ANN architectures. The ANN resulting in the lowest average Absolute Difference is 13 hidden nodes and 5000 iterations for optimization, which is the same architecture for both the MSE and APD. Therefore, the architecture that will be used for our first choice investigations will be CySp-13/5000.

The architecture that will be used as a second choice will be CySp-15/2000. The reason behind this is because the architecture has 4% APD higher, 1% AD higher and 10 units higher in MSE, therefore, similar in error metrics. Thus, the second choice ANN will be CySp-15/2000. In summary, two architectures will be used to for testing the predictive ability of a non-discriminatory neural network, specifically CySp-13/5000 and CySp-15/2000.

4.3.1 Software Implementation

Testing was performed on a sphere and a cylinder in order to compare later with the grasp prediction of ANN's dedicated to one type of geometry. The test performed investigated the prediction of grasping cylinder C22 at heights 2.5" and 3.5", and the sphere S1 at heights 1.0" and 1.5".

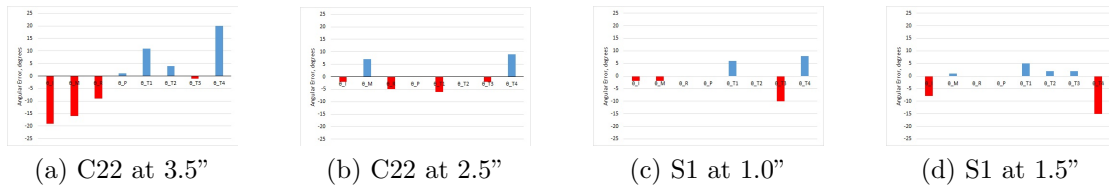


Figure 4.20: Angular error between the actual and ANN predicted angles for grasping sphere S1 and cylinder C22. The predictions were performed with CySp-15/2000 trained with 100 data sets. Blue bars indicate *overgrasping* and red bars indicate *undergrasping*.

Figure 4.20 presents the difference in angles between the actual and prediction with CySp-15/2000. As presented, the largest error is in the index and middle finger motor angles, which result in *undergrasping*. It is noticeable that *undergrasping* occurs most frequently when grasping cylinder C22 at height 3.5”.

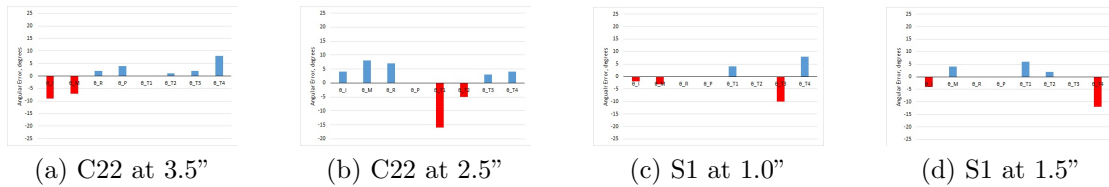


Figure 4.21: Angular error between the actual and ANN predicted angles for grasping sphere S1 and cylinder C22. The predictions were performed with CySp-13/5000 trained with 100 data sets. Blue bars indicate *overgrasping* and red bars indicate *undergrasping*.

In Figure 4.21, the angular error between the actual and prediction with CySp-13/5000, is presented for all grasps. It is observed that between CySp-15/2000 and CySp-13/5000, there is an improvement in prediction clearly shown in grasping cylinder C22 at 3.5”, by yielding overall less angular error. However, there is worse performance in grasping C22 at 2.5” and S1 at heights 1.0” and 1.5”. The largest issue resulting from the predictions is *undergrasping*.

In conclusion, an ANN trained with both cylinders and spheres was able to predict the resulting joint angles for both a cylinder and sphere. Among one of the most frequent issues was *undergrasping*, especially when predicting grasp patterns for cylinders.

4.4 ANN Architecture Comparison

The combined ANN (CySp) will be compared to the distinct sphere (Sp) and cylinder (Cy) ANNs in order to evaluate and compare its performance with the individual ANNs. The cylinder C22 test object data set with results using Cy-11/2400 and Cy-13/4600 will be compared to the performance of CySp-15/2000 and CySp-13/5000. Similarly, a comparison will be performed for sphere S1 with results from Sp-11/3000 and Sp-12/5000 will be compared to the performance of CySp-15/2000 and CySp-13/5000.

In the software analysis, the term *undergrasping* will be used which implies that individual predicted motor angles (equivalent to a predicted grasp pattern) are less than the actual ones. In other words, the prediction made by the ANN is *under* the actual angular value. This scenario is more distinguishable in hardware since *undergrasping* will result in an active finger not contacting the object. Additionally, *overgrasping* is the scenario in which the angular prediction is greater than the actual angle. In hardware, *overgrasping* would result in squeezing the object which could result in the object and/or the fingers being damaged.

4.4.1 Cy and CySp: Comparison of Predictive Performance

Figure 4.22 shows the error magnitude between the actual motor angles and the predictions by all the ANNs for grasping cylinder C22. It is observed that the predictions for all heights are mixed with both *under* and *overgrasping*. For grasping

C22 at height 2.5”, CySp-15/2000 results in the lowest error for 4 fingers (index, middle, ring, and pinky). Among these four fingers, the largest error is found in the prediction of the middle finger, θ_M . It is also noted that the architecture with the largest error is CySp-13/5000. The thumb joint that is most accurate in the prediction is joint θ_{T1} and finger θ_I .

For grasping C22 at height 3.5”, more error results from the predictions. It is observed that both the prediction of the index and middle finger position results in the largest error. On the other hand, the prediction of the ring finger is most accurate. The architecture that results in the highest error for all the joint angles is CySp-15/2000, while CySp-13/5000 results in the lowest error. Among the joints in the thumb, θ_{T2} and θ_{T3} have the lowest error in predictions.

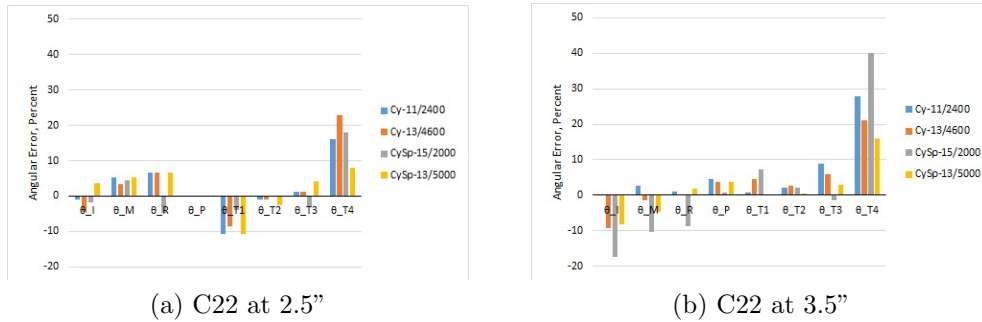


Figure 4.22: The angular error the actual and predicted joint angles for grasping. The results are provided by Cy-11/2400 and Cy-13/4600 trained with 76 data sets and CySp-15/2000 and CySp-13/5000 trained with 108 data sets.

Overall, it has been observed that the combined architectures performed better than the cylindrical ANN at both heights 2.5 and 3.5” for grasping cylinder C22. However, accuracy of motor angle predictions were not uniform across both heights. For instance, at height 2.5” CySp-13/5000 resulted in the largest error and CySp-

15/2000 in the lowest error. However, the converse is true at height 3.5”, where CySp-15/2000 yields the largest error and CySp-13/5000 the lowest error.

4.4.2 Sp vs. CySp

The errors between actual and predicted motor angles are presented in Figure 4.23. It is important to note that sphere S1 is grasped at heights 1.0” and 1.5”, therefore the only active fingers for the grasp are the index, middle and the thumb. Therefore, no real discussion could take place for the ring and pinky fingers, with respect to the actual motor angles since they are not active in grasping.

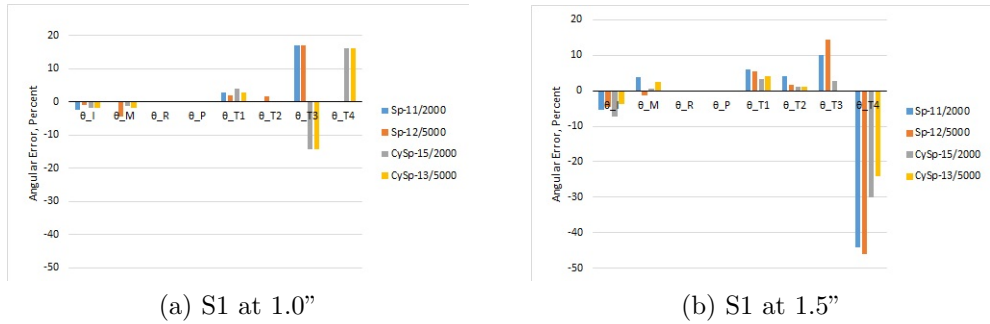


Figure 4.23: The angular error the actual and predicted joint angles for grasping. The results are provided by Sp-11/2000 and SP-12/5000 trained with 24 data sets and CySp-15/2000 and CySp-13/5000 trained with 108 data sets.

Among the thumb joints, θ_{T2} is the most accurate and θ_{T4} the least accurate. The most accurate architecture is CySp-13/5000, and least accurate are the sphere architectures Sp-11/2000 and Sp-12/5000. At height 1.5”, Figure 4.23, the sphere architectures are less accurate at predicting grasp patterns of the four fingers compared to the combined architectures (CySp), see Table 4.4. Among the thumb joints, the motor angle predicted with the lowest error is for θ_{T2} . In general, the combined ar-

chitecture CySp-13/5000 performs with the lowest error across the fingers and thumb joints.

Table 4.4: Average (all motor angles) absolute percent error of different ANNs with respect to *known* objects. Values presented are in terms of percentage.

ANN	S1		C22	
	1.0"	1.5"	2.5"	3.5"
Cy-11/2400	-	-	5.203	2.959
Cy-13/4600	-	-	6.041	6.053
Sp-11/3000	2.783	9.192	-	-
Sp-12/5000	3.268	9.129	-	-
CySp-15/2000	4.674	5.645	4.494	10.992
CySp-13/5000	4.588	4.409	5.131	4.702

In summary, for grasping sphere S1 at heights 1.0" and 1.5", the thumb joint that resulted in the lowest angular error is θ_{T2} . This is promising because this thumb joint provides adduction, which is the most influential joint for grasping. It is also observed that the combined architectures, CySp-15/2000 and CySp-13/5000, perform best at providing an accurate prediction of angular motor angles for grasping compared to architectures specifically dedicated to predicting motor angles for sphere grasping.

4.5 Grasping *Unknown* Near Spheres/Cylinders

To further verify the predictions of developed ANNs, we evaluated how well they can predict grasp patterns of *unknown* non-traditional shapes, and then applied it to the hardware of the biomimetic artificial hand. Non-traditional shapes are objects that do not fall easily in the category of sphere or cylinder. Hence, in this verification step, we will be using all 3 ANNs to predict the grasp of the object. Two objects will

be used for testing as presented in Figure 3.37, a decorative apple (UO1) and orange (UO2). Neither object has a uniform radius or fit in a specific shape category.

The geometric characteristics and box dimensions of the non-uniform shape objects are shown in Table 3.7. Note that the box dimensions are those of a rectangle superimposed to the side profile of the object, hence a horizontal and vertical dimension. These dimensions are used to evaluate the G_r and L_r ratios of the object. A limitation of taking the dimensions of a superimposed box over the object is that other details of the object shape along its axis are not captured. Moreover, distinct objects can have the same box dimensions which could potentially require more information about the object in order to be grasped appropriately. At this stage of research, we consider only convex objects.

The G_r and L_r ratios and the reference height for each object are presented as inputs to the ANN architectures for distinct cylinders (Cy-11/2400 and Cy-13/4600), distinct spheres (Sp-11/3000 and Sp-12/5000), and combined (CySp-15/2000 and CySp-13/5000). The predicted motor angles for grasping each object apple or UO1 and orange or UO2 at all heights (1.0, 1.5, 2.5, 3.5) and for each ANN are tabulated in Tables 4.5 and 4.6 respectively.

Subsequently, each object is placed in suspended at the correct height relative to the H2 biomimetic artificial hand and the motors are commanded to their final values using the motion profile in Equations 4.1-4.8. The performance of each predicted set of motor angles is discussed based on the observations from the hardware verification. Note that in Tables 4.5 and 4.6 numbers in underlined bold indicate that the respective finger was active but it did not contact the object, i.e. was *undergrasping*. Also, ANNs architectures in underlined bold indicate an *unsuccessful* grasp with the object not securely held when the string was released.

As presented in Table 4.5 and hardware observations, the index, θ_I , performed the poorest and experienced the most *undergrasping*. Moreover, *undergrasping* is observed when the object is located at larger heights. Also, there is noticeable motor angle boundary for the index motor in which *undergrasping* is present. For θ_I , at all heights and for all architectures, if the motor angle prediction was smaller than about 37° , it was observed that no contact was made with the object.

It is observed that at height 3.5", the index was not required for grasping based on the $G_r < 1.0$, however, the cylindrical architectures predicted the activation of θ_I . Cylindrical architectures are the least successful since one of the predictions resulted in a unsuccessful grasps (at height 3.5"). It is noted that the combined ANN architectures (CySp-15/2000 and CySp-13/5000) resulted in successful grasps as observed during hardware verification. The images for all the grasps for the different architectures and two test objects are presented in the Appendix in Figures 12, 13, 14, and 15.

The angular values predicted for object UO2 by all the architectures are tabulated in Table 4.6. Among some of the first observations is that the finger that results in the most *undergrasping* is the index finger, once again. At height 2.5", all the predictions made for the index finger result in *undergrasping*. Moreover, at height 3.5", Sp-12/5000 is the only architecture that has a successful grasp and all the active fingers come into contact with the object. The only occasions that result in *unsuccessful* grasping are at height 3.5", in particular Cy-11/2400 and Sp-11/3000. The images of all the grasps are presented in Figures 16, 17, 18, and 19.

In conclusion, the motor angles were successfully predicted for 94% (45/48) of the grasps with the various architectures since they resulted in successful grasps. However, at large heights such as 2.5" and 3.5", *undergrasping* was evident for both objects. Both the cylindrical and spherical architectures resulted in unsuccessful

grasps (UO1 with Cy-13/4600 at 3.5", UO2 with Cy-11/2400 at 3.5", UO2 with Sp-11/3000 at 3.5"), however, the non-discriminatory architecture only predicted successful grasps. The attractive aspect of the non-discriminatory ANN is that the user does not need to specify whether the object is a sphere or cylinder, the user only needs to input the horizontal width and vertical length, and the architecture will interpolate a solution between sphere and cylinder. This can be especially useful if a vision system is incorporated to capture the object major dimensions, where the combined architecture will predict the motor angles for the artificial hand. The issues regarding *undergrasping*, can be resolved if a controller is implemented, where the finger is required to contact the object in order to complete its task. The implementation of a feedback controller is part of the next step in this research, in Chapter 5.

Table 4.5: ANN predictions by various architectures on *unknown* object UO1 (apple). The motor angle values in underlined bold indicate an active finger not making contact with the object.

Height (in)	Architecture	θ_I deg	θ_M deg	θ_R deg	θ_P deg	θ_{T1} deg	θ_{T2} deg	θ_{T3} deg	θ_{T4} deg
1.0	Cy-11/2400	39	64	0	0	102	145	28	0
	Cy-13/4600	38	71	0	0	105	145	28	1
	Sp-11/3000	37	65	0	0	101	146	24	0
	Sp-12/5000	35	69	0	0	101	149	26	0
	CySp-15/2000	41	70	0	0	101	146	26	0
	CySp-13/5000	38	70	0	0	97	147	26	0
1.5	Cy-11/2400	37	63	0	0	116	141	31	3
	Cy-13/4600	<u>37</u>	70	0	0	111	139	30	5
	Sp-11/3000	41	70	0	0	109	147	24	3
	Sp-12/5000	41	65	0	0	108	144	23	3
	CySp-15/2000	42	70	0	0	105	145	28	1
	CySp-13/5000	42	69	0	0	110	145	28	0
2.5	Cy-11/2400	<u>26</u>	52	40	0	134	138	36	7
	Cy-13/4600	<u>29</u>	57	41	0	123	136	39	11
	Sp-11/3000	<u>37</u>	67	45	0	121	143	38	4
	Sp-12/5000	38	64	45	0	120	143	39	3
	CySp-15/2000	<u>30</u>	67	43	0	117	142	35	5
	CySp-13/5000	<u>31</u>	66	39	0	125	143	32	6
3.5	Cy-11/2400	<u>29</u>	<u>52</u>	43	<u>44</u>	143	129	50	10
	<u>Cy-13/4600</u>	<u>35</u>	59	41	<u>45</u>	137	131	50	12
	Sp-11/3000	0	<u>54</u>	37	57	139	133	45	19
	Sp-12/5000	0	60	38	54	140	134	44	20
	CySp-15/2000	0	<u>53</u>	39	53	137	131	49	15
	CySp-13/5000	0	<u>50</u>	37	51	143	131	46	17

Table 4.6: ANN predictions by various architectures on *unknown* object UO2 (orange). The motor angle values in underlined bold indicate an active finger not making contact with the object.

Height	Architecture	θ_I	θ_M	θ_R	θ_P	θ_{T1}	θ_{T2}	θ_{T3}	θ_{T4}
1	Cy-11/2400	40	66	0	0	103	145	28	0
	Cy-13/4600	40	71	0	0	105	146	27	0
	Sp-11/3000	<u>36</u>	62	0	0	100	143	22	0
	Sp-12/5000	<u>33</u>	72	0	0	98	149	24	0
	CySp-15/2000	40	67	0	0	103	145	27	0
	CySp-13/5000	39	68	0	0	96	146	27	0
1.5	Cy-11/2400	38	65	0	0	117	141	31	3
	Cy-13/4600	39	71	0	0	112	139	30	4
	Sp-11/3000	40	66	0	0	107	143	21	2
	Sp-12/5000	39	67	0	0	105	144	22	1
	CySp-15/2000	41	69	0	0	106	145	29	1
	CySp-13/5000	43	68	0	0	108	144	29	0
2.5	Cy-11/2400	<u>28</u>	54	40	0	133	139	36	7
	Cy-13/4600	<u>31</u>	58	41	0	123	137	38	10
	Sp-11/3000	<u>39</u>	66	44	0	119	139	34	3
	Sp-12/5000	<u>41</u>	62	48	0	116	144	39	0
	CySp-15/2000	<u>33</u>	67	42	0	119	143	34	5
	CySp-13/5000	<u>36</u>	68	41	0	123	143	33	4
3.5	<u>Cy-11/2400</u>	<u>31</u>	<u>55</u>	42	<u>44</u>	141	130	49	9
	Cy-13/4600	<u>37</u>	<u>60</u>	41	46	137	132	49	12
	<u>Sp-11/3000</u>	0	60	35	56	138	130	37	19
	Sp-12/5000	0	67	39	54	148	134	39	19
	CySp-15/2000	0	<u>57</u>	38	51	139	132	45	13
	CySp-13/5000	<u>21</u>	<u>59</u>	40	49	140	132	45	13

CHAPTER 5

Event Based Grasp Controller

Grasping has been performed with only kinematic information with respect to the angular position of the motors actuating the fingers. There has not been any feedback from the system to indicate that the object has been grasped or that the fingers have made contact with the object. Instead, ANNs have been trained to predict and provide the user the motor angles in order to grasp an object based on its relative location and geometric information. However, some of the drawbacks encountered are:

- An active finger did not contact the object. The ANN would predict a motor angle for an active finger, however the prediction would result in *undergrasping*. This affects the force distribution balance on the object as well as the risk of not exerting enough force to hold the object.
- Not being able to hold the object. Even with the predictions of motor angle positions, the grasp of the object is not guaranteed. There have been a few instances in which the object was not grasped due to *undergrasping*.

Both of these issues could be mitigated by adding a feedback system. An event based two-stage grasp controller which will help avoid “blind” grasping and ensure that the object is successfully and securely grasped is investigated.

The controller will incorporate a kinematic stage for the initial portion of the grasp. At the beginning, the motor angles required to grasp the object will be predicted using an ANN trained with both cylinders and spheres. This stage of the controller will be open loop and will not have any direct feedback from the envi-

ronment, hence “blind”. The second stage of the controller will ensure a successful grasp is made by incorporating sensor fusion in the grasping algorithm. The previous chapters have introduced and extensively discussed the results of the first stage of the controller based on pure kinematic grasping which has had positive results. This chapter will introduced the second stage of the controller based on sensor information and proceed with integrating the two stages in hardware demonstration.

5.1 Sensor Fusion Feedback - Hardware

In order to determine if the object has been engaged or not, feedback from the environment needs to be introduced to the control system. The two environmental sensors explored are a QTC force sensor and current sensor. The resulting information from both sensors will provide clear indication whether the object has been engaged or not by the fingers.

5.1.1 QTC Force Sensor

The QTC force sensor is used to indicate contact with an object. The sensors are placed in the tips of the fingers in two places; one location closer to the end, and the other closer to the interphalangeal joint since, based on the size of the object, a different contact region will be used to grasp the object (see Figure 3.14 in Chapter 3).

It was noted that the among all the grasps, the fingers most active were the index, middle, and ring. Therefore, it was decided that the sensors be placed on these fingers. Given the size of the sensor with respect to the approximate potential contact

area of the finger, two sensors were placed on each finger, Figure 5.1, to cover 66% of the contact area, yielding 6 sensors on the hand¹.

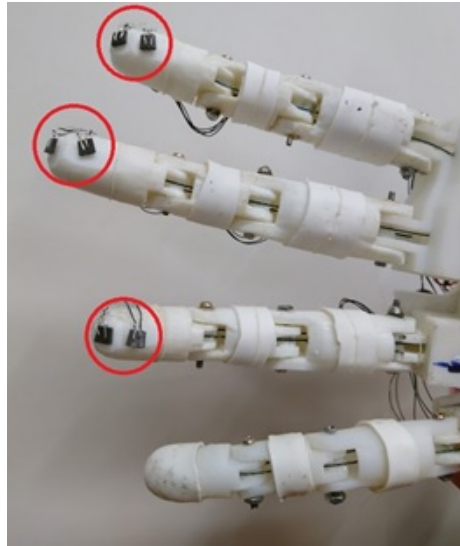


Figure 5.1: Images of the QTC sensors attached to the different locations of the finger tips.

Note that the QTC sensor output is only used as an indicator in the feedback that contact is made. Because contact is not repeatable, the surface area for contact cannot be predicted with high accuracy and as such the control scheme does not prioritize the QTC sensor.

5.1.2 Servo Motor Current Sensors

There are four current sensors to monitor the current consumption of the motor for the index, middle, ring and pinky fingers. The control scheme will consider a control action for the fingers and not for the thumb. More current drawn from the motor will relate to more torque output by the motor which will indicate that a finger

¹Note that we installed sensors on both H1 and H2 on the same locations since the fingers were the same for both hands

is interacting and “squeezing” an object during the stages of the grasping process. Hence, the grasping scheme will incorporate torque monitoring for control purpose [91], [92].

The current sensors use an Allegro hall effect transducer ACS712ELC, which determine a correlation between the current being pulled by a load, i_P , and output voltage, v_{out} , to the user, as presented in Equation 5.1.

$$v_{out} = i_P \times 0.25 + 2.5 \quad (5.1)$$

Where i_P is in Amps, and v_{out} is in volts. Now, since the current pull from the motors is known, the motor torque output can also be estimated. The servo motors for the fingers are MG996R Tower-Pro² and have a stall torque of 9.5 lb-in at 2.5 A. The running current for the motors are about 700 mA with no load. However, the bias current needs to be eliminated in order to normalize the offset. Therefore the relationship between the current pull, i_P , and the motor output torque, τ_m , is presented in Equation 5.2.

$$\tau_m = (i_P) \times 5.3 \quad (5.2)$$

The relationship between the output voltage and the motor torque output is given by Equation 5.3.

$$\tau_m = (v_{out} - 2.5) \times 21.2 \quad (5.3)$$

The value of voltage output provided by the current sensor can be used to estimate the torque output of the motor in real-time.

²Information about the servo motor can be found at the following URL: http://www.electronicoscaldas.com/datasheet/MG996R_Tower-Pro.pdf

5.2 Sensor Fusion Feedback - Algorithm

A study was performed to observe the sensor values while grasping objects and their resulting sensor values. All the objects, found in Table 1 in the Appendix, were grasped at all the standard measuring heights and both sensor values (current and force) were recorded. Note that the objects that were grasped were semi rigid and their weight did not exceed 2 pounds.

Figure 5.2 presents the frequency in which the force sensor was activated during grasping. The inset values are the number of times the sensor was detected (green) or not detected (red). Among the different sensors on the fingers, the case where a sensor was detected with most frequency was IndexL, however it was only activated 20% of the time³. Moreover, it was observed that the sensor values were either fully on or off when activated, and rarely (5%) was there a gradation in force sensed.

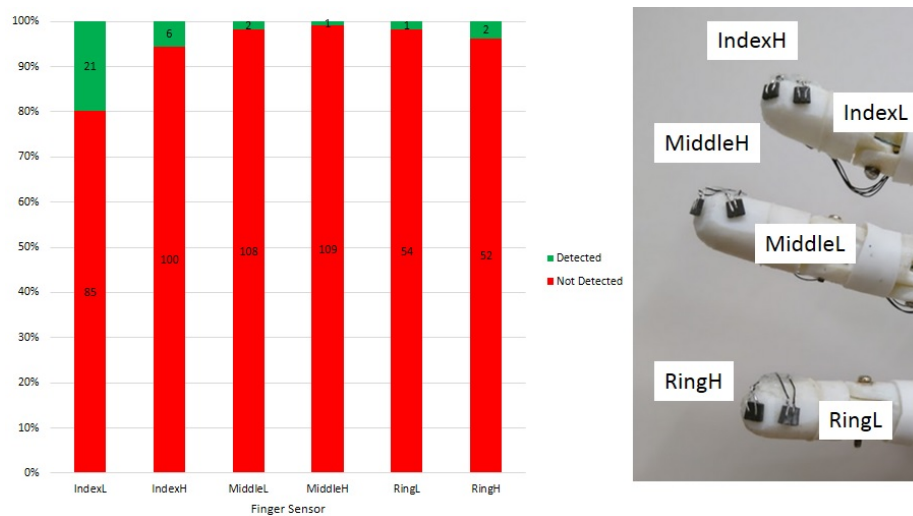


Figure 5.2: Frequency of sensor activation based on experimentally grasping objects.

³Out of 101 occasions, only 21 of those occasions resulted in a reading from the sensor

However, as opposed to the force sensors, the current sensors performed reliably at every contact. Figure 5.3 presents a histogram of the output torques required for each finger at various grasps. There are clear distinctions for the peak frequency values for each finger. The average torque values required to grasp all objects, for each finger, are presented in Table 5.1.

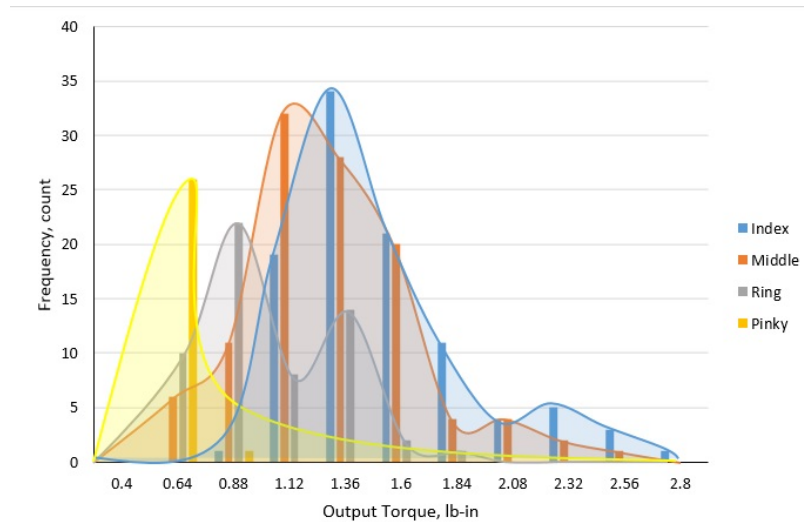


Figure 5.3: This is a histogram of the final torques required to grasp all the objects. There were 112 grasps consisting of different grasp poses.

Table 5.1: Average and Standard deviation of torque values (lb-in) required for a successful grasp, based on finger.

	τ_I	τ_M	τ_R	τ_P
Average	1.43	1.23	0.98	0.54
Standard Dev.	0.35	0.33	0.30	0.12

Based on observations of both sensors, it was determined that the current sensor performed more reliably than the QTC sensor. This is due to the uncertainty of finger contact area as well as contact dynamics during and after grasping. The sensors work

best when the contact force is normal to the sensor, however, based on the geometry of the object and under actuation of the finger, this scenario cannot be controllably replicated. Moreover, once the finger contacts the object, based on the actuation of the finger, the force direction vector changes, creating a sliding/slipping motion between the object and the finger. This motion cannot be detected with the QTC sensor and hence the non normal forces acting on the sensor surface are not captured to activate the sensor. Careful attention was taken to address these issues in terms of mounting and trying to optimize the normal contact force on the object, however, it was concluded, that at this time, the QTC sensor would not be incorporated in the closed loop controller due to its unreliable performance.

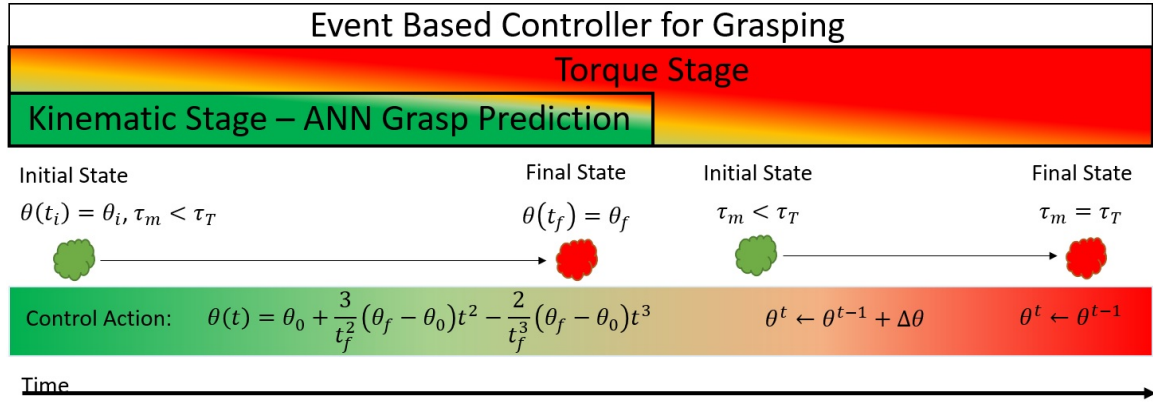


Figure 5.4: Decision making diagram for closed loop grasping algorithm

The proposed closed loop grasping algorithm is presented in Figure 5.4 with the steps being as follows:

1. Identify the geometry of the object and input to the system the diameter, D , length, L , and relative height, H .
2. Have the appropriate ANN (Cy, Sp, or CySp) predict the angular positions for each joint in order for the hand to grasp the object.

- (a) Kinematic Stage - Activate the motors to controllably reach the predicted angle in a given time, using Equation 3.12, for each finger

$$\theta_i(t) = \theta_0 + \frac{3}{t_f^2}(\theta_f - \theta_0)t^2 - \frac{2}{t_f^3}(\theta_f - \theta_0)t^3$$

Where t_f is the desired duration of the motion, and θ_0 and θ_f are the initial and ANN predicted final angle of each motor. However, an introduction to the Kinematic Stage of control is the constant monitoring of the motor torque. If the amount of motor torque output exceeds or equals the torque threshold, then the control signal stops the motor. The reason being that when the torque output exceeds the threshold, this will indicate a resistance to the finger motion on the physical system, most probably that the finger has made contact with the object. Therefore, to prevent object/hand damage (*overgrasping*) a stopping control ($\dot{\theta} = 0$) signal is sent to prevent the motors from reaching the ANN predicted final position θ_f .

- (b) Torque Stage - Once the fingers are at their predicted location, assuming that the motion of the fingers have not been interrupted by precipitated contact, the next control signal will be to continue motor motion until the torque values surpass a threshold. The torque thresholds, τ_t , used are the average values in Table 5.1. Hence, the control law for each finger, at every time step, will be

while $\tau_m \leq \tau_t$

$\theta_f(t) \leftarrow \theta_f(t - 1) + \delta\theta$

end

Where the subscript f indicates the different fingers on the hand that have current sensors installed (index, middle, ring, pinky).

The proposed controller is envisioned to perform in the following two ways to address the issue of *undergrasping* and *overgrasping*. Figure 5.5 indicates how the current sensor readings would ideally behave during an ideal grasping scenario with respect to time. The top graph represents the angular position of the motor and the bottom represents the sensor values of the torque output of the motor with respect to time. At the initial motion state, the motor begins at a zero position, similarly the torque value begins at a non-zero baseline torque value (based on the nature of having the motors on). As the motor moves, according to Equation 3.12, to the final predicted position, θ_f , there is a point of contact between the finger and object, as shown in Figure 5.5. At the point of contact, there is a noticeable rise in torque value from the sensor, due to the object/finger engagement. As the motor continues to move towards the predicted final position, the torque value increases until it reaches the torque threshold value. In an ideal scenario, the torque threshold value and the final angular value will be reached at the same time. This will indicate that the finger has reached a predicted position and that there is contact with an object surface.

It has been observed during the kinematic grasping experiments (Chapters 3 and 4), an issue arises when the finger does not make contact with the object based on the predicted final position of the motors, what was termed *undergrasping*. In the scenario of *undergrasping*, the motor reaches the final position, however, the torque values from the sensors have not reached the predetermined threshold. As presented in Figure 5.6, at the predicted final angular position, the torque sensor value has not changed from the base line torque value. This indicates that the finger has not come into contact with an object and there is no resistance to motion. In this scenario

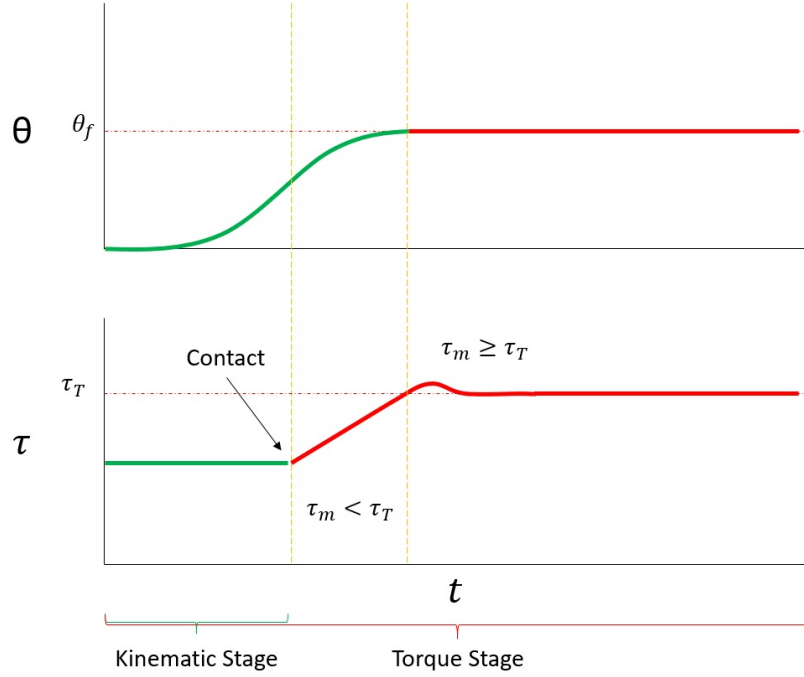


Figure 5.5: Schematic of ideal scenario in terms of sensor data during grasping event. The top graph is the angular position of the motors and on the bottom graph is the torque values from the sensors on the motors.

of *undergrasping*, the control action, by the event based controller, is to increase the motor angle value iteratively, hence 5.4.

$$\theta^t \leftarrow \theta^{t-1} + \Delta\theta \quad (5.4)$$

Keep in mind, that in process of iteratively increasing the angular value of the motor, if the value of the torque sensor is equal or surpasses the torque threshold value, then the control action is to stop the motor values from increasing.

Another issue already experienced that needs to be addressed is *overgrasping*, where the prediction of the ANN is greater than required to contact the object. In hardware, this can result in damaging the object or the hand, or indentation/perforation of the finger into the object. This *overgrasping* scenario is schematically presented in Figure 5.7. It is observed that the sensor value for torque exceeds

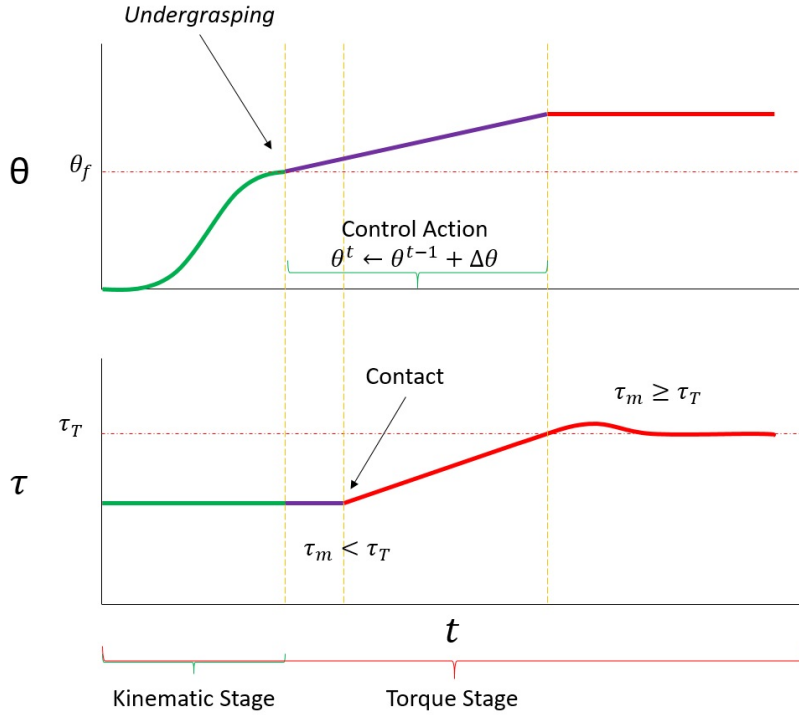


Figure 5.6: *Undergrasping* scenario during grasping event. The top graph is the angular position of the motors and on the bottom graph is the torque values from the sensors on the motors.

the threshold before the motor arrives to its predicted final position. It can be inferred that contact has been made and the finger is pressing firmly against the object, therefore it is not recommended that the finger keep closing to its predicted final position. As a result, the control action in this case is to stop the motors before reaching their final position.

These are the envisioned control methods by which both *undergrasping* and *overgrasping* are addressed with the event based two-stage controller. The performance of the proposed control algorithm was evaluated by performing test on previously grasped objects and comparing the grasps with the ANN predicted kinematic and force feedback controllers.

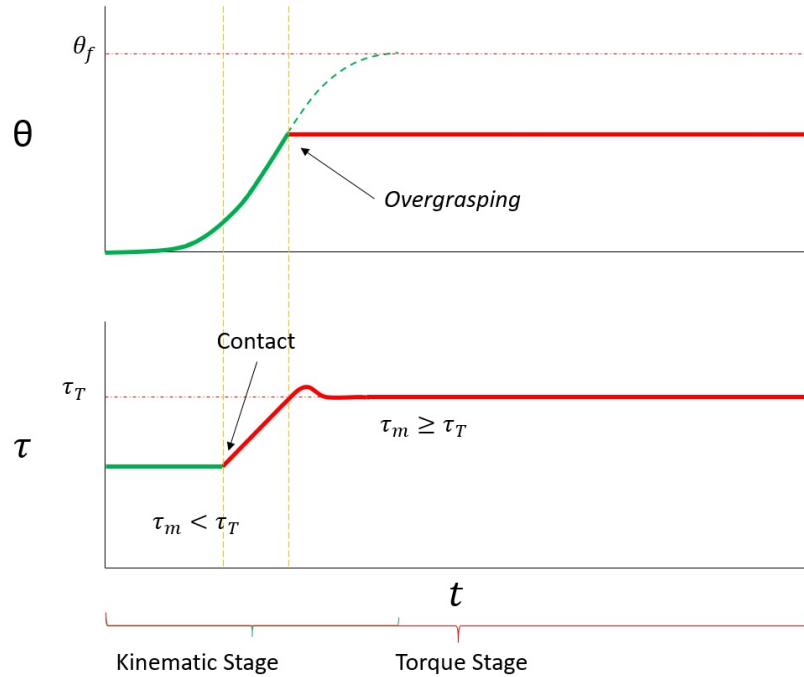


Figure 5.7: *Overgrasping* scenario during grasping event. The top graph is the angular position of the motors and on the bottom graph is the torque values from the sensors on the motors.

5.3 Grasp Comparison - Test Objects

Two objects were used for testing, cylinder C22 and sphere S1. The predictions based on the ANNs identified for investigation. The purpose of this comparison is to detail the observed improvements from the first iteration hand, H1, and the second iteration hand, H2, and then the introduction of the multiple types of ANN architectures.

5.3.1 Cylinder

The cylinder used for testing is C22. In Figure 5.8, the predicted grasp patterns are plotted against the actual grasp pattern for C22 at heights 2.5" and 3.5". The predictions compared are those that were previously used such as the cylinder based ANN (Cy-11/2400, Cy-13/4600), combined, (CySp-15/2000, CySp-13/5000) along

with the two stage controller with each ANN (Cy-11/2400-EB, Cy-13/4600-EB, CySp-15/2000-EB, CySp-13/5000-EB).

It is observed that for the fingers θ_I , θ_M , θ_R , and θ_P , the values for the closed loop control scheme have higher final motor angle values than those without closed loop control. The reason being that the control scheme torque values require the fingers to close more to create a firm grasp on the object.

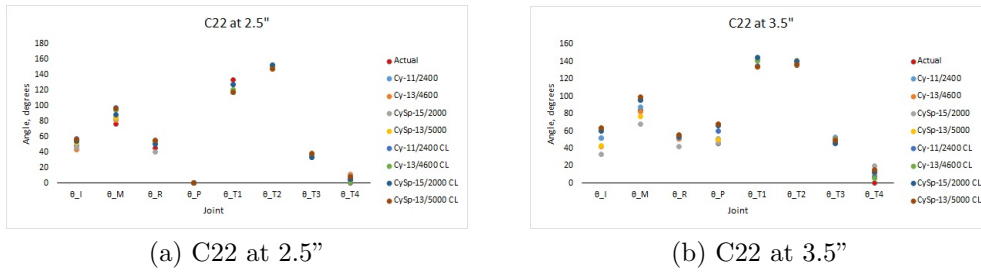


Figure 5.8: Comparison of grasp pattern predictions and actual motor angles for grasping C22.

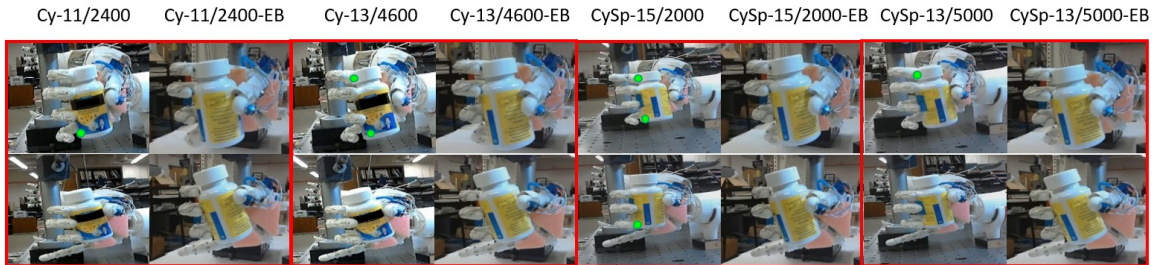


Figure 5.9: Images of the final grasp of all the open and closed loop predictions (top is at height 3.5” and bottom is at 2.5”).

Note Cy-13/4600 prediction of the grasp at height 2.5” results in the index *undergrasping* (Figure 4.9). However, *undergrasping* is eliminated with Cy-13/4600-EB. It should also be noted that at height 3.5”, the resulting grasp pattern from

Cy-11/2400 (Figure 4.9 d) and Cy-13/4600 (Figure 4.10 d) results in *undergrasping* for the index and pinky. Final images of the various grasps are found in Figure 5.9, where the figures on the right of the highlighted box include the use of the event based two stage controller. It is clearly shown that the event based controller eliminates the cases of *undergrasping* found when there was no controller and just using pure kinematics grasping based on ANN predictions.

In Figure 5.10, images of the final position of the fingers resulting from various grasping schemes are presented. The same test object, C22, is grasped with two different hands, H1 and H2, and with different control algorithms. It is noted that with H1 and H2, using pure kinematics to grasp the object, there are occasions when at least one finger does not contact the object (Cy-9/2600 at 3.5", Cy-12/4600 at 3.5", Cy-11/2400 at 3.5", Cy-13/4600), but in general all grasps result in not dropping the object. However, these issues are not present when the event based controller is used.

In summary, the event based two stage controller resulted in successful grasp and addressed issues of *undergrasping* and *overgrasping*.

5.3.2 Sphere

The sphere used for testing is sphere S1, grasped from height 1.0" and 1.5". The comparison of the resulting grasps from the pure kinematic predictions and controller at both heights are presented in Figure 5.11. A general trend observed in the non-thumb joints is that the final angles with the event based controller result in higher angular values compared to the purely kinematic predictions. Moreover, the resulting grasps with the event based controller are firm and secure.

The images of the final grasps for all the tests are presented in Figure 5.12. As previously, the images of the pure kinematic prediction are placed next to the

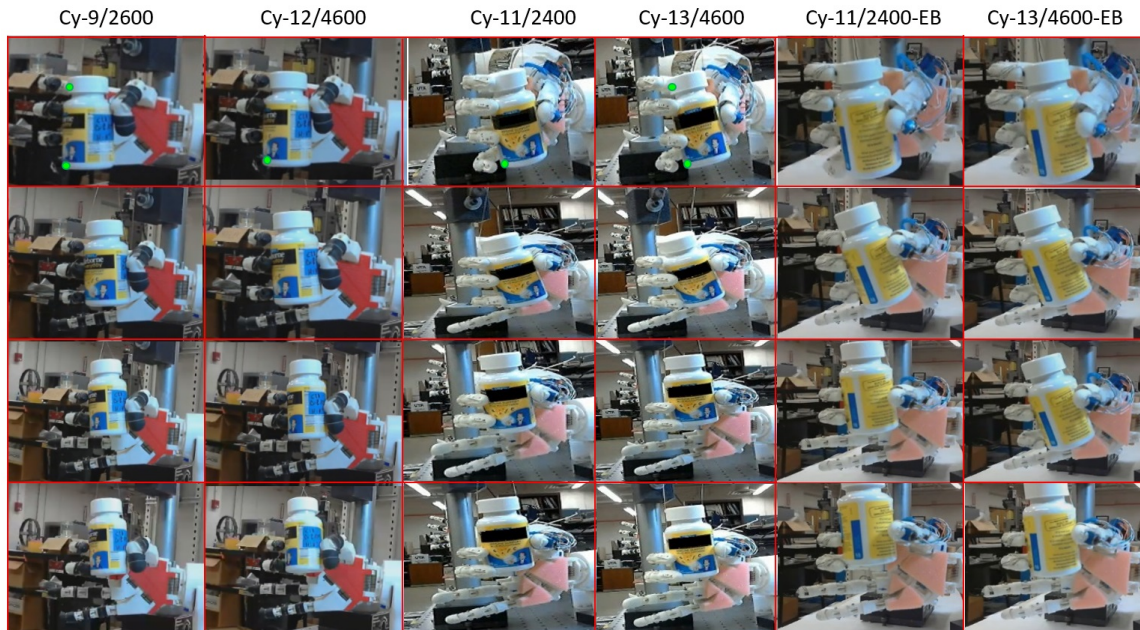


Figure 5.10: The stages of progression, from left to right, of the improvement of the development of the artificial hand (H1 and H2) as well as the control schemes utilized for grasping C22.

final grasps with the addition of the event based controller. An overall perspective of grasping sphere S1, with H1 and H2 and addition of the event based two-stage controller is found in Figure 5.13. It can be noted that architectures that did not use the controller, the finger would not come into contact with the object (such as in Sp-11/300 at 3.5" and Sp-12/5000 at 3.5"). However, after the introduction of the controller, the hand was able to securely grasp the object without any occasions of *undergrasping*.

In conclusion, in grasping both test objects, C22 and S1, the event based two-stage controller was able to securely grasp the objects. Moreover, issues observed previously using pure kinematic grasping, such as *undergrasping* and not grasping, were not present when the two stage controller was employed.

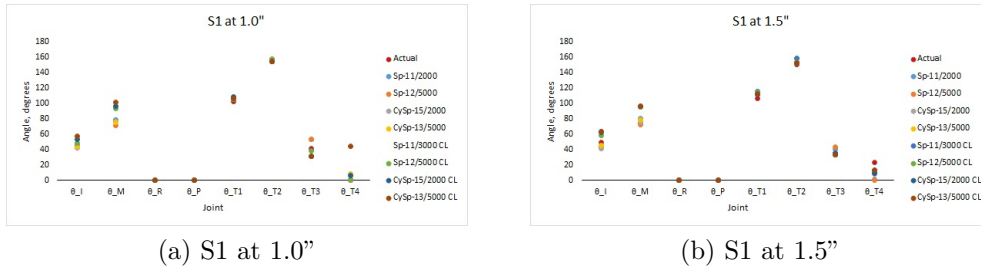


Figure 5.11: Comparison of grasp pattern predictions and actual motor angles for grasping S1.

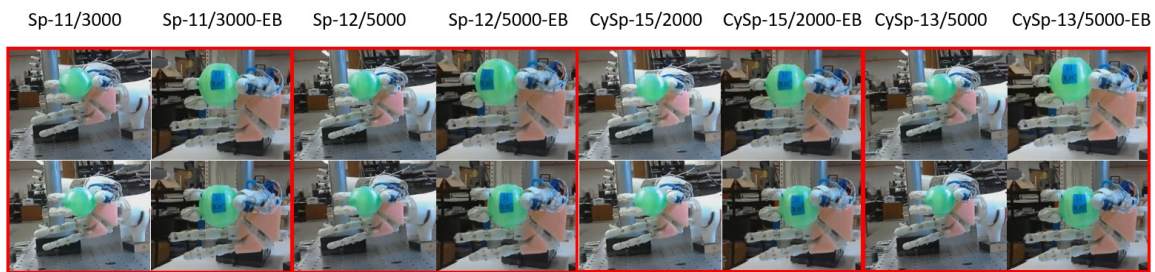


Figure 5.12: Images of the predicted grasps on object S1 for the open and closed loop control schemes. On the top the grasps are made at a height of 1.0'' and the bottom row are made at 1.5''.

5.4 Grasp Comparison - Unknown Objects

Unknown objects refer to lack of knowledge of the actual grasp pattern required to hold the object. Therefore, there cannot be a comparison with an experimental benchmark to indicate the accuracy of the predictions made by the ANN. The same objects, UC1, UC2, and US1, are with the event based two-stage controller.

5.4.1 Unknown Cylinders

When grasping objects UC1 and UC2, the predictions made by all ANNs are tabulated in Table 2 and 3. In these tables, the numbers in bold indicate an active finger that did not make contact with the object. For UC1, the finger that result in *undergrasping* are θ_I and θ_P . Lower bounds are observed where the event of *undergrasping* occurs, such as about 40° at 2.5'' and 3.5'' for θ_I and 46° at 3.5'' for

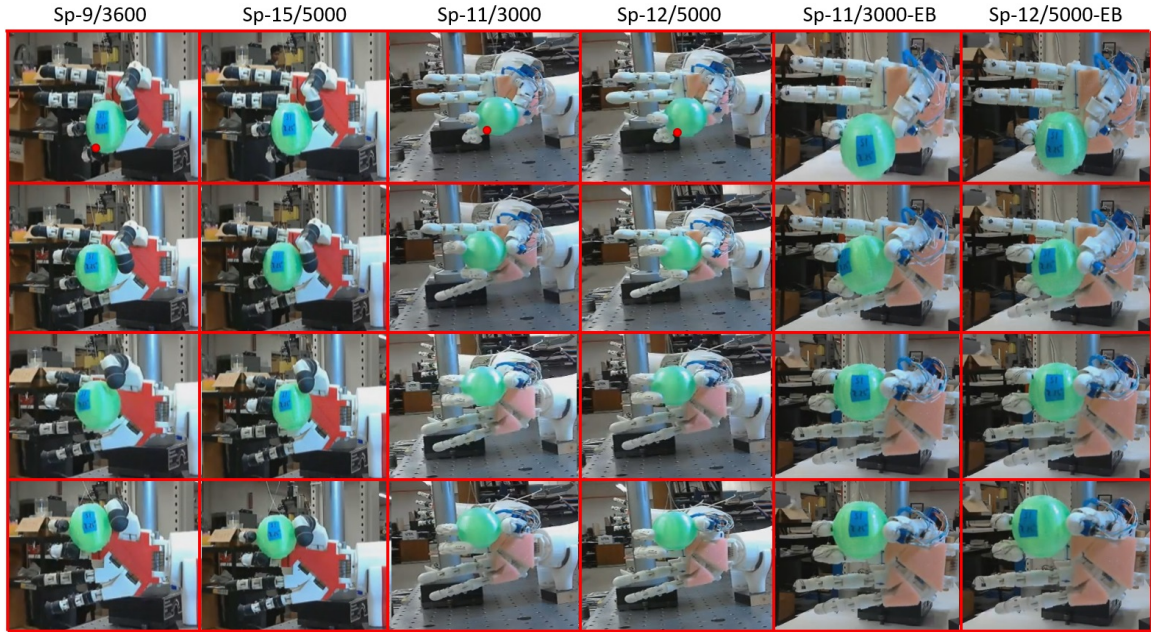


Figure 5.13: The stages of progression, from left to right, of the improvement of the development of the artificial hand (H1 and H2) as well as the control schemes utilized for grasping S1.

θ_P . Figure 5.14 presents all the grasps performed in the last major row in Table 2. Note that all the open loop grasps are juxtaposed with the CL grasps, moreover, there are no occasions of *undergrasping* for the grasps with event based controller.

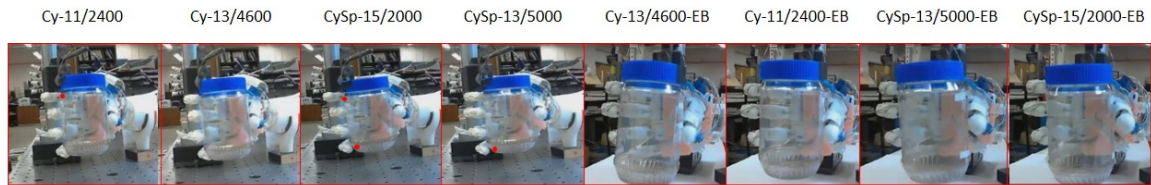


Figure 5.14: Comparison of all the grasps that were performed with the various control algorithms. Red dots indicate the location of *undergrasping*.

From Table 3, it is observed that *undergrasping* occurs more frequently at height 1.5". Similar to Table 2, there are observable boundary angles that result in *undergrasping* such as 52° for θ_I at all heights. Figure 3 presents the resulting grasps from

all the ANN architectures at a relative height of 1.5". Note that on two occasions, the object fell from the hand (Cy-13/4600 and CySp-13/5000) due to insufficient friction and contact. The bottom of the object should be horizontally aligned near the middle finger, nevertheless, after the actuators reached their final predicted position, the tension line that keeps the object suspended was dropped and the object slipped below the hand.

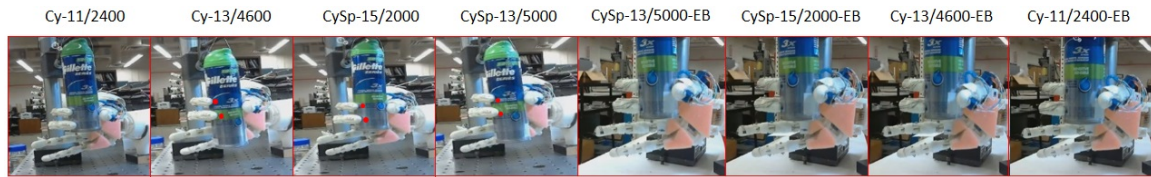


Figure 5.15: Comparison of all the grasps that were performed with the various control algorithms. The red dots indicate the location of *undergrasping*.

In conclusion, some of the major observations are that the grasps from the event based two-stage controller result in successful grasps. Also, the relative height that results in the highest frequency of *undergrasping* occasions is 3.5" when the controller was not used.

5.4.2 Unknown Spheres

The unknown sphere used is US1, and the final grasp pattern from the various control algorithms is tabulated in Table 4. One of the initial observations is that at relative height 3.5", for the open loop kinematic grasps, the middle finger *undergrasps* at all cases. Nevertheless, when introducing the event based controller, the middle finger, as well as all the other fingers, make contact with the object.

The unknown sphere was able to be grasped successfully in all cases, however in some cases, one or more active finger was not able to contact the object. This is

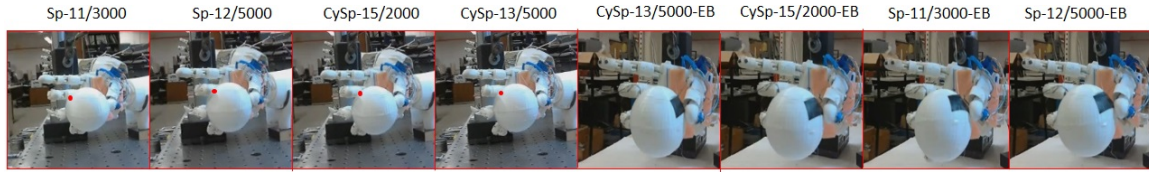


Figure 5.16: Compared grasps on object US1 at a relative height 3.5”.

not the case with the event based controller. These results mimic those found when grasping the unknown cylinders UC1 and UC2.

5.5 Grasping Near Cylinders/Spheres

Some objects to be grasped are neither perfect cylinders nor spheres. Two objects were used to test the robustness of the ANN predictions in being able to predict a successful grasp. The objects selected for testing are UO1 and UO2, found in Figure 3.37. In this section, there will be a comparison between the grasps using pure kinematic open loop control with ANN predictions, and the grasps resulting from the event based two stage controller. It is worth noting that some of the kinematic grasps were unsuccessful, hence the object dropped and was not held by the hand. Only the first choice ANN architectures will be considered for each grasp, hence Cy-13/4600, Sp-12/5000, CySp-13/5000, Cy-13/4600-EB, Sp-12/5000-EB, CySp-13/5000-EB. Final images of the grasps are presented in Figures 5.17 and 5.18.

Figure 5.17 presents the comparison of the grasps on object UO1 at all relative heights (1.0, 1.5, 2.5, 3.5”) with the various selected ANN architectures. It is noted that all the grasps performed by the event based controller were successful. However, for the cases where the algorithm for grasping did not include feedback for control, only one of the cases resulted in the object dropping (Sp-12/5000 at 3.5” height). It is also noted that the images contain green dots, which indicate that the active finger for grasping did not make contact with the object. This is observed for the index

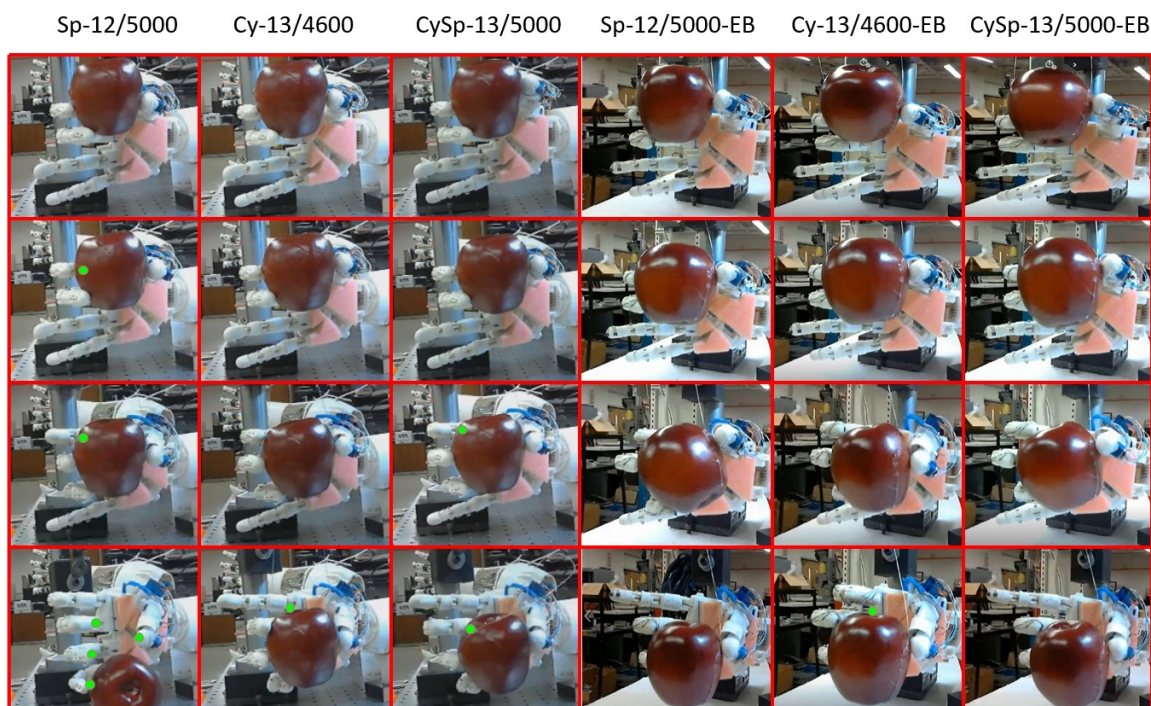


Figure 5.17: Various images of the comparison for grasping object UO1. The green dots indicate locations where there is *undergrasping*. The top row grasp is at height 1.0” and progresses down to the last row of 3.5”.

finger at 2.5” and then the middle finger at height 3.5”. Another observation is the error in active finger prediction made by the ANN. At a relative height of 3.5”, the index finger is not required because it is observed that the finger would not come into contact with the object, however Cy-13/4600 predicts that the index finger should actuate. As a result, the actuator for the index finger is set to the predicted position, then the torque stage of the controller provides the actuator with a control signal to increase until the torque sensor value has reached the predefined threshold. However, it is observed that the torque sensor values would not increase because there would be no object to oppose its motion. As a result, the finger closes all the way because the torque sensor does not trigger to stop of the motors.

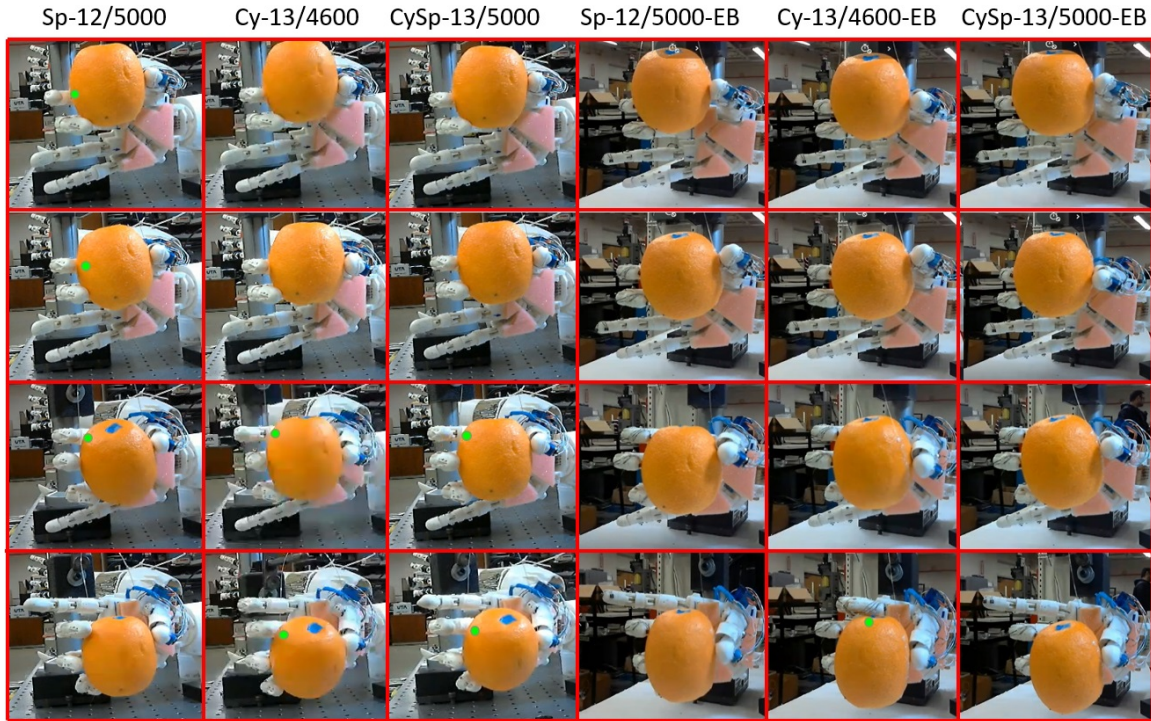


Figure 5.18: Various images of the comparison for grasping object UO1. The red dots indicate locations where there is *undergrasping*. The top row grasp is at 1.5” and progresses down to the last row of 3.5”.

In Figure 5.18, the final images of the grasp of object UO2 performed with various ANN architectures and controllers are presented. All the results gathered from the pure kinematic grasp and the event based controller result in successful grasps. However, the similar issues observed in grasping object UO1 were also present here. For instance, in the open loop control grasps, in various occasions, as indicated with a red dot, active fingers predicted for the grasp were unable to contact the object, resulting in *undergrasping*. On the other hand, *undergrasping* is not present with the closed loop controller, nonetheless, there is a issues with faulty prediction. At relative height 3.5”, Cy-13/4600 predicts that the index finger should grasp the object, however, this is a faulty prediction because the index finger does not make contact be the object. At height 3.5” the other ANNs (Sp-12/5000-CL and CySp-

13/5000-CL) are able to predict the correct number of fingers required for grasping, however, this is not the case for Cy-13/4600-CL.

In conclusion, the closed loop controller is able to address the issues of *under-grasping* found with the open loop control grasps. However, issues may arise with the prediction by the ANN. In the first stage of the controller, the ANN predicts which joints should be active and to what motor angle they should turn to adequately grasp the object. At this stage, the ANN could falsely predict the activation of a finger that in reality should not be active because it will not be able to make contact with the object. Therefore, after the ANN has false predictions, the second stage of the controller increases the motor angle until a preset torque threshold for grasping is experienced by the motor. A feasible way to alleviate this issues is by improved predictions of the ANN by increasing training data. Another potential solution would be to investigate additional sensors such as a proximity sensor, which could provide a check or verification that indeed the active finger should move its predictive position.

CHAPTER 6

Conclusions and Recommended Future Research

The goal of this research was to investigate a biomimetic hand could have the ability to grasp an object successfully using an event based two-stage data driven control method and to leverage machine learning algorithms for grasp pattern prediction. In this research, two biomimetic hands (H1 and H2) have been developed and the evolution and subsequent improvements have been discussed.

H1 was the first iteration hand that was manufactured and assembled in the MARS Lab for research in Human-Robot Interaction. It is a 5-DOF hand driven with tendons for actuating the fingers. Research was conducted to investigate and improve grasping with the assumption that the objects to be grasped would be rigid (non deforming) cylinders (21 count) and spheres (7 count) at 4 distinct heights of the object relative to the hand. A methodology was developed that studied the ANN architecture selection process for both shapes (discriminatory) and a non discriminatory ANN. The training procedure as well as the testing procedure for this stage of research as performed demonstrate the successful application of ANN machine learning algorithms for the prediction of appropriate motor angles for grasping. The predictions were tested in both software and hardware, on “known” and “unknown” objects, as well as objects that are neither cylindrical or spherical. Issues observed with this initial model, which included the contact area of the thumb as well as thumb positioning for effective and successful grasping. Two of these drawbacks are inherent of the system, therefore the second iteration of the artificial hand aimed to address these issues.

H2 was developed with the highlight of an improved thumb for grasping. Unlike H1, the thumb in H2 was not tendon driven, instead each actuator used for finger control was mounted at the joint. H2 is an 8-DOF hand, 4-DOF are from the tendon driven fingers and 4 from the individual joints in the thumb. The user now has the ability to control individual joints in the thumb for improved positioning. The same procedures of training the hand with the same cylinders and spheres and then testing were performed on “unknown” objects and objects that are neither cylindrical or spherical. The results from the experiments in both software and hardware were successful. The addition of a 4-DOF thumb improved the grasping process by providing a geometric base to act as supporting reaction base for grasping. A drawback from the thumb in H1 was that the contact area for the thumb was unorthodoxly positioned for grasping, which would also risk resulting in an unsuccessful grasp. This issue was ameliorated with the improved thumb since the user can train the thumb to position at a desired pose for contact with the object in the process of grasping. However, there was still an issue present in both versions of the hand, that was not addressed was *undergrasping*, *i.e.* an active finger predicted for grasping not contacting the object. This issue can lead to an unbalanced grasp or even dropping the object after the grasping process. This issue is inherent of the system since the trained ANNs predict the kinematic grasp pattern. Open loop control does not provide any information to the control system about the issue of “under” grasping, hence this will always be a challenge for the control system.

A event based two-stage data driven controller was then developed to mitigate the issue of *undergrasping* by producing a successful and secure grasp. Two types of sensors were investigated; force sensors at the finger tips to indicate contact with an object and current sensors were incorporated to the actuators to indicate the amount of current drawn by the motor. The current sensor readings were measured and

correlated to torque values and used for grasp strength understanding. An algorithm was developed to consider the stages of grasping, pre-hension and contact securely and hold. The first stage would prioritize the use of the kinematic control of the predicted motor angles and the second stage would use feedback from the two sensors to ensure that the active fingers make contact with the object. Training was performed with all the objects and sensor values were collected for every grasp to observe the sensor performance during each grasps. It was noted that the current sensor was more reliable than the force sensor in providing repeatable grasp information. After training, it was noted that average torque values had very small standard deviation for all objects, hence threshold values were used to indicate grasping of the object and an algorithm was developed to prevent *undergrasping*. The resulting algorithm required a two-stage controller ensured reliable grasps of all the objects both cylinders and spheres, as well as unknown objects and non cylindrical/spherical objects. Given this success, there are some issues observed during experimentation.

The system does not consider the weight of the object. Therefore, two objects with exact geometric properties will have different grasp patterns if the weights are significantly different. For a successful grasp, more torque will be required by the actuators to grasp the heavier object because there is greater risk of slip. The current control architecture does not have a way to detect slip nor to perform a control action to prevent slip. Slip detection would require additional sensors that would be able to capture relative motion between the object and the artificial hand. This is important to consider after grasping is achieved, since slipping can occur when manipulating or transporting an object. These challenges can be addressed in future research with the addition of new sensors and control algorithms.

6.1 Future Outlook

To improve the training process, more objects can be added. Furthermore, grasping non cylinders and spheres is recommended for future research. Some of the objects that grasped were not perfect cylinders or spheres, however they closely resembled a cylinder or sphere. Future research should investigate how to grasp objects that have non uniform geometries. If ANNs will be employed for grasp pattern prediction, the inputs of the ANN would need to adequately capture relevant object features for grasping. Therefore, a vision system would be recommended to investigate the identification and extraction of relevant object features in order for an ANN to predict grasp patterns for the desired object in real-time.

Another area of investigation are grasps other than power grasps. Throughout all the research presented here, the only type of grasp investigated is power. However, it is beneficial to investigate other grasps such as tripodal or later grasps based on the size and geometry of different objects. In order to perform this, a similar training process would be required for different types of grasps.

There is a high level of satisfaction with the progress made with this research with the biomimetic artificial hands H1 and moreover with H2. The Human-Robot Interaction research has been conveniently developed using NI LabVIEW, for ease of expandability to various interaction modalities and visualization of environmental signals. The following are areas which should be extensively researched, due to the already established research platform:

- Voice recognition and command execution for cooperative grasping
- Object recognition and feature extraction for grasp pattern prediction
- Upper arm prosthetics for grasping
- Brain-machine interfacing for brain controlled robotics/grasping

In summary, the Human-Robot Interaction platform is envisioned to expand into different areas of robotics research. Moreover, the development of the event based two-stage data driven controller will be used to further investigate different sensors and data fusion schemes fo improved interaction.

Appendix

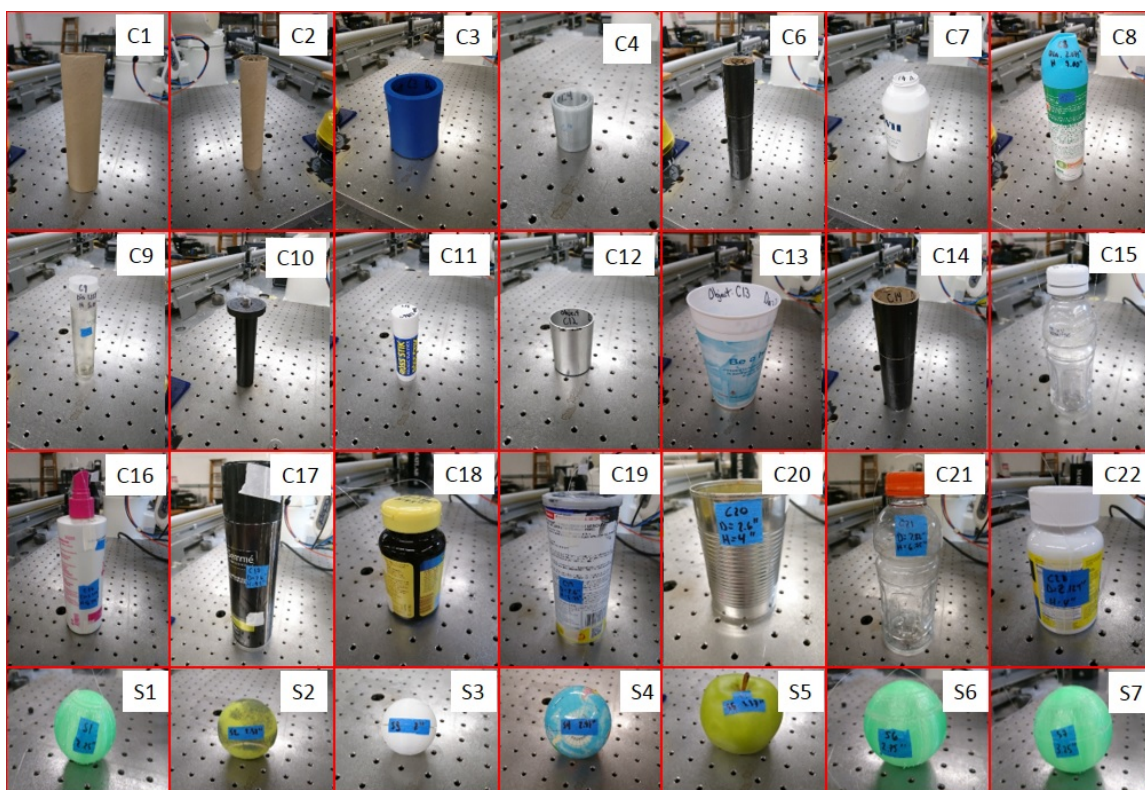


Figure 1: Images of all the objects used for grasping. Dimensions of the objects are presented in Table 1.

Table 1: Objects, spheres and cylinders that were grasped by the mechanical hand. Diameter and height of the object were give as well as its normalized dimension relative to the height of the hand and the grasp space.

Object	Description	Height (in)	Dia. (in)	Norm. Height (in/in)	Norm. Dia. (in/in)
C1	paper towel roll	7.76	2.03	2.069	0.549
C2	paper towel roll	7.95	1.59	2.120	0.430
C3	can koozie	4.00	3.23	1.067	0.873
C4	3D manufactured cylinder	1.88	1.40	0.501	0.378
C6	paper towel roll	8.12	1.671	2.167	0.452
C7	pill storage cylinder	3.54	2.51	0.944	0.68
C8	air freshener	7.00	2.07	1.867	0.560
C9	shampoo container	5.00	1.13	1.333	0.305
C10	aluminum stand	5.00	1.10	1.333	0.298
C11	glue stick	4.19	1.15	1.117	0.310
C12	aluminum cylinder	2.50	1.87	0.667	0.507
C13	plastic cup	6.50	3.50	1.733	0.946
C14	paper towel roll	7.81	2.10	2.083	0.568
C15	juice bottle	7.00	2.16	1.867	0.584
C16	hair product bottle	6.37	2.20	1.700	0.596
C17	hair spray bottle	9.50	2.6	2.533	0.703
Continued on next page					

Table 1 – continued from previous page

Object	Description	Height (in)	Dia. (in)	Norm. Height (in/in)	Norm. Dia. (in/in)
C18	pill container	4.50	2.37	1.200	0.642
C19	product container	6.25	2.60	1.667	0.703
C20	aluminum can	4.00	2.60	1.067	0.703
C21	sports drink bottle	6.75	2.82	1.800	0.762
C22	pill container	4.00	2.13	1.067	0.575
S1	3D printed sphere	2.25	2.25	0.600	0.608
S2	tennis ball	2.57	2.57	0.687	0.696
S3	3D printed sphere	2	2	0.533	0.541
S4	foam toy globe	2.91	2.91	0.776	0.786
S5	decorative apple	3.37	3.37	0.899	0.911
S6	3D printed spheres	2.75	2.75	0.733	0.743
S7	3D printed sphere	3.25	3.25	0.867	0.878

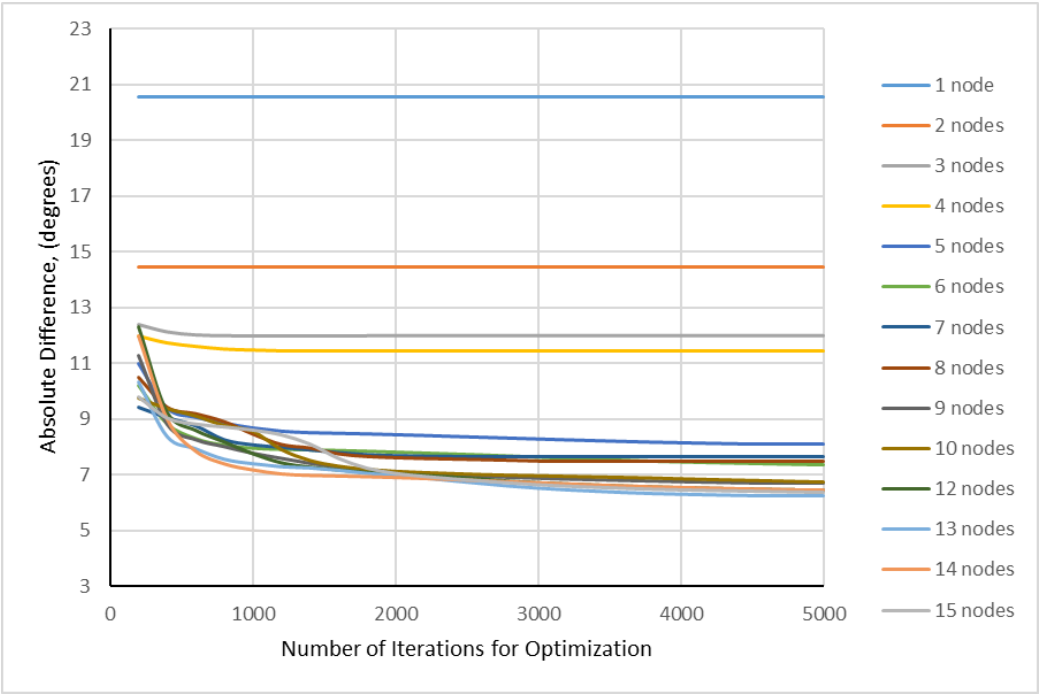


Figure 2: Absolute difference between the ANN prediction and the actual values, with respect to the number of iterations for optimization and number of nodes in the hidden layer. This was performed with 76 data sets for training.

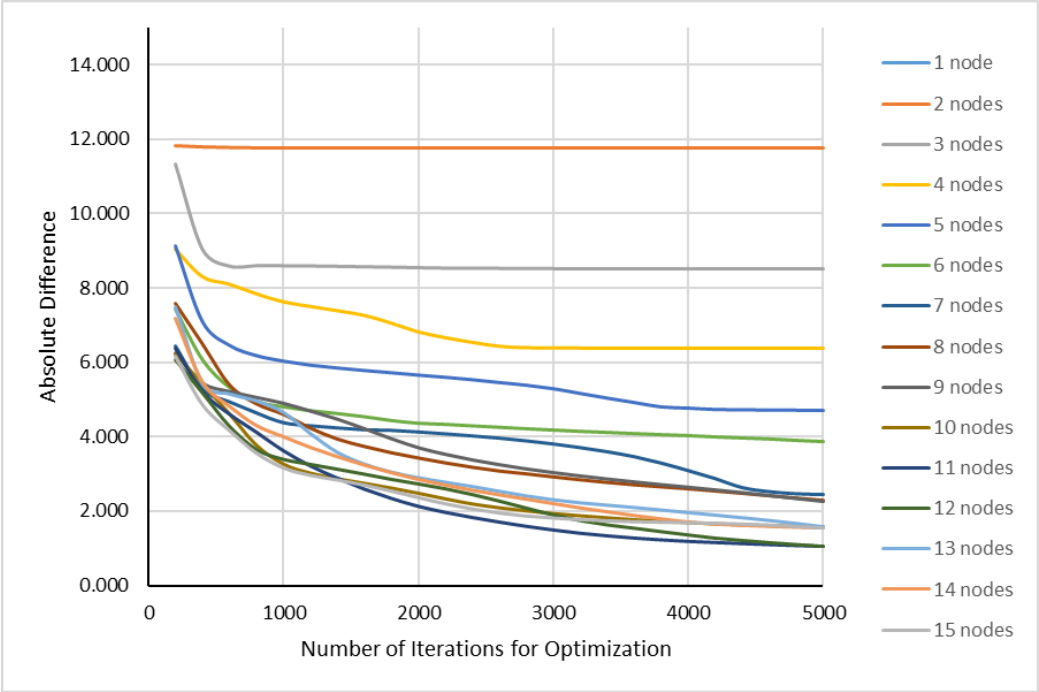


Figure 3: Absolute Difference of the resulting architectures. Training was performed with 24 data sets.

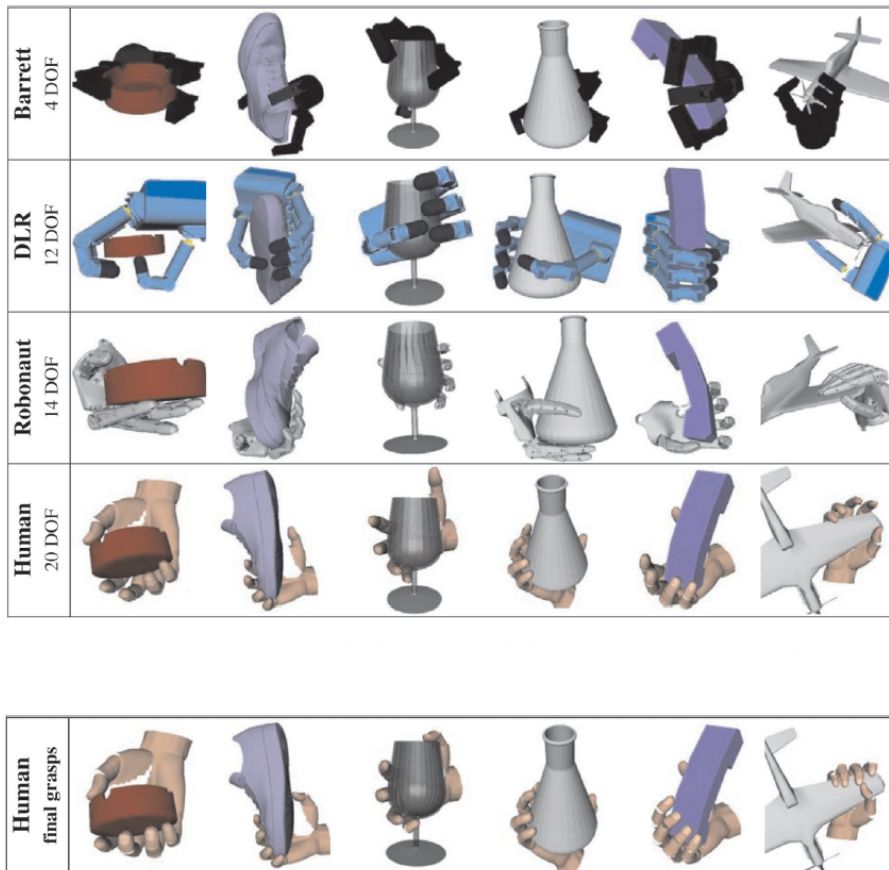
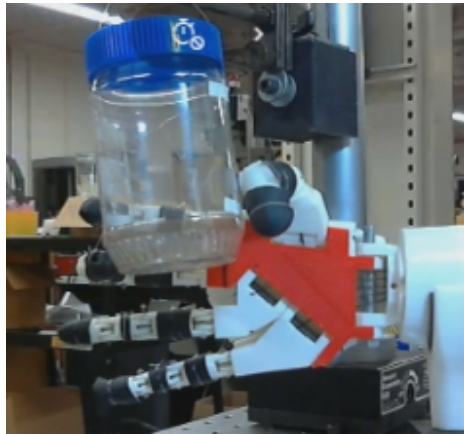
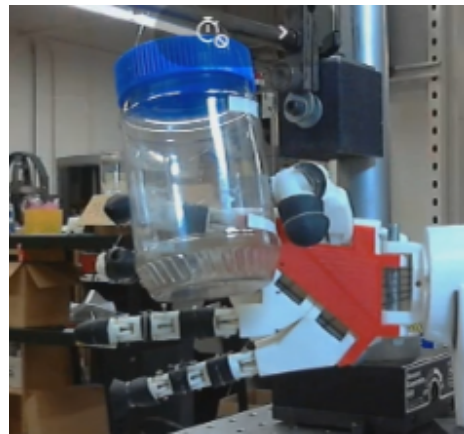


Figure 4: Different resulting hand postures depending on objects and hand chosen for study (program used is *GraspIt!*) [3].



(a) Height 1.0"



(b) Height 1.5"

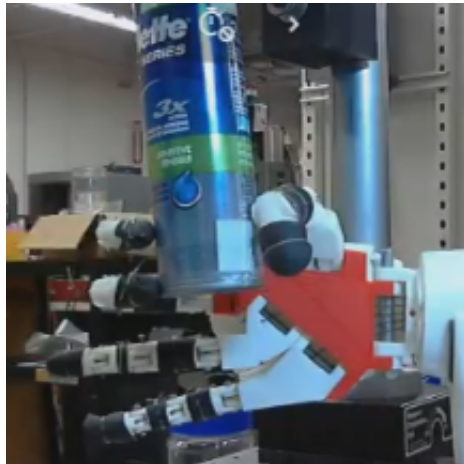


(c) Height 2.5"



(d) Height 3.5"

Figure 5: Final grasp positions of the fingers for object UC 1. The ANN was trained with an Cy-9/2600.



(a) Height 1.0"



(b) Height 1.5"

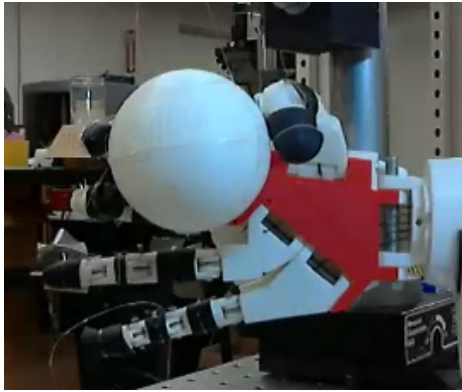


(c) Height 2.5"

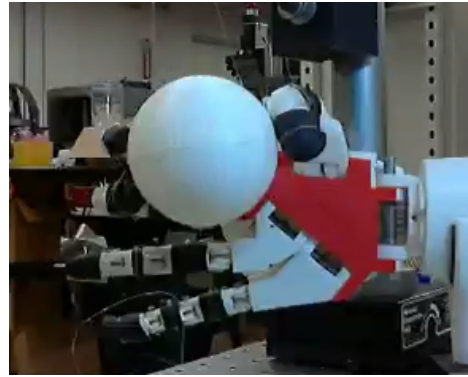


(d) Height 3.5"

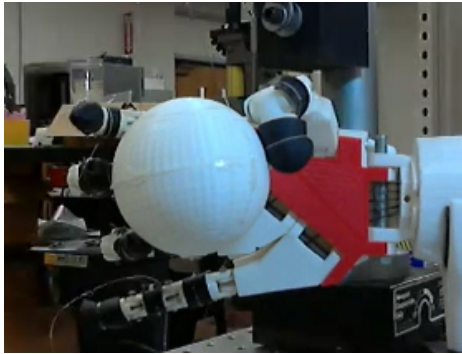
Figure 6: Final grasp positions of the fingers for object UC 2. The ANN was trained with Cy-9/2600.



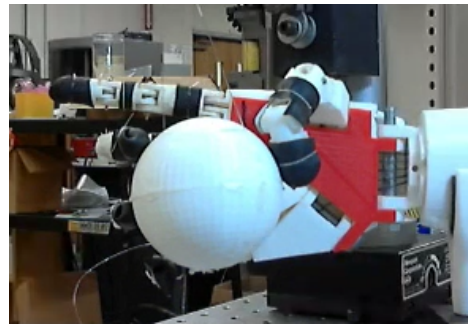
(a) Height 1.0"



(b) Height 1.5"



(c) Height 2.5"



(d) Height 3.5"

Figure 7: Final grasp positions of the fingers for object US1 with Sp-9/3600.

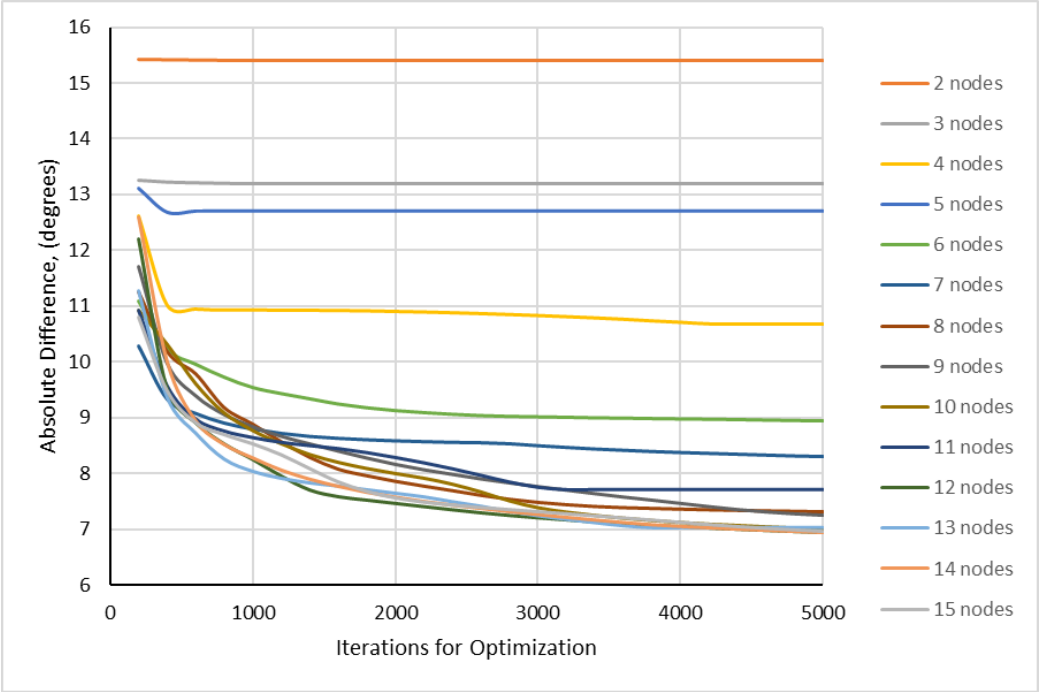


Figure 8: Absolute difference of the various ANN architectures. The ANN was trained with 100 data sets.

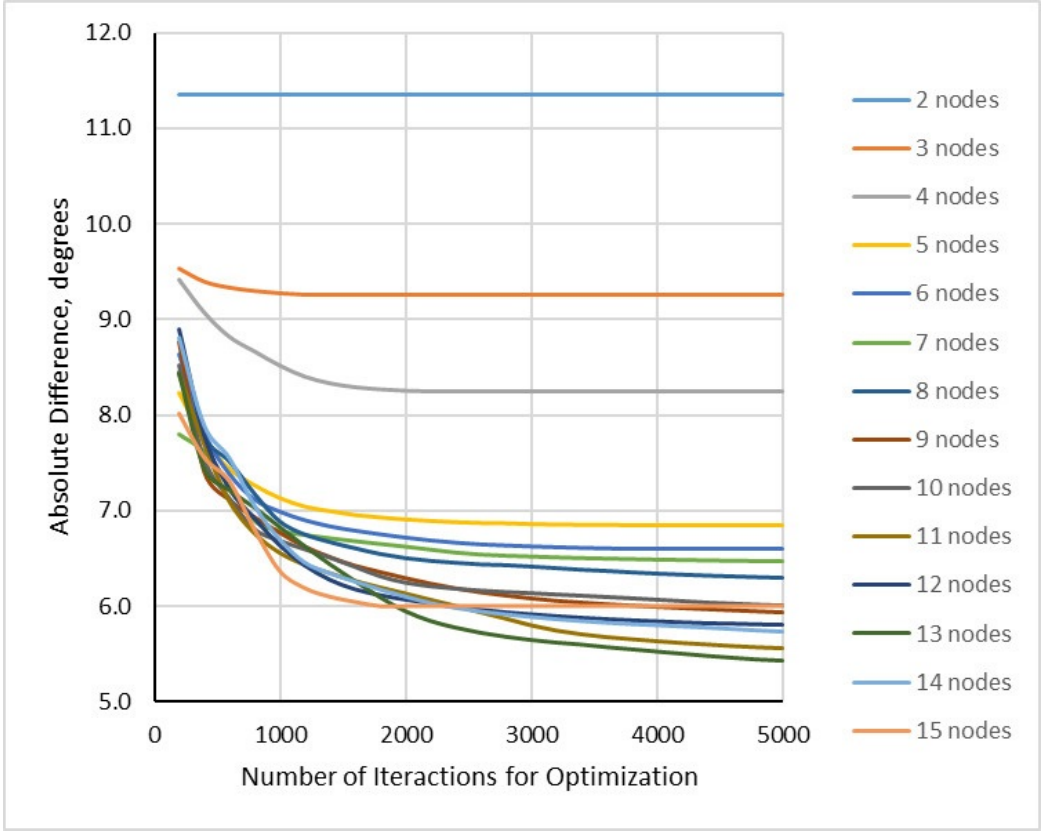


Figure 9: The average Absolute Difference across all data sets for the different architectures. These results were gathered from 76 training data sets.

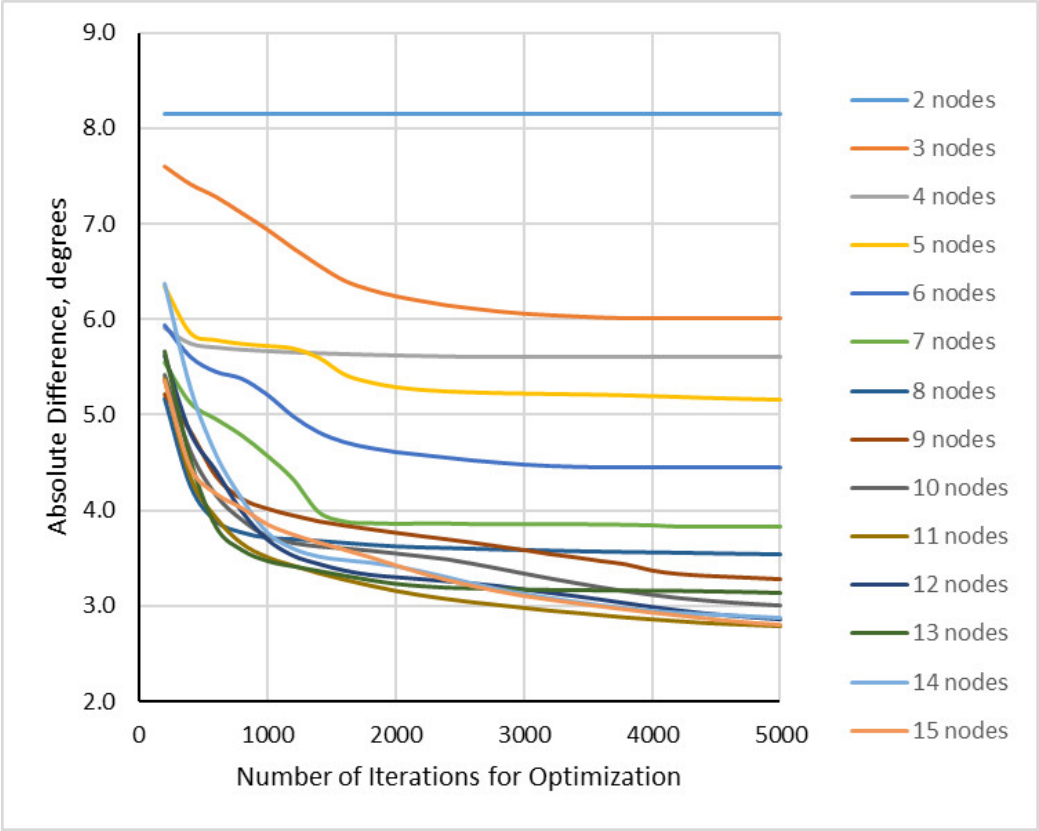


Figure 10: The average Absolute Difference across all data sets for the different architectures. These results were gathered from 24 training data sets.

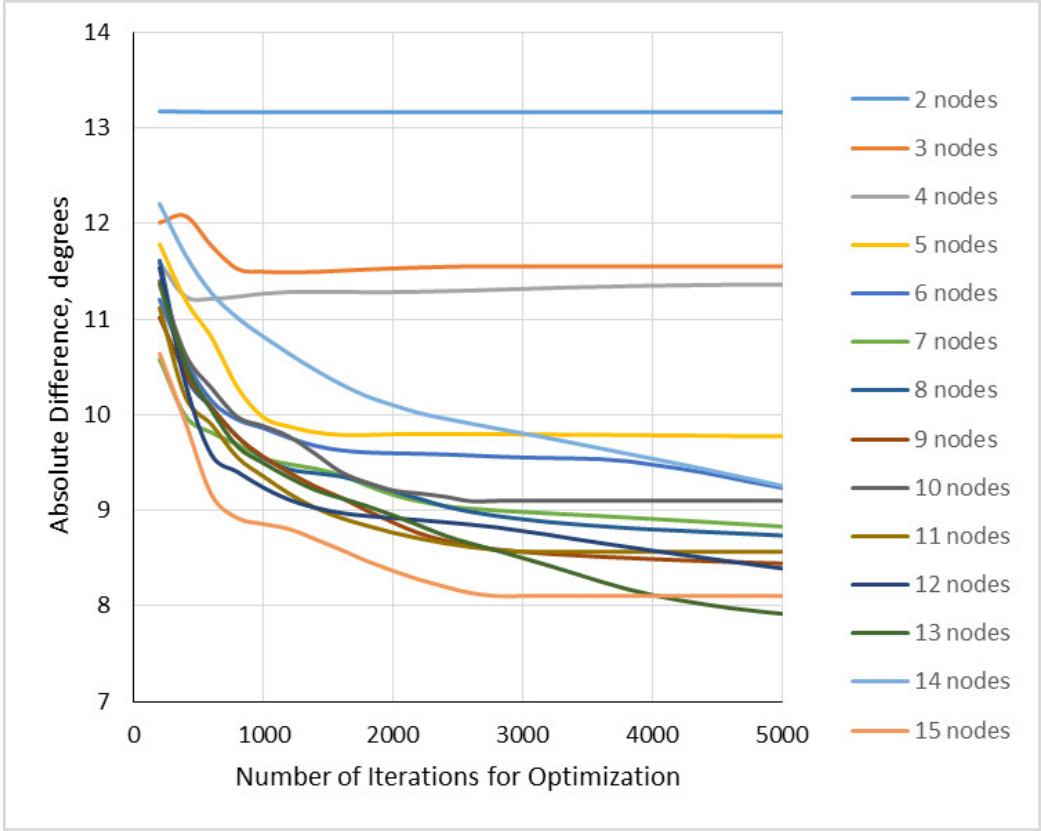


Figure 11: The average Absolute Difference of each ANN architecture with 108 training sets. This set incorporated both spheres and cylinders for training.

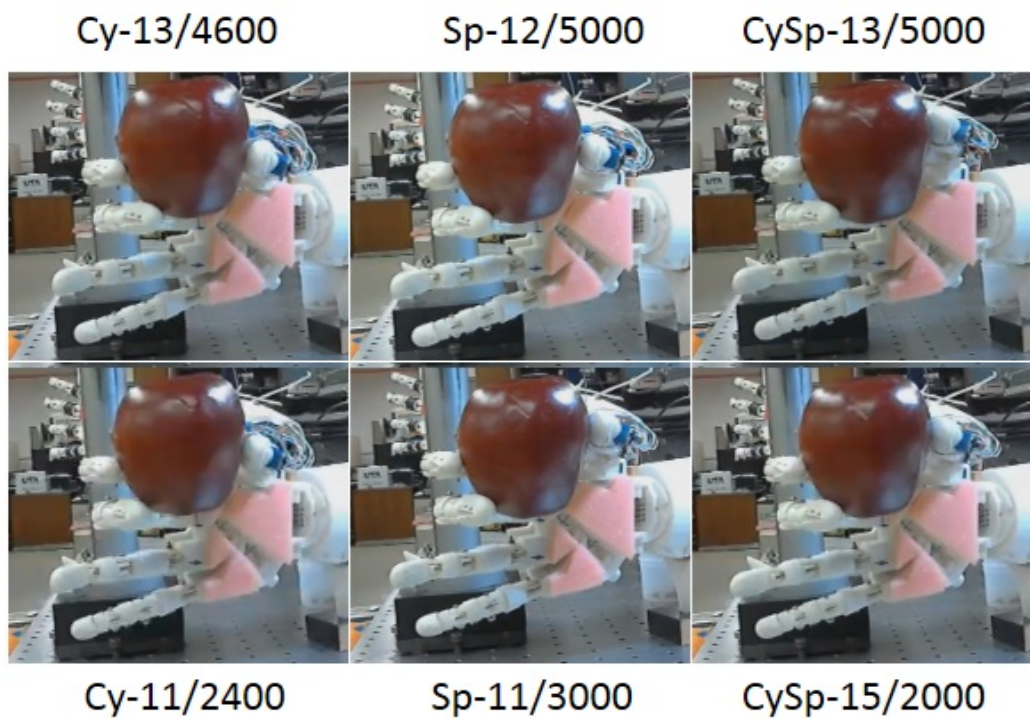


Figure 12: Comparison of grasps of objects OU2 at height 1.0", with all 6 ANN architectures.

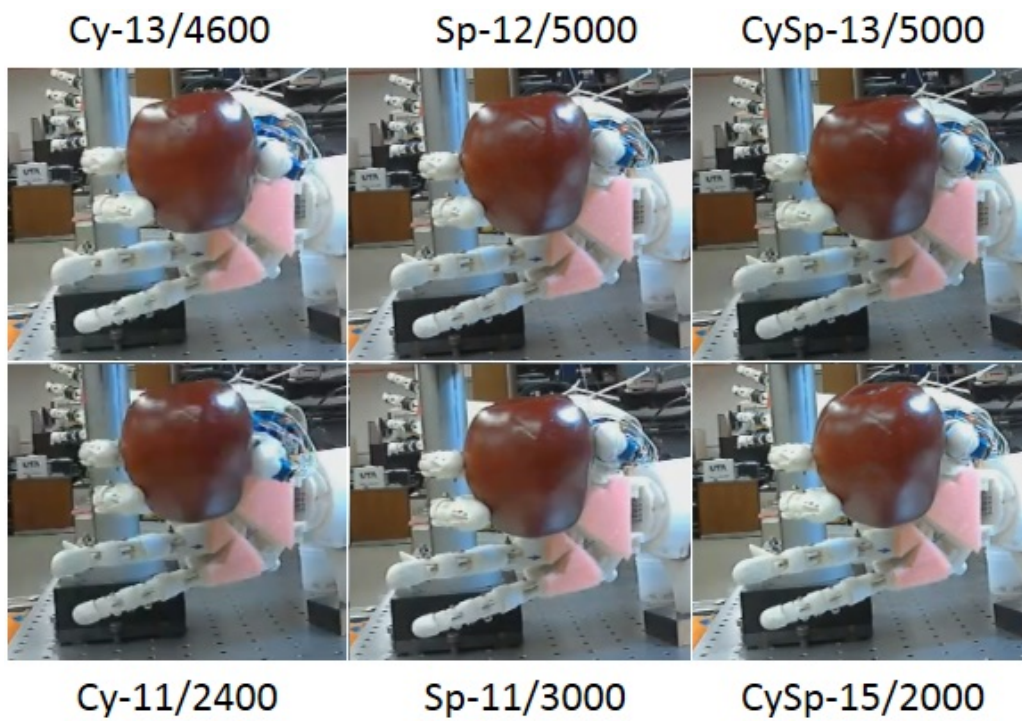


Figure 13: Comparison of grasps of objects OU2 at height 1.0", with all 6 ANN architectures.

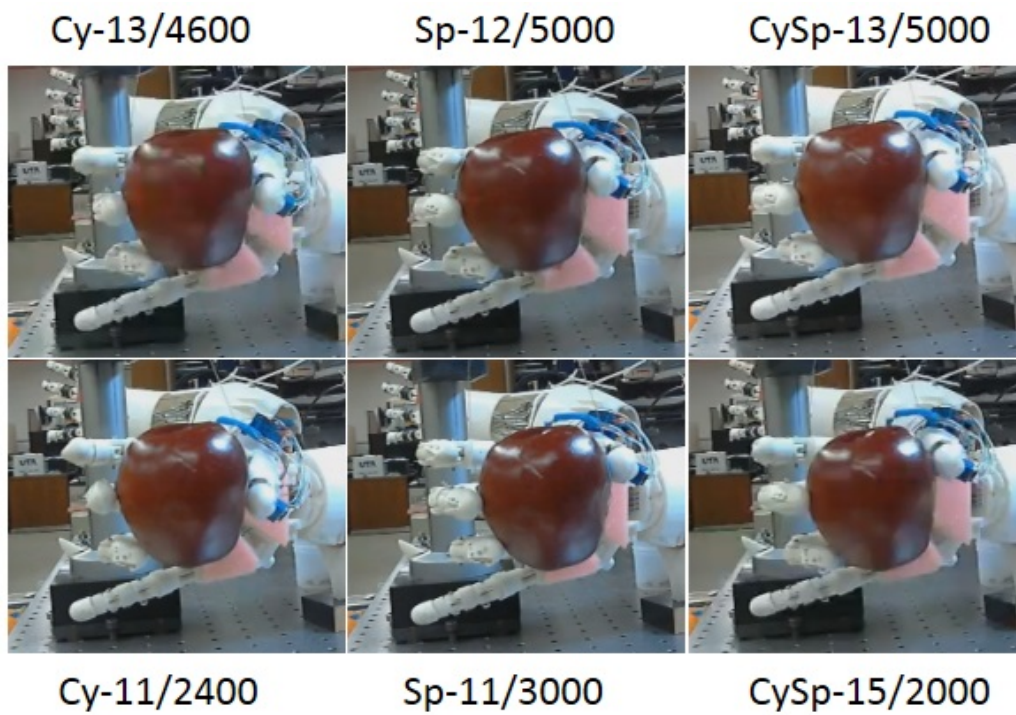


Figure 14: Comparison of grasps of objects OU2 at height 1.0", with all 6 ANN architectures.

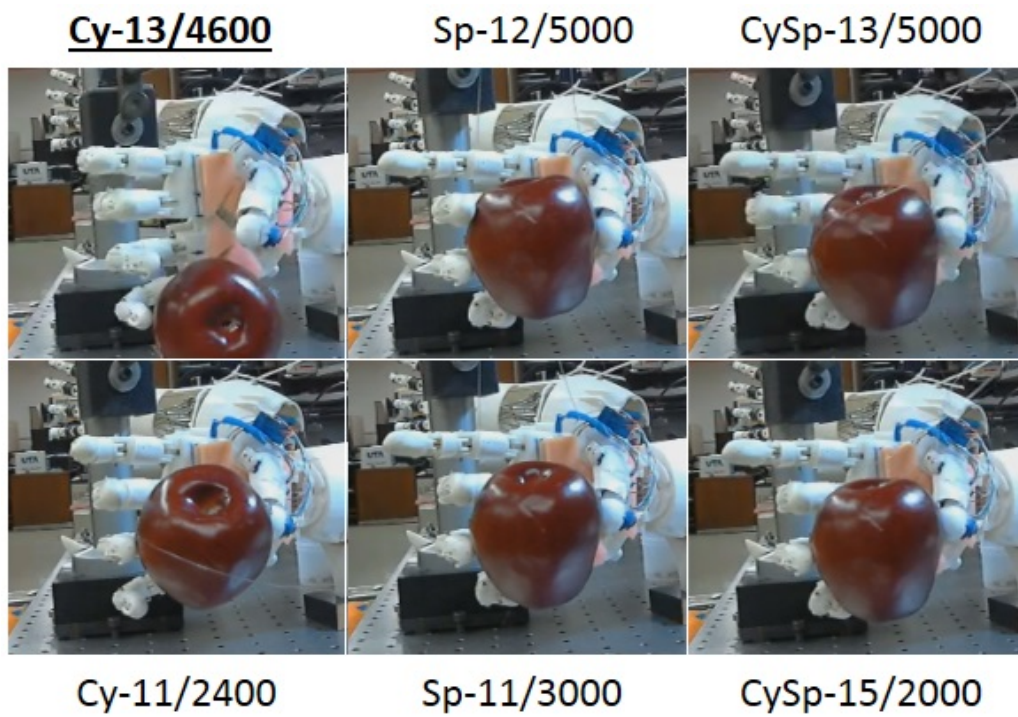


Figure 15: Comparison of grasps of objects OU2 at height 1.0", with all 6 ANN architectures.

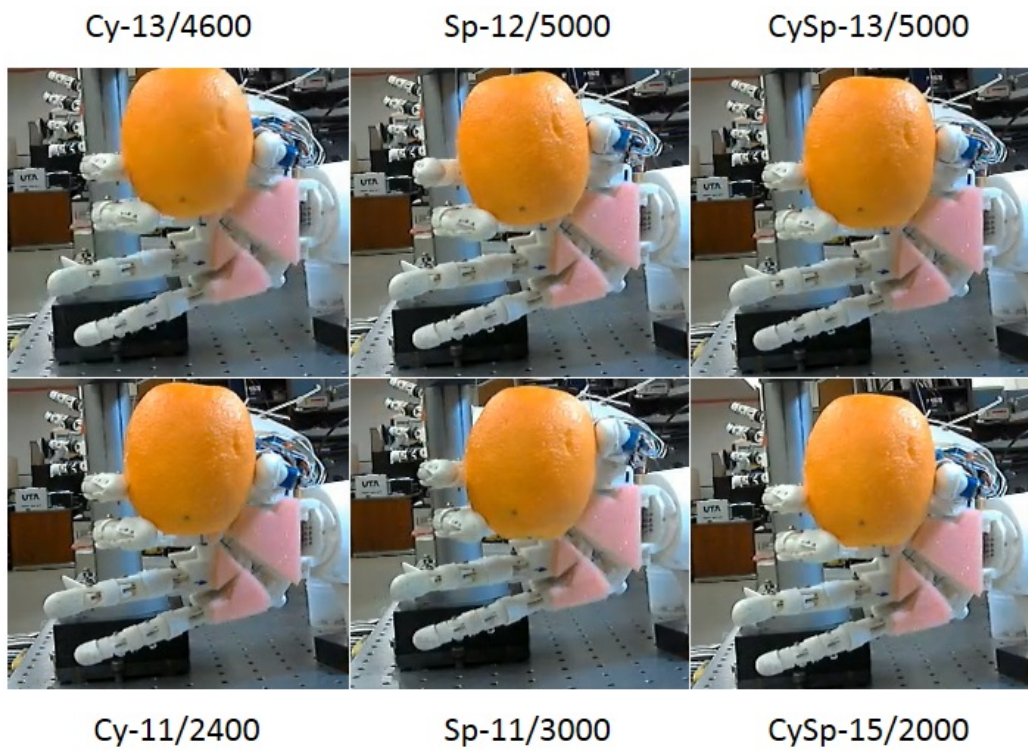


Figure 16: Comparison of grasps of objects OU2 at height 1.0", with all 6 ANN architectures.

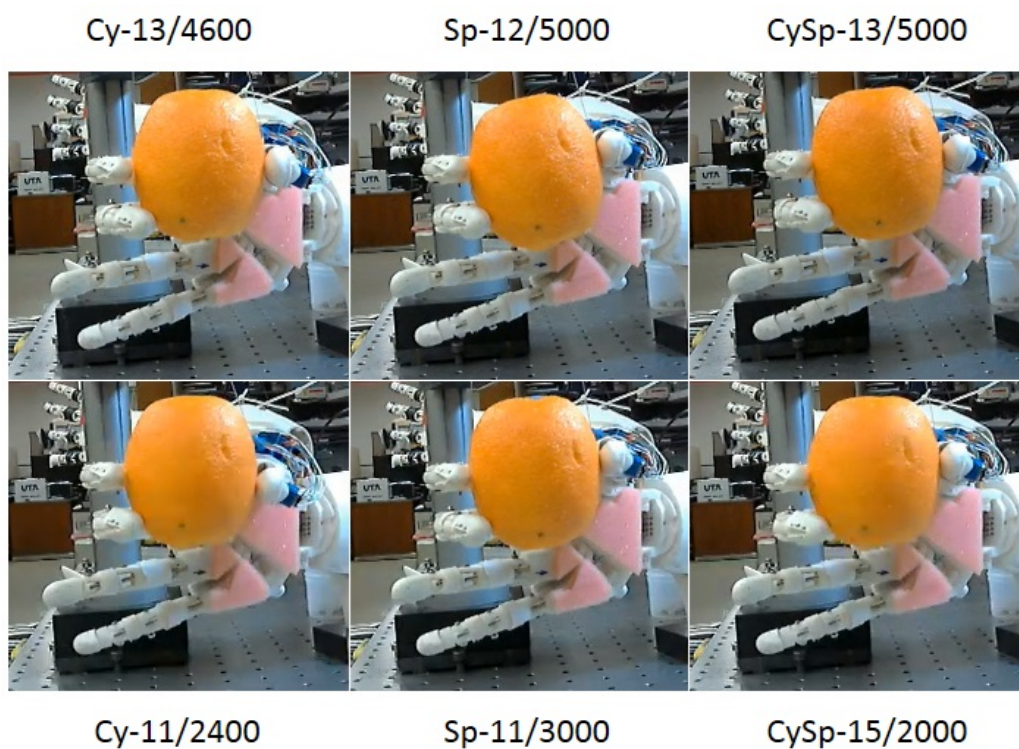


Figure 17: Comparison of grasps of objects OU2 at height 1.5", with all 6 ANN architectures.

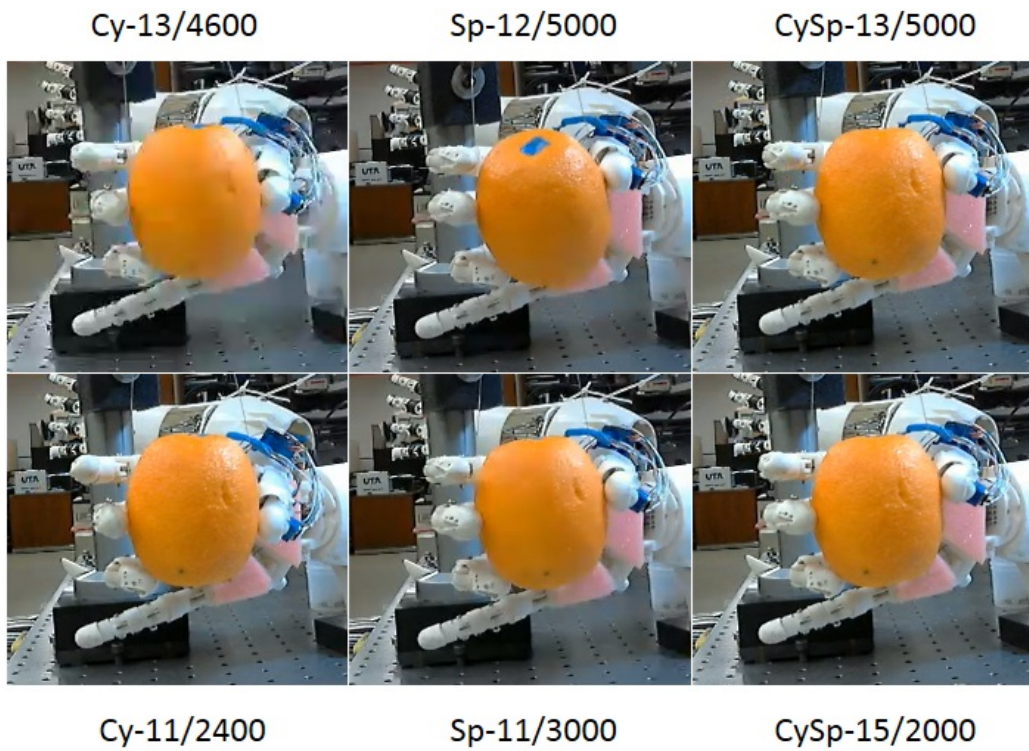


Figure 18: Comparison of grasps of objects OU2 at height 1.5", with all 6 ANN architectures.

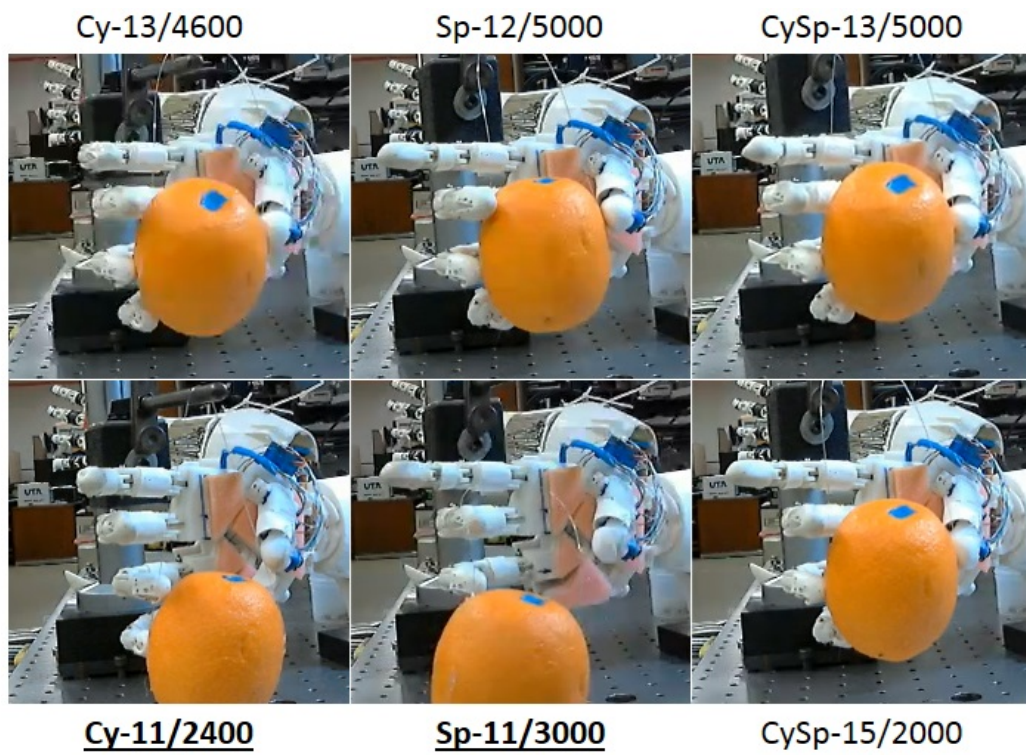


Figure 19: Comparison of grasps of objects OU2 at height 1.5", with all 6 ANN architectures.

Table 2: ANN predictions by various architectures on unidentified object UC1. The values in bold indicate when the finger has moved however, no contact was made with the object.

Height	Architecture	θ_I	θ_M	θ_P	θ_P	θ_{T1}	θ_{T2}	θ_{T3}	θ_{T4}
1	Cy-11/2400	42	69	0	0	109	144	0	0
	Cy-13/4600	43	72	0	0	108	146	0	0
	CySp-15/2000	36	63	0	0	110	143	24	0
	CySp-13/5000	40	69	0	0	101	143	26	0
	Cy-11/2400-CL	52	84	0	0	120	140	30	2
	Cy-13/4600-CL	55	89	0	0	115	139	30	2
	CySp-15/2000-CL	46	80	0	0	113	142	29	3
	CySp-13/5000-CL	50	88	0	0	110	141	31	0
1.5	Cy-11/2400	42	70	0	0	120	140	0	0
	Cy-13/4600	45	73	0	0	115	139	0	0
	CySp-15/2000	40	63	0	0	113	142	24	0
	CySp-13/5000	42	62	0	0	110	141	26	0
	Cy-11/2400-CL	63	95	0	0	120	140	0	0
	Cy-13/4600-CL	59	96	0	0	115	154	34	10
	CySp-15/2000-CL	62	97	0	0	112	153	35	9
	CySp-13/5000-CL	64	97	0	0	113	153	33	13
2.5	Cy-11/2400	33	59	41	0	129	140	0	0
	Cy-13/4600	39	61	42	0	125	139	0	0
	CySp-15/2000	40	67	40	0	124	142	24	0
	CySp-13/5000	42	65	42	0	120	141	26	0
	Cy-11/2400-CL	49	82	42	0	129	139	34	8
	Cy-13/4600-CL	53	94	56	0	125	139	35	10
	CySp-15/2000-CL	43	81	40	0	124	142	30	3
	CySp-13/5000-CL	53	91	54	0	120	141	34	1
3.5	Cy-11/2400	34	57	41	46	136	133	0	0
	Cy-13/4600	40	59	40	48	135	134	0	0
	CySp-15/2000	35	67	37	45	142	137	24	0
	CySp-13/5000	44	77	45	45	135	135	26	0
	Cy-11/2400-CL	50	89	48	63	137	133	45	13
	Cy-13/4600-CL	52	89	47	71	135	134	45	17
	CySp-15/2000-CL	57	94	51	70	142	137	32	7
	CySp-13/5000-CL	54	94	54	72	135	135	40	6

Table 3: ANN predictions by various architectures on unidentified object UC2. The values in bold indicate when the finger has moved however, no contact was made with the object.

Height	Architecture	θ_I	θ_M	θ_P	θ_P	θ_{T1}	θ_{T2}	θ_{T3}	θ_{T4}
1	Cy-11/2400	51	86	0	0	109	151	0	0
	Cy-13/4600	48	83	0	0	110	152	0	0
	CySp-15/2000	50	81	0	0	102	148	24	0
	CySp-13/5000	51	91	0	0	110	151	26	0
	Cy-11/2400-CL	62	94	0	0	109	151	33	6
	Cy-13/4600-CL	63	103	0	0	110	152	32	4
	CySp-15/2000-CL	75	115	0	0	102	148	40	10
	CySp-13/5000-CL	73	116	0	0	110	151	27	8
1.5	Cy-11/2400	52	88	0	0	119	148	0	0
	Cy-13/4600	50	85	0	0	120	147	0	0
	CySp-15/2000	40	63	0	0	113	142	24	0
	CySp-13/5000	52	84	0	0	115	149	26	0
	Cy-11/2400-CL	68	109	0	0	119	148	34	9
	Cy-13/4600-CL	73	109	0	0	120	147	34	9
	CySp-15/2000-CL	71	110	0	0	106	147	43	9
	CySp-13/5000-CL	73	110	0	0	115	149	31	10
2.5	Cy-11/2400	46	81	53	0	129	151	0	0
	Cy-13/4600	47	80	51	0	130	151	0	0
	CySp-15/2000	53	79	51	0	118	149	24	0
	CySp-13/5000	49	79	46	0	119	150	26	0
	Cy-11/2400-CL	74	114	70	0	129	151	34	13
	Cy-13/4600-CL	73	113	69	0	130	151	34	13
	CySp-15/2000-CL	71	115	67	0	118	149	44	9
	CySp-13/5000-CL	73	116	72	0	119	150	35	12
3.5	Cy-11/2400	55	88	52	52	135	146	5	0
	Cy-13/4600	49	88	51	51	138	146	5	0
	CySp-15/2000	53	78	54	50	134	147	24	0
	CySp-13/5000	53	84	52	49	132	145	26	0
	Cy-11/2400-CL	74	116	71	80	135	146	40	11
	Cy-13/4600-CL	77	118	73	80	138	146	40	10
	CySp-15/2000-CL	74	119	72	80	134	147	46	16
	CySp-13/5000-CL	77	118	75	80	132	145	42	17

Table 4: ANN predictions by various architectures on unidentified object US1. The values in bold indicate when the finger has moved however, no contact was made with the object.

Height	Architecture	θ_I	θ_M	θ_P	θ_P	θ_{T1}	θ_{T2}	θ_{T3}	θ_{T4}
1	Sp-11/3000	37	65	0	0	101	146	28	0
	Sp-12/5000	35	69	0	0	101	149	28	1
	CySp-15/2000	41	69	0	0	101	145	24	0
	CySp-13/5000	38	69	0	0	97	146	26	0
	Sp-11/2000-CL	55	110	0	0	102	147	25	0
	Sp-12/5000-CL	50	102	0	0	101	149	26	0
	CySp-15/2000-CL	51	106	0	0	102	146	26	0
	CySp-13/5000-CL	51	97	0	0	97	147	26	0
1.5	Sp-11/3000	41	69	0	0	109	147	28	0
	Sp-12/5000	41	65	0	0	108	144	28	1
	CySp-15/2000	42	70	0	0	104	145	24	0
	CySp-13/5000	42	69	0	0	110	145	26	0
	Sp-11/2000-CL	53	98	0	0	109	148	24	3
	Sp-12/5000-CL	49	93	0	0	108	144	24	3
	CySp-15/2000-CL	54	91	0	0	105	145	28	1
	CySp-13/5000-CL	46	80	0	0	110	145	28	1
2.5	Sp-11/3000	37	65	45	0	121	142	28	0
	Sp-12/5000	38	63	45	0	120	143	28	1
	CySp-15/2000	30	67	43	0	117	142	24	0
	CySp-13/5000	30	65	39	0	126	143	26	0
	Sp-11/2000-CL	57	89	61	0	121	143	39	4
	Sp-12/5000-CL	61	92	62	0	121	143	40	3
	CySp-15/2000-CL	59	94	64	0	117	142	35	5
	CySp-13/5000-CL	66	91	49	0	125	144	32	6
3.5	Sp-11/3000	0	54	36	57	139	133	28	0
	Sp-12/5000	0	60	38	55	140	134	28	1
	CySp-15/2000	0	54	39	53	137	131	24	0
	CySp-13/5000	0	50	37	51	142	131	26	0
	Sp-11/2000-CL	0	96	44	61	139	133	47	19
	Sp-12/5000-CL	0	93	44	78	140	134	44	20
	CySp-15/2000-CL	0	99	52	80	137	131	49	15
	CySp-13/5000-CL	0	94	46	80	142	131	46	17

REFERENCES

- [1] U. H. Shah, “Human robot interaction using knowledge base approach,” Ph.D. dissertation, 2016.
- [2] V. J. Santos and F. J. Valero-Cuevas, “Stochastic analysis of anatomical data suggests three characteristic types of thumb kinematics,” in *Summer Bioengineering Conference*, 2003, pp. 25–29.
- [3] M. T. Ciocarlie and P. K. Allen, “Hand posture subspaces for dexterous robotic grasping,” *The International Journal of Robotics Research*, vol. 28, no. 7, pp. 851–867, 2009.
- [4] N. Hockstein, C. Gourin, R. Faust, and D. J. Terris, “A history of robots: from science fiction to surgical robotics,” *Journal of robotic surgery*, vol. 1, no. 2, pp. 113–118, 2007.
- [5] N. Mavridis, “A review of verbal and non-verbal human–robot interactive communication,” *Robotics and Autonomous Systems*, vol. 63, pp. 22–35, 2015.
- [6] R. Kumar, P. Berkelman, P. Gupta, A. Barnes, P. S. Jensen, L. L. Whitcomb, and R. H. Taylor, “Preliminary experiments in cooperative human/robot force control for robot assisted microsurgical manipulation,” in *Robotics and Automation, 2000. Proceedings. ICRA’00. IEEE International Conference on*, vol. 1. IEEE, 2000, pp. 610–617.
- [7] D. B. Camarillo, T. M. Krummel, and J. K. Salisbury, “Robotic technology in surgery: past, present, and future,” *The American Journal of Surgery*, vol. 188, no. 4, pp. 2–15, 2004.

- [8] R. H. Taylor, A. Menciassi, G. Fichtinger, and P. Dario, “Medical robotics and computer-integrated surgery,” in *Springer handbook of robotics*. Springer, 2008, pp. 1199–1222.
- [9] J. Zlotowski, D. Proudfoot, K. Yogeewaran, and C. Bartneck, “Anthropomorphism: opportunities and challenges in human–robot interaction,” *International Journal of Social Robotics*, vol. 7, no. 3, pp. 347–360, 2015.
- [10] M. M. A. de Graaf and S. B. Allouch, “The relation between people’s attitude and anxiety towards robots in human-robot interaction,” in *2013 IEEE RO-MAN*, Aug 2013, pp. 632–637.
- [11] E. Mattar, “A survey of bio-inspired robotics hands implementation: New directions in dexterous manipulation,” *Robotics and Autonomous Systems*, vol. 61, no. 5, pp. 517–544, 2013.
- [12] Z. Li, P. Hsu, and S. Sastry, “Grasping and coordinated manipulation by a multifingered robot hand,” *The International Journal of Robotics Research*, vol. 8, no. 4, pp. 33–50, 1989.
- [13] C. R. Mason, J. E. Gomez, and T. J. Ebner, “Hand synergies during reach-to-grasp,” *Journal of Neurophysiology*, vol. 86, no. 6, pp. 2896–2910, 2001.
- [14] B. Huang, S. El-Khoury, M. Li, J. J. Bryson, and A. Billard, “Learning a real time grasping strategy,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 593–600.
- [15] M. Santello, M. Flanders, and J. F. Soechting, “Patterns of hand motion during grasping and the influence of sensory guidance,” *Journal of Neuroscience*, vol. 22, no. 4, pp. 1426–1435, 2002.
- [16] R. Dillmann, O. Rogalla, M. Ehrenmann, R. Zollner, and M. Bordegoni, “Learning robot behaviour and skills based on human demonstration and advice: the

- machine learning paradigm,” in *ROBOTICS RESEARCH-INTERNATIONAL SYMPOSIUM-*, vol. 9, 2000, pp. 229–238.
- [17] R. Zollner, O. Rogalla, R. Dillmann, and M. Zollner, “Understanding users intention: programming fine manipulation tasks by demonstration,” in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 2. IEEE, 2002, pp. 1114–1119.
- [18] R. Zollner, T. Asfour, and R. Dillmann, “Programming by demonstration: Dual-arm manipulation tasks for humanoid robots,” in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 1. IEEE, 2004, pp. 479–484.
- [19] H. Dang, J. Weisz, and P. K. Allen, “Blind grasping: Stable robotic grasping using tactile feedback and hand kinematics,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5917–5922.
- [20] C. Ferrari and J. Canny, “Planning optimal grasps,” in *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*. IEEE, 1992, pp. 2290–2295.
- [21] J. Laaksonen, V. Kyrki, and D. Kragic, “Evaluation of feature representation and machine learning methods in grasp stability learning,” in *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*. IEEE, 2010, pp. 112–117.
- [22] A. Miller, P. Allen, V. Santos, and F. Valero-Cuevas, “From robotic hands to human hands: a visualization and simulation engine for grasping research,” *Industrial Robot: An International Journal*, vol. 32, no. 1, pp. 55–63, 2005.
- [23] Y. Bekiroglu, J. Laaksonen, J. A. Jorgensen, V. Kyrki, and D. Kragic, “Assessing grasp stability based on learning and haptic data,” *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 616–629, 2011.

- [24] K. Bernardin, K. Ogawara, K. Ikeuchi, and R. Dillmann, "A sensor fusion approach for recognizing continuous human grasping sequences using hidden markov models," *IEEE Transactions on Robotics*, vol. 21, no. 1, pp. 47–57, 2005.
- [25] M. C. Carrozza, G. Cappiello, S. Micera, B. B. Edin, L. Beccai, and C. Cipriani, "Design of a cybernetic hand for perception and action," *Biological cybernetics*, vol. 95, no. 6, pp. 629–644, 2006.
- [26] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 102, no. 127, pp. 126–133, 1981.
- [27] H. Zhang and R. Paul, "Hybrid control of robot manipulators," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2. IEEE, 1985, pp. 602–607.
- [28] O. Khatib and J. Burdick, "Motion and force control of robot manipulators," in *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, vol. 3. IEEE, 1986, pp. 1381–1386.
- [29] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.
- [30] M. Zecca, G. Cappiello, F. Sebastiani, S. Roccella, F. Vecchi, M. Carrozza, and P. Dario, "Experimental analysis of the proprioceptive and exteroceptive sensors of an underactuated prosthetic hand," in *Proceeding of the ICORR 2003 (The Eighth International Conference on Rehabilitation Robotics)*. Citeseer, 2003.
- [31] S. Roccella, M. C. Carrozza, G. Cappiello, P. Dario, J.-J. Cabibihan, M. Zecca, H. Miwa, K. Itoh, and M. Marsumoto, "Design, fabrication and preliminary results of a novel anthropomorphic hand for humanoid robotics: Rch-1," in *In-*

- telligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 1. IEEE, 2004, pp. 266–271.
- [32] M. Li, K. Hang, D. Kragic, and A. Billard, “Dexterous grasping under shape uncertainty,” *Robotics and Autonomous Systems*, vol. 75, pp. 352–364, 2016.
- [33] L.-M. Fok and J. Wang, “Two recurrent neural networks for grasping force optimization of multi-fingered robotic hands,” in *Neural Networks, 2002. IJCNN’02. Proceedings of the 2002 International Joint Conference on*, vol. 1. IEEE, 2002, pp. 35–40.
- [34] Y. Xia, J. Wang, and L.-M. Fok, “Grasping-force optimization for multifingered robotic hands using a recurrent neural network,” *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 549–554, 2004.
- [35] Q. V. Le, D. Kamm, A. F. Kara, and A. Y. Ng, “Learning to grasp objects with multiple contact points,” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 5062–5069.
- [36] S. Levine, C. Finn, T. Darrell, and P. Abbeel, “End-to-end training of deep visuomotor policies,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.
- [37] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, “Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics,” *arXiv preprint arXiv:1703.09312*, 2017.
- [38] J. Mahler and K. Goldberg, “Learning deep policies for robot bin picking by simulating robust grasping sequences,” in *Conference on Robot Learning*, 2017, pp. 515–524.
- [39] S. M. Nancy, M. A. Tawfik, and I. A. Baqer, “A novel approach to control the robotic hand grasping process by using an artificial neural network algorithm,” *Journal of Intelligent Systems*, vol. 26, no. 2, pp. 215–231, 2017.

- [40] C. Yang, Y. Jiang, Z. Li, W. He, and C.-Y. Su, “Neural control of bimanual robots with guaranteed global stability and motion precision,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1162–1171, 2017.
- [41] P.-C. Yang, K. Sasaki, K. Suzuki, K. Kase, S. Sugano, and T. Ogata, “Repeatable folding task by humanoid robot worker using deep learning,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 397–403, 2017.
- [42] G. Ghazaei, A. Alameer, P. Degenaar, G. Morgan, and K. Nazarpour, “Deep learning-based artificial vision for grasp classification in myoelectric hands,” *Journal of neural engineering*, vol. 14, no. 3, p. 036025, 2017.
- [43] K. Fang, Y. Bai, S. Hinterstoisser, and M. Kalakrishnan, “Multi-task domain adaptation for deep learning of instance grasping from simulation,” *arXiv preprint arXiv:1710.06422*, 2017.
- [44] S. Gu, E. Holly, T. Lillicrap, and S. Levine, “Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates,” in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 3389–3396.
- [45] I. Lenz, H. Lee, and A. Saxena, “Deep learning for detecting robotic grasps,” *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.
- [46] J. Redmon and A. Angelova, “Real-time grasp detection using convolutional neural networks,” in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1316–1322.
- [47] S. Kumra and C. Kanan, “Robotic grasp detection using deep convolutional neural networks,” in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 769–776.

- [48] M. Kerzel and S. Wermter, “Neural end-to-end self-learning of visuomotor skills by environment interaction,” in *International Conference on Artificial Neural Networks*. Springer, 2017, pp. 27–34.
- [49] D. Guo, T. Kong, F. Sun, and H. Liu, “Object discovery and grasp detection with a shared convolutional neural network,” in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2038–2043.
- [50] G. Cotugno, K. Althoefer, and T. Nanayakkara, “The role of the thumb: study of finger motion in grasping and reachability space in human and robotic hands,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1061–1070, 2017.
- [51] V. K. Nanayakkara, G. Cotugno, N. Vitzilaios, D. Venetsanos, T. Nanayakkara, and M. N. Sahinkaya, “The role of morphology of the thumb in anthropomorphic grasping: a review,” *Frontiers in Mechanical Engineering*, vol. 3, p. 5, 2017.
- [52] M. Regoli, N. Jamali, G. Metta, and L. Natale, “Controlled tactile exploration and haptic object recognition,” in *Advanced Robotics (ICAR), 2017 18th International Conference on*. IEEE, 2017, pp. 47–54.
- [53] P. Vicente, L. Jamone, and A. Bernardino, “Towards markerless visual servoing of grasping tasks for humanoid robots,” in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 3811–3816.
- [54] L. Y. Ku, E. Learned-Miller, and R. Grupen, “Associating grasp configurations with hierarchical features in convolutional neural networks,” in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 2434–2441.
- [55] P.-C. Huang, L. Sentis, J. Lehman, C.-L. Fok, A. K. Mok, and R. Miikkulainen, “Tradeoffs in real-time robotic task design with neuroevolution learning for im-

- precise computation,” in *Real-Time Systems Symposium, 2015 IEEE*. IEEE, 2015, pp. 206–215.
- [56] F. Ficuciello, G. Palli, C. Melchiorri, and B. Siciliano, “Planning and control during reach to grasp using the three predominant ub hand iv postural synergies,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 2255–2260.
- [57] —, “Postural synergies of the ub hand iv for human-like grasping,” *Robotics and Autonomous Systems*, vol. 62, no. 4, pp. 515–527, 2014.
- [58] —, “Experimental evaluation of postural synergies during reach to grasp with the ub hand iv,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 1775–1780.
- [59] L. Villani, F. Ficuciello, V. Lippiello, G. Palli, F. Ruggiero, and B. Siciliano, “Grasping and control of multi-fingered hands,” in *Advanced Bimanual Manipulation*. Springer, 2012, pp. 219–266.
- [60] C. E. Ábrego, P. S. Shiakolas, and M. R. Sobhy, “Developing an educational and research human robot interaction environment for a mechanical finger/hand,” in *ASME 2015 International Mechanical Engineering Congress and Exposition*. American Society of Mechanical Engineers, 2015.
- [61] G. Langevin. (2016, October) open source 3d printed life size robot. [Online]. Available: <http://inmoov.fr/>
- [62] E. S. Olson, “Analysis and design of a two-wheeled robot with multiple user interface inputs and vision feedback control,” p. 100, 2010. [Online]. Available: <https://login.ezproxy.uta.edu/login?url=http://search.proquest.com/docview/753922467?accountid=7117>

- [63] M. R. Cutkosky, “On grasp choice, grasp models, and the design of hands for manufacturing tasks,” *IEEE Transactions on robotics and automation*, vol. 5, no. 3, pp. 269–279, 1989.
- [64] S. C. Jacobsen, J. E. Wood, D. Knutti, and K. B. Biggers, “The utah/mit dextrous hand: Work in progress,” *The International Journal of Robotics Research*, vol. 3, no. 4, pp. 21–50, 1984.
- [65] C. Loucks, V. Johnson, P. Boissiere, G. Starr, and J. Steele, “Modeling and control of the stanford/jpl hand,” in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4. IEEE, 1987, pp. 573–578.
- [66] G. A. Bekey, R. Tomovic, and I. Zeljkovic, “Control architecture for the belgrade/usc hand,” in *Dextrous robot hands*. Springer, 1990, pp. 136–149.
- [67] M. Fischer, P. van der Smagt, and G. Hirzinger, “Learning techniques in a data-glove based telemanipulation system for the dlr hand,” in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 2. IEEE, 1998, pp. 1603–1608.
- [68] H. Liu, J. Butterfass, S. Knoch, P. Meusel, and G. Hirzinger, “A new control strategy for dlr’s multisensory articulated hand,” *IEEE Control systems*, vol. 19, no. 2, pp. 47–54, 1999.
- [69] J. Butterfaß, M. Grebenstein, H. Liu, and G. Hirzinger, “Dlr-hand ii: Next generation of a dextrous robot hand,” in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 1. IEEE, 2001, pp. 109–114.
- [70] C. Borst, M. Fischer, S. Haidacher, H. Liu, and G. Hirzinger, “Dlr hand ii: experiments and experience with an anthropomorphic hand,” in *Robotics and Automation, 2003. Proceedings. ICRA’03. IEEE International Conference on*, vol. 1. IEEE, 2003, pp. 702–707.

- [71] C. Fantuzzi, C. Rossi, A. Tonielli, and G. Vassura, “A smart sensory and actuation system for the university of bologna robotic hand: Latest developments and implementations,” in *Proc. of Int. Symp. on Measurement and Control in Robotics*, 1992, pp. 515–522.
- [72] A. Eusebi, C. Fantuzzi, C. Melchiorri, M. Sandri, and A. Tonielli, “The ub hand ii control system: design features and experimental results,” in *Industrial Electronics, Control and Instrumentation, 1994. IECON'94., 20th International Conference on*, vol. 2. IEEE, 1994, pp. 782–787.
- [73] F. Lotti, P. Tiezzi, G. Vassura, L. Biagiotti, G. Palli, and C. Melchiorri, “Development of ub hand 3: Early results,” in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 4488–4493.
- [74] H. Kawasaki and T. Komatsu, “Mechanism design of anthropomorphic robot hand: Gifu hand,” *Journal of robotics and mechatronics*, vol. 11, no. 4, pp. 269–273, 1999.
- [75] H. Kawasaki, H. Shimomura, and Y. Shimizu, “Educational–industrial complex development of an anthropomorphic robot hand ‘gifu hand’,” *Advanced Robotics*, vol. 15, no. 3, pp. 357–363, 2001.
- [76] T. Mouri, H. Kawasaki, K. Yoshikawa, J. Takai, and S. Ito, “Anthropomorphic robot hand: Gifu hand iii,” in *Proc. Int. Conf. ICCAS*, 2002, pp. 1288–1293.
- [77] H. Kawasaki, T. Komatsu, and K. Uchiyama, “Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand ii,” *IEEE/ASME transactions on mechatronics*, vol. 7, no. 3, pp. 296–303, 2002.
- [78] M. C. Carrozza, P. Dario, F. Vecchi, S. Roccella, M. Zecca, and F. Sebastiani, “The cyberhand: on the design of a cybernetic prosthetic hand intended to be interfaced to the peripheral nervous system,” in *Intelligent Robots and Systems*,

- 2003.(IROS 2003). *Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 3. IEEE, 2003, pp. 2642–2647.
- [79] A. Bicchi, “Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity,” *IEEE Transactions on robotics and automation*, vol. 16, no. 6, pp. 652–662, 2000.
- [80] N. Wettels, A. R. Parnandi, J.-H. Moon, G. E. Loeb, and G. S. Sukhatme, “Grip control using biomimetic tactile sensing systems,” *IEEE/ASME Transactions On Mechatronics*, vol. 14, no. 6, pp. 718–723, 2009.
- [81] A. Cranny, D. P. Cotton, P. H. Chappell, S. Beeby, and N. White, “Thick-film force, slip and temperature sensors for a prosthetic hand,” *Measurement Science and Technology*, vol. 16, no. 4, p. 931, 2005.
- [82] D. J. Magee, *Orthopedic physical assessment*. Elsevier Health Sciences, 2014.
- [83] N. D. Mansfield, P.J., *Essentials of Kinesiology for the Physical Therapist Assistant*, 2010.
- [84] B. Kosko, *Neural networks and fuzzy systems: a dynamical systems approach to machine intelligence/book and disk*. Prentice Hall, Upper Saddle River, 1992.
- [85] E. Rich and K. Knight, *Artificial intelligence*. McGraw-Hill New York:, 1991, vol. 199, no. 1.
- [86] I. Steinwart and A. Christmann, *Support vector machines*. Springer Science & Business Media, 2008.
- [87] R. FUKUNAGA, *Statistical pattern recognition*. Academic Press., 1990.
- [88] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.
- [89] P. Ponce-Cruz and F. D. Ramírez-Figueroa, “Intelligent control for labview,” *Intelligent Control Systems with LabVIEW*, pp. 1–8, 2010.

- [90] J. C. John *et al.*, “Introduction to robotics: mechanics and control,” *Reading: Addison-Wesley*, 1989.
- [91] D. Kanoulas, J. Lee, D. G. Caldwell, and N. G. Tsagarakis, “Center-of-mass-based grasp pose adaptation using 3d range and force/torque sensing,” *arXiv preprint arXiv:1802.06392*, 2018.
- [92] G. Walck, R. Haschke, M. Meier, and H. J. Ritter, “Robot self-protection by virtual actuator fatigue: Application to tendon-driven dexterous hands during grasping,” in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 2200–2205.

BIOGRAPHICAL STATEMENT

Christopher Abrego graduated locally from Euless Trinity High School in 2007. That same year, he started his undergraduate education in mechanical engineering at the University of Texas (UT) at Arlington. During his junior year as an undergraduate, he found an interest in mathematics and began a second bachelors in mathematics. In December of 2012, Christopher graduated with a Bachelors degree in mechanical engineering and mathematics. A semester later, he joined the graduate program at UT Arlington in the department of mechanical engineering as a masters student. After two semesters, he joined the BS to Ph.D. program with a final aim to achieve a doctoral degree in mechanical engineering. In May of 2018, he graduated the doctoral program in mechanical engineering from UT Arlington.

Christopher performed all his research under the guidance of Professor Dr. Panos Shiakolas in the Manufacturing Automation and Robotic Systems (MARS) Laboratory. His main research interests are machine learning, Human-Robot Interaction (HRI), and Mechatronics. He was involved in a variety of research projects both in the lab and outside. Among one of the most distinguishable research endeavors has been the development of a HRI system for grasping with two design iterations of a biomimetic hand. The HRI platform has multiple modalities of interaction that users can interact with the biomimetic hand. Moreover, the software platform used for control is National Instruments LabVIEW, which provides a easily expandable platform for the integration of new interaction modalities.