

Methods and Applications for Controlling Biomimetic Robotic Hands

Matthew A. Paluszek

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
In
Mechanical Engineering

Shashank Priya, Chair
Andrew J. Kurdila
Alfred L. Wicks

November 21st, 2013
Blacksburg, Virginia

Keywords: humanoid robotics, data glove, brain computer interface, strain sensor, force sensor

Methods and Applications for Controlling Biomimetic Robotic Hands

Matthew A. Paluszek

ABSTRACT

Vast improvements in robotics and wireless communication have made teleoperated robots significantly more prevalent in industry, defense, and research. To help bridge the gap for these robots in the workplace, there has been a tremendous increase in research toward the development of biomimetic robotic hands that can simulate human operators. However, current methods of control are limited in scope and do not adequately represent human muscle memory and skills. The vision of this thesis is to provide a pathway for overcoming these limitations and open an opportunity for development and implementation of a cost effective methodology towards controlling a robotic hand.

The first chapter describes the experiments conducted using Flexpoint bend sensors in conjunction with a simple voltage divider to generate a cost-effective data glove that is significantly less expensive than the commercially available alternatives. The data glove was able to provide sensitivity of less than 0.1 degrees. The second chapter describes the molding process for embedding pressure sensors in silicone skin and data acquisition from them to control the robotic hand. The third chapter describes a method for parsing and observing the information from the data glove and translating the relevant control variables to the robotic hand. The fourth chapter focuses on the feasibility of the brain computer interfaces (BCI) and successfully demonstrates the implementation of a simple brain computer interface in controlling a robotic hand.

Acknowledgements

I would like to thank Dr. Shashank Priya, who not only gave me the opportunity to become a graduate student under him, but also pushed me to find accomplish something great as an engineer. I would also like to thank my Committee members, Dr. Alfred Wicks and Dr. Andrew Kurdila, who expanded my understanding of instrumentation and robotics in their respective classes.

I would also like to thank my colleagues with the Humanoid Robotics team: Michael Okyen, Steve Powell, Mike Stevens, and Eric Baldrighi, who were with me on late nights, through demos and presentations, and drove me to think of multiple methods and options to use to accomplish my goals. The same goes for my Senior Design students from 2011-2012: Mary Kyle Manson, Sarit Cliffer, Michael Osborne, and Thomas Howe, who were instrumental in designing and testing the early phases of the data glove as part of their design project.

The final chapter of this thesis would not have been possible if not for the assistance of Stephen LaConte and Read Montague, who graciously allowed me to use their very expensive and otherwise unusable fMRI machine to begin the start of BCI work with Virginia Tech.

I must also thank my parents, who supported me both financially as well as scholastically, as the advice and input of my father, an electrical engineer, was a huge benefit to my studies. From helping me troubleshoot circuits to MATLAB code, my father spent countless hours with me to help sort through and further my understanding of the engineering process and project planning.

Table of Contents

Chapter 1 : Data Glove Control	1
1.1 Need for Teleoperation	1
1.2 Current Control Options	1
1.3 Data Glove Technology	3
1.4 Sensing Methods.....	5
1.5 Sensor Analysis	10
1.6 Sensor Placement.....	11
1.7 Glove Selection.....	13
1.8 Testing the Data Glove	16
1.9 Results.....	21
1.10 Noted issues and Future Fixes.....	24
Chapter 2 : Robotic Hand Feedback	25
2.1 Need for Robotic Hand Feedback	25
2.2 Methods of Feedback	26
2.3 Piezoresistive Attempts	29
2.4 Molding Process.....	34
2.5 Sensor Validation	38
2.6 Results	43
2.7 Conclusions and further work	43
Chapter 3 : Robotic Hand Control	45
3.1 Method to Control The Robot	45
3.2 Deciding on a Controller.....	45
3.3 Reading Signals from the Glove	48

3.4 Parsing the Data Glove.....	49
3.5 Labview Code	52
3.6 Future Work	59
Chapter 4 : Brain Computer Interface Application.....	61
4.1 Beyond Data Glove Controllers	61
4.2 Brain Computer Interfaces	62
4.3 Non-Invasive BCI Methods	64
4.4 Invasive Methods	68
4.5 BCI Studies in this thesis	71
4.6 Experimental Setup	73
4.7 Experimental Process and Results	75
4.8 Future Applications of Research.....	81
Chapter 5 : Summary and Conclusions.....	84
Appendix A: Molding Process.....	86
Appendix B: Hand Control Code.....	90
References.....	95

List of Figures

Figure 1-1: The Dexterous Anthropomorphic Sensing Hybrid(DASH) hand, shown in comparison with an adult male hand.....	2
Figure 1-2: Sayre glove, with 10 degrees of freedom tracked using fiberoptic cables	4
Figure 1-3: Flexpoint bend sensors, in 1", 2", and 3" lengths.....	8
Figure 1-4: Voltage divider diagram. R_1 is the sensor, and R_2 is the static reference resistor.....	10
Figure 1-5: Circular grip(above) and power grip(below) are ideal grips for a number of teleoperative purposes.....	12
Figure 1-6: 3M™ Dual Lock™ Fastener (left) and close-up of 3M Dual Lock Fasteners(right).....	14
Figure 1-7: Data Glove Diagram(above) and physical Data glove(below), with sensors attached to dual lock tabs, but without wires.....	15
Figure 1-8: Tracking Points on the data glove.....	18
Figure 1-9: Two frames indicating the start of the switch being flipped	19
Figure 1-10: Voltage output before and after switch flipped.....	19
Figure 1-11: Data Glove with tracking points(above) and link segment model(below)	20
Figure 1-12: Proximal Angle vs. Voltage with linear fit(above), Distal Angle vs. Voltage with linear fit(below).	22
Figure 1-13: Data glove in the extended position(left) and at 25 degrees of flexion(right).....	23

Figure 2-1: Biotac robotic manipulator sensor.....	27
Figure 2-2: Takktile Sensor, with penny for size comparison.....	28
Figure 2-3: Schematic of fabric arrangement for pressure sensor.....	30
Figure 2-4: 6 inch fabric based sensor embedded in silicone.	32
Figure 2-5: Fabric based sensor surrounded in tape(above). Early attempts at embedding fabric based sensor in silicone(below).	33
Figure 2-6: Flexiforce sensor 4201(left) and sensor composition(left)	34
Figure 2-7: Left and right sides of the negative mold.	35
Figure 2-8: Positive Fingertip Mold.....	36
Figure 2-9: Fully molded embedded sensor attached to robotic fingertip.....	37
Figure 2-10: Theoretical response of 100 lbs sensor resistance in blue	38
Figure 2-11: Aluminum plate with bolts to hold sensor sample in place.	40
Figure 2-12: Experimental setup with 2.75 pounds of force applied with drill guide.....	41
Figure 2-13: Plot of sensor response due to weight in terms of resistance	41
Figure 2-14: Recommended circuit and governing equation from Tekscan	42
Figure 2-15: Inverting amplifier with linear fit.	43
Figure 3-1: Simple flow chart explaining the process of obtaining, parsing and using signals to control the robotic hand.....	48
Figure 3-2: Neutral Hand Position(top), distal finger flexion(middle), proximal finger flexion(bottom)	54
Figure 3-3: Thumb distal flexion(top), thumb proximal flexion(bottom).	55
Figure 4-1: Locations of various brain functions.....	63
Figure 4-2: Subject with high density EEG electrode array cap.....	65

Figure 4-3: A subject using a MEG interface.	67
Figure 4-4: Electrocorticography array attached to skull of a patient prior to surgery	69
Figure 4-5: American Sign Language A(left) and B(right) which are imitated by the robotic hand during different stages of brain activity.	75
Figure 4-6: A subject being prepared for fMRI readings	76
Figure 4-7: High-resolution image of the subject's brain.	77
Figure 4-8: Plot showing the training phase of the experiment.....	78
Figure 4-9: Image showing subject's brain, specifically blood flow during gripping action of the right hand.	79
Figure 4-10: Image showing the ideal response of the brain in red, and the actual response of the brain in black, which is seen trending with the ideal response.	80
Figure A- 1: Negative molds(left) with .75 scale and .5 scale positive mold.....	86
Figure A- 2: Positive mold with flexiforce sensor fitted into slot.	86
Figure A- 3: Left and right negative mold attached using guide holes.	87
Figure A- 4: Positive mold a sensor placed into silicone in full negative mold.....	87
Figure A- 5: Full mold assembly clamped to cure.....	88
Figure A- 6: Pulling the left and right mold apart.	88
Figure A- 7: Silicone mold with sensor and positive mold inside.....	89
Figure A- 8: Final silicone skin with only the sensor inside.....	89
Figure B- 1: Front Panel of the data glove controller.....	93
Figure B- 2: Labview panel with sliders visible.....	94

List of Tables

Table 1-1: Commercially available data gloves, with their respective sensing methods, degrees of freedom, sensitivity, and cost.	5
Table 3-1: Table of theoretical 12-bit values to be observed from the analog voltage ports.....	50
Table 3-2: Series of flexed and extended 12-bit values, observed bit values from the maple, translated values from governing equations, and values after stopper program.	57
Table 4-1: Table of advantages and disadvantages of various BCI methods.	71

Chapter 1 : Data Glove Control

1.1 Need for Teleoperation

There are numerous application domains that require human operators to regularly expose themselves to dangerous situations and environments in the line of duty, such as Explosive Ordnance Disposal(EOD) technicians, the agents tasked with containing the leakage during the Fukushima disaster, and even the astronauts. All of these tasks could ideally be performed by robots, and remove humans from harms way. However, current control schemes for robotic automation have limitations as they cannot match a human's expertise, and thus humans must still risk life and limb to accomplish these dangerous tasks. In recent years, teleoperated robots have increasingly become available, but still lack full implementation into the wider world due to prohibitively high costs. If these robotics and their controllers could be made affordable, there would be an increased demand for human controlled robotics that function at the same level as a human technician. This scenario of humans remotely controlling the robotic with the same degrees of freedom and range of motion as a human being will open many new possibilities.

1.2 Current Control Options

A human-like robotic manipulator provides an excellent compromise between strength and dexterity. However, the question that needs to be answered is “how to control the robotic manipulator?” In its ideal form, a human hand is comprised of 22 degrees of freedom, though this is often reduced using coupled motions to more simple versions such as the 16 degrees of freedom DASH hand. However, more complex

versions of robotic hands have been demonstrated, such as the Shadow Hand, which comprises 22 fully actuated degrees of freedom.



Figure 1-1: The Dexterous Anthropomorphic Sensing Hybrid(DASH) hand, shown in comparison with an adult male hand.

When selecting a robotic controller the key concern is user familiarity. For example, a computer mouse is a simple two degree of freedom controller that is commonly utilized. Only, in some cases it requires specialized training for new users, particularly children who have not trained those specific motor skills (Joiner, Messer, Light, & Littleton, 1998). However, even with certain non-traditional mice that are used for gaming or other technical purposes, the maximum degrees of freedom that can be operated by one hand would be limited. Another controller option is a multi-axis mouse or joystick, which has become popular for Computer Assisted Drafting (CAD) programs. These controllers allow for an object to be rotated, panned, zoomed, and tilted in 3-D space using a single hand. These tools require some training, as they are not as

familiar to the average user as a simple mouse, though they allow for more complex control of a system.

While these tools have suitable features desired in a variety of applications, they fail to take advantage of one of the key features of a human-like manipulator and that is familiarity. Using a human-like hand for robotic teleoperation not only provides the capability of using many human tools that are essential but it also allows for a very clear association from the user to operate said manipulator as the user can use the manipulator to effectively mimic their own methods of using tools. As such, it would be ideal to give the user a controller that capitalizes on this inherent familiarity with a user's own hand to apply to the robotic manipulator.

1.3 Data Glove Technology

A human-like user control interface could be developed using a data glove. A data glove, which is also called a wired glove or sensor glove, is a fabric glove worn by a human user that utilizes sensors to detect the positions of the user's digits, and translate them to an external program to be further used to control or record hand positions. Using a data glove, it would be possible for a human user to effectively control a biomimetic robotic manipulator by directly mimicking their own motions, creating a seamless meshing of human teleoperative control.



Figure 1-2: Sayre glove, with 10 degrees of freedom tracked using fiberoptic cables.

(Atwell, C. (2012, Oct. 15). *Are gesture-sensing interfaces set to replace your mouse and keyboard? ms digit and leap motion say yes*. Retrieved from

<http://www.element14.com/community/community/news/blog/2012/10/15/are-gesture-sensing-interfaces-set-to-replace-your-mouse-and-keyboard-ms-digit-and-leap-motion-say-yes>). Used under fair use, 2013.

One of the first data gloves invented was the “Sayre Glove,” in 1977 as part of virtual reality simulations in an era where instrumentation and processing were in their infancy. The Sayre Glove used an early version of fiber-optic sensors consisting of small illuminated tubes whose light intensity varied based on the angle of bending in the finger (Sturman & Zeltzer, 1994). Since this initial foray into sensing hand positioning, there has been a keen interest in virtual reality applications in the 1990’s (Carlson, 2003), which began to flow over into the commercial world, particularly with the Nintendo Power Glove (Sturman & Zeltzer, 1994). However, due to the difficulties in reading and processing analog signals, typical consumer applications remained limited due to prohibitive costs for accurately sensing and reading information. In the end, data glove technology ended as something of a gimmick.

Despite this initial setback into commercial realms, data glove technology continues to be explored, but in a limited scope. From a brief survey of the available data gloves, it becomes quickly apparent that typical data gloves are not affordable for the average consumer, as noted in the table 1-1.

Name	Degrees of Freedom(DOF)	Sensing Method	Sensitivity	Cost(USD)
5dt Ultra 5	5	Fiber Optic	<1 Deg	\$1,000
5dt Ultra 14	14	Fiber Optic	<1 Deg	\$5,000
Cyber Glove III	22	Resistive	.5 deg	\$15,000
P5	5	Resistive	.5 Deg	\$79.00

Table 1-1: Commercially available data gloves, with their respective sensing methods, degrees of freedom, sensitivity, and cost. (Fifth Dimension Technologies, 2005) (Cyber Glove Systems, LLC, 2011) (Cyberworld Inc., 2013)

As is evident from the majority of the listed data gloves still in production, obtaining an accurate data glove with more than 5 degrees of freedom is exceedingly expensive. Thus one of the objectives of this thesis was to determine a methodology to create a data glove that (i) allows for at least 10 degrees of freedom, (ii) is user friendly, and (iii) can be used by numerous hand sizes. The goal was to achieve all of these features while keeping the costs of production below \$250.

1.4 Sensing Methods

One of the most important factors when designing a data glove is determining the method through which the finger positions will be sensed. In the past 30 years of research on data glove development multiple methods for sensing have been investigated. Some of

the most prevalent methods are visual tracking joints using resistive bend sensors and fiber-optic bend sensors.

From the very onset of data glove technology, fiber optic bend sensors have been used in data gloves, highlighted in the Sayre Glove, which was noted to be the first of its kind (Sturman & Zeltzer, 1994). Fiber optic cables operate using thin glass or plastic tubes to project light along the length of the wire with little or no losses over long distances using total internal reflection along the interior of the cable. These cables have become popular in recent years due to their applications for transmitting data over long distances (Data Connect Enterprise, 2013). Fiber optics are also used for non destructive evaluation as they can detect small deformations. As the change occurs in radius along the length of the cable there is modulation of the intensity at the end of the cable, which can be detected by a simple photosensitive cell. The change in light intensity can be further amplified using series of fiber-Bragg gratings along the length of the wire located directly on the platform subjected to bending. As these gratings are rotated or shifted relative to the incoming light beam, the resultant change in light intensity is read by the photosensitive cell.

Despite this being one of the oldest technologies used by the data gloves, this method still remains popular for a number of reasons. Using specialized fiber-Bragg gratings allows for very sensitive readings of joint locations, while minimizing the interference from outside sources, namely temperature dependence seen in some resistive sensors. It entirely eliminates the electromagnetic interference at the location of the sensor (Baek, Jeong, & Lee, 2002). As such, these sensors are the exclusive option for researchers intending to perform studies nearby large magnets, such as those of an fMRI

used to assess a stroke patient's motor skills. However, these sensors are quite expensive, as producing a custom fiber optic sensor with the correct fiber-Bragg grating specifications makes the sensors themselves expensive, while also requiring a suitably sensitive photosensitive system to read varying light intensity from the sensors.

Another less explored option for sensing human hand position is using visual tracking to determine the joint location using an external camera filming an operator's hand. This can be done in a multitude of ways, such as using a single camera with markers along the user's hand to determine joint locations (Han, 2010) or using a combination of a traditional camera and depth-mapping camera to give a 3D image of the hand (Biswas, 2011). This method is popular because it is very inexpensive, only requiring a camera and relatively simple software. Furthermore, the user is not tethered to any machinery, allowing for mostly uninhibited movement. However, there are some distinct flaws in this method. Namely, using markers along each joint of a human hand requires a significant amount of time to place each marker in their designated spot. Furthermore, if any marker falls off or becomes hidden from the viewing cameras, the software will not be able to recognize the position of the hand. As such, this method is typically limited to academic settings. However, usage of depth mapping and traditional cameras is quickly becoming a more popular method of tracking joint locations. This system has already been become available on a larger scale for tracking human limbs in the Microsoft Kinect (Microsoft, 2013), while a smaller system called the Leap Motion (Leap Motion, Inc, 2013) will be available to consumers for the purpose of tracking hand movements and gestures for computer user interfaces. Still, like traditional cameras, the

Kinect and Leap Motion are limited to what is visible in the camera's field of view, and as such, lacks portability for users.

A third method that was selected for the design of the data glove in this thesis, was resistive based strain sensors. Resistive sensors are common, inexpensive, and simple to use as they require a conventional circuit and small input voltage to return the relevant data. They have a large variety of uses, which range from detecting small deflections in structural elements of bridge (Kang, Schulz, Kim, Shanov, & Shi, 2006), to large deflection purposes such as observing changes in finger position (Gentner & Classen, 2009). Regardless of the application platform, the operation of resistive strain sensors remains the same. At their core, resistive sensors consist of conductive material that changes its resistance due to the volumetric changes under external strain. In other words, as the sensor is strained either in compression or tension from outside forces, the resistivity of the sensor changes due to a thickening or thinning or the resistive element from the strain, allowing the electric circuit to detect a small change in voltage. The voltage change can then be correlated with the magnitude of strain observed by the sensor



Figure 1-3: Flexpoint bend sensors, in 1", 2", and 3" lengths. (Flexpoint Sensor Systems, Inc. (2005). *About the Bend Sensor*. Retrieved Dec 2011, from Flexpoint

flexible sensor systems: <http://www.flexpoint.com/companyInfo/bendSensor.htm>)

Image used with permission of Clark Mower, Flexpoint Sensor Systems, 2013.

One of the most common resistive based strain sensors are simple wired strain gages that are used in various industrial applications. However, these sensors are typically limited to relatively small strains, such as bending and stretching of metal samples, which are very small. In comparison to these samples, an ideal resistive sensor for the data glove would need to be able to absorb large magnitudes of bending to measure the position of a human finger as it bends. As such, a commercial resistive sensor available from Flexpoint was selected for experimentation. Flexpoint bend sensors are designed specifically to detect various degrees of bending for a multitude of applications, including medical, robotics, and automotive uses, among others (Flexpoint Sensor Systems, Inc., 2005). The sensors work in a similar fashion as a wired resistive strain sensor, in that as the sensor is subjected to a bending mode strain, the resistance changes. However, unlike a conventional wired strain gage, Flexpoint bend sensors are comprised of proprietary conductive carbon ink, which undergoes mechanical deformation along its length as it bends. These mechanical deformations result in structural failures that reduce the conductivity and flow of current through the sensor, resulting in a change in resistance.

An additional benefit of using a Flexpoint bend sensor is that it exhibits a significant change in resistance and as such only requires a simple voltage divider and sensor as opposed to some strain gage circuits, which require up to 4 sensors to obtain accurate readings.

1.5 Sensor Analysis

Initial experiments Flexpoint bend sensors showed that the resistance range of the sensors lie between 18 k Ω to 90 k Ω with some variation between each individual sensor. Because most development boards for prototyping hardware use a 3.3 V output, the development board for the data glove uses a voltage divider with a 22 k Ω resistor.

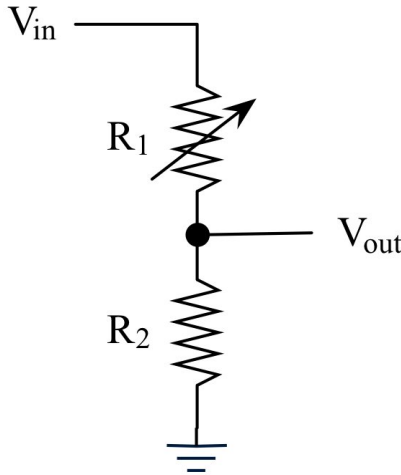


Figure 1-4: Voltage divider diagram. R_1 is the sensor, and R_2 is the static reference resistor.

$$V_{out} = \frac{R_2}{R_1 + R_2} * V_{in} \quad (1.1)$$

By applying the theoretical equation to the initially observed resistances, a change from 1.72 volts to .78 volts accounting for a change of roughly 1 volt difference was observed. This range could be further increased using an operational amplifier.

1.6 Sensor Placement

Having selected the ideal sensor for the data glove, the next step was to consider how many sensors are necessary in order to obtain the needed degrees of freedom to control a human-like hand.

For this data glove, 10 degrees of freedom were decided to be necessary to successfully control the DASH hand and generate the desired grips in order to manipulate a number of tools. These degrees of freedom include finger proximal flexion, distal-medial flexion, thumb distal flexion, thumb medial flexion, and thumb proximal flexion. With these degrees of freedom included, the data glove was capable of differentiating between crucial grips, such as the power grip, precision pinch, and circular grip, which would encompass a majority of typical objects to be handled during teleoperation. Several other degrees of freedom were considered for the data glove but were not included due to difficulty in implementation. These degrees of freedom include finger abduction, wrist flexion/extension, and wrist ulnar and radial deviation.

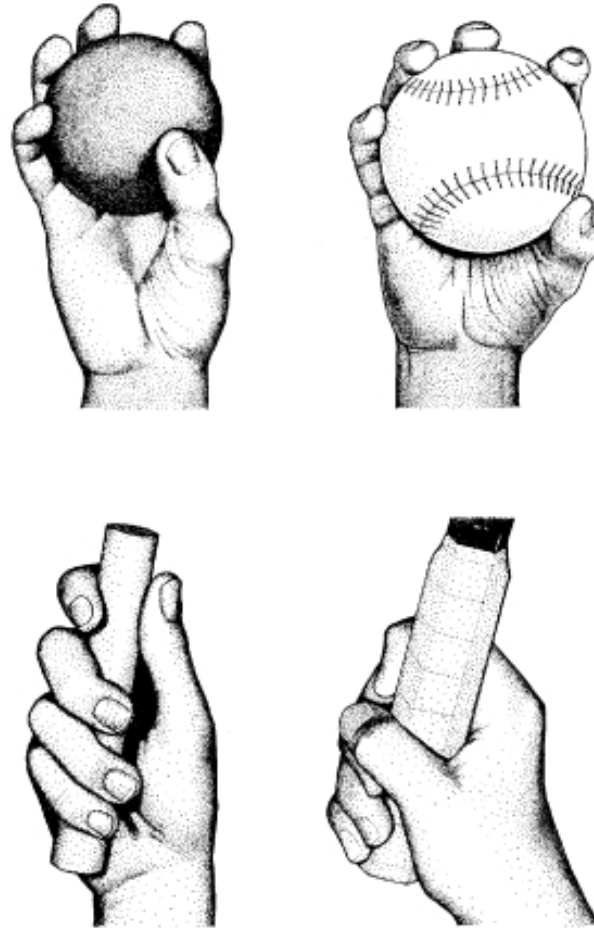


Figure 1-5: Circular grip(above) and power grip(below) are ideal grips for a number of teleoperative purposes. (*The Human Story: The Hands*. (2011). Retrieved from Online Digital Education Connection: http://en.wikipedia.org/wiki/File:EEG_cap.jpg). Used under fair use, 2013.

Placing the sensors required selecting two distinct contact points near the joint in question to track the desired degree of freedom. In typical data gloves, the sensors are sewn into the fabrics, allowing the sensor to bend naturally about the joint. However, one of the objectives of this data glove was to make it user friendly, and as such having sensors sewn into the fabric would be difficult to replace or troubleshoot components in

the event of failure. Furthermore, developing a custom fabric glove would increase the cost of the data glove significantly, particularly in the prototyping stage. Rather than try to fabricate a custom fabric glove, it was far more efficient to find a method to retrofit an existing glove.

1.7 Glove Selection

A variety of gloves were tested for both user comfort as well as adequacy for mounting sensors. Among several key factors is how well the glove fits multiple users, if the glove allowed full range of motion for the user, if the sensors would mount to the glove fabric, and if the glove was comfortable. Some gloves, such as thick rubber kitchen gloves, fit multiple users, allowed for full range of motion, and supported the sensors. However, the glove was exceedingly uncomfortable for a user, due to the lack of breathability of the rubber. Other gloves, such as a thin cotton glove, fit multiple users comfortably, but the sensor was not suitably fixed to the glove, and would move from its fixed base, causing inconsistent readings from the bend sensors.

Considering several options, the ideal glove for retrofitting into a data glove was a battling glove from Underarmor. Consisting of a spandex and nylon weave with leather, the glove is both comfortable and flexible enough to fit a variety of hands. The fingertips were cut off from the glove to aid in comfort for the wearer as well as to allow for larger hand sizes to fit into the glove by not limiting the glove based on finger length alone. The sensors were attached using simple plastic dual lock connectors made by 3M, which allowed for sensors to be moved with respect to the user's fingers to provide a more ideal fit and better bending angles for the sensors, while also providing the necessary stiffness to fix the sensors on the glove.

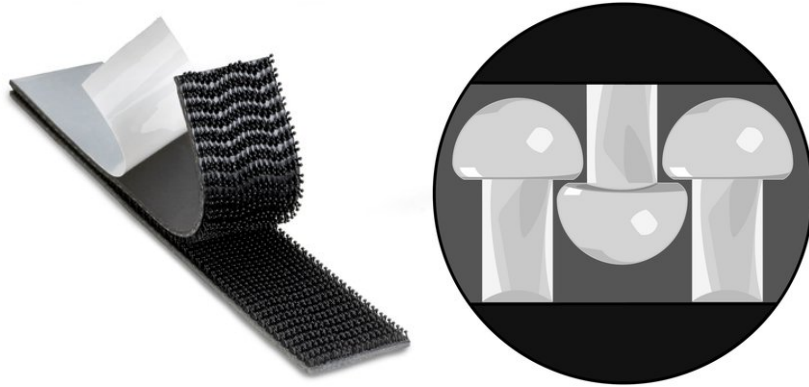


Figure 1-6: 3M™ Dual Lock™ Fastener (left) and close-up of 3M Dual Lock Fasteners(right). The adhesive on the backs of the fastener are simple to apply to the glove fabric, while the interlocking plastic makes for a firm hold, while also showing little wear over time. Image used with permission of Stacey L. Wagner, 3M, 2013.

The other end of the sensor was slid into a small plastic slot made from a binder tab. By folding the binder tab into quarters and then gluing the resultant slot, the binder tab becomes an excellent method to allow the sensor to bend, while also allowing the sensor to move axially along the finger. However, some problems arose with this method for the proximal sensors, as any slight abduction of the fingers would cause the sensor to buckle, creating unintentional bending and throwing off readings. However, a solution was devised that involved using a wider binder slot, allowing the user to abduct their finger, while leaving the sensor free of buckling. Using the same hook and loop connectors, these binder tabs were connected to the glove to allow for bending. Another potential issue is that the sensors for the proximal digits overlap with the sensors for the medial-distal joints while the user's fingers are fully extended. To solve this, the medial-distal sensor was attached directly to the plastic tab used to flex the proximal sensor, allowing the proximal sensor to pass beneath the medial-distal sensor without causing

any issues. With that system laid out, dual lock tabs were attached directly to the glove fabric, eight two-inch sensors and three three-inch Flexpoint bend sensors were attached to the glove.

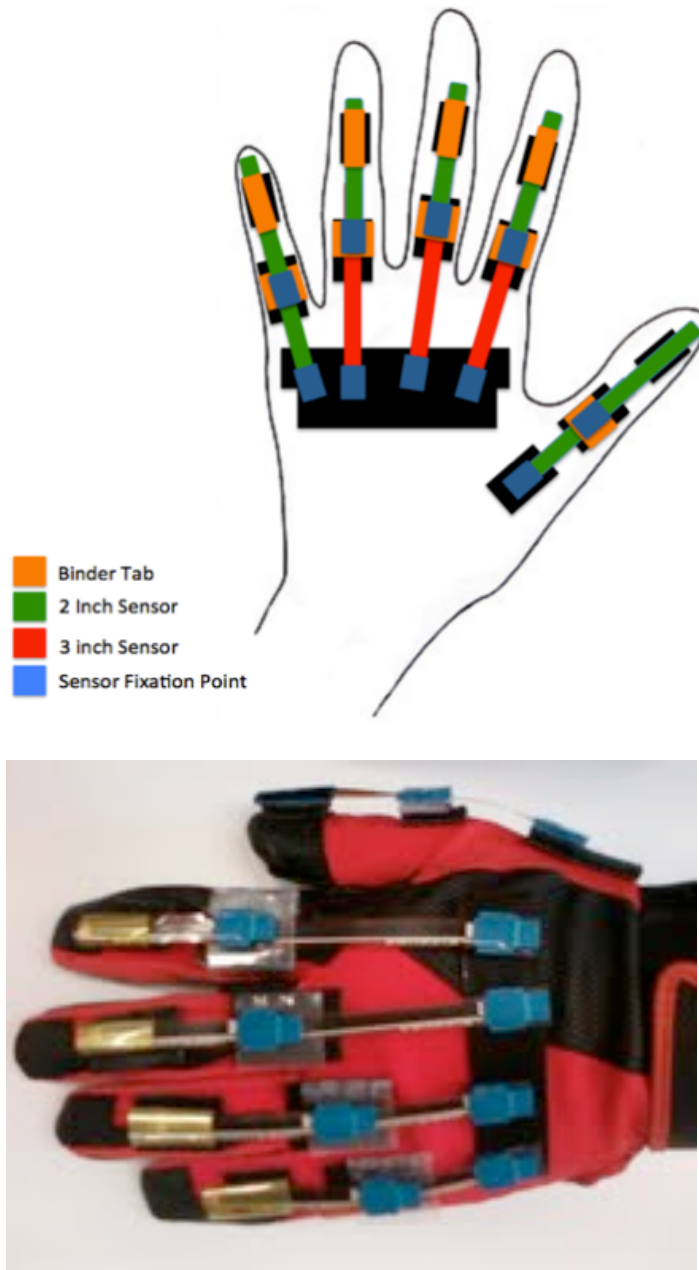


Figure 1-7: Data Glove Diagram(above) and physical Data glove(below), with sensors attached to dual lock tabs, but without wires. Note that the proximal sensors are free to slide beneath the distal sensors to prevent sensor buckling.

Each sensor needs a voltage in and out to obtain a reading, requiring a total of 22 wires to run along the glove. To address concerns of wire clutter and tangles, it was determined necessary to use a 25-port D-Sub connector to act as an intermediary between the sensors and development board, which would allow the gloves to be independent of the board and enable the gloves to be interchangeable with different boards.

Knowing the range of voltage changes from previous testing, there were several other key criteria for the development of the planned data glove. One factor to consider is the number of analog to digital ports on the board. While it is possible to use an external A/D converter to increase the potential number of A/D ports for a board, it is far easier and likely less expensive to find a board which has the required number of at least 10 A/D ports. Also, because this data glove is focused on being a cost effective alternative option compared to commercial alternatives, price is a key factor in selecting a microcontroller.

With these factors considered, a Maple microcontroller board was chosen, which appears to be an ideal option for wiring a data glove. It has 15 A/D ports with 12 bit conversion, which would allow 4096 individual values over 3.3 volts, allowing for a minimum resolution of .8 mV. Along the observed ranges from a bent sensor, this would mean that the Maple board is able to resolve roughly 1175 individual values. Furthermore, the Maple board costs only \$45, consistent with the goal of producing a cost effective data glove.

1.8 Testing the Data Glove

Once the data glove was built and wired, the next step was to determine the effectiveness of the data glove from a practical standpoint. With the developed glove and

circuit the goal was to be able to observe a sensitivity of less than 1 degree of flexion in each digit. The testing was conducted using a National Instruments 16 bit A/D converter for ± 10 volts. This data acquisition system provides a minimum sensitivity of 0.3 mV, which would produce a good comparison with the 0-3.3V 12 bit A/D sensitivity, which is 0.8 mV.

Ideally, it will be possible to compare how much voltage is generated by the data glove to the current finger position. To accomplish this, a high-speed camera in tandem with the DAQ was used. To successfully use the camera to track the joint angles, several distinct marks were placed on the side of the data glove being filmed. Because the objective was to track the joint, three points were placed along the side of the glove, near the finger. Using white acrylic paint, a dot is placed before the knuckle, at the knuckle, and after the knuckle, which allowed an angle to be processed from the finger positions. Then, using a high-speed camera filming at 60 FPS, simple flexing motion was performed with the distal and proximal index joints.

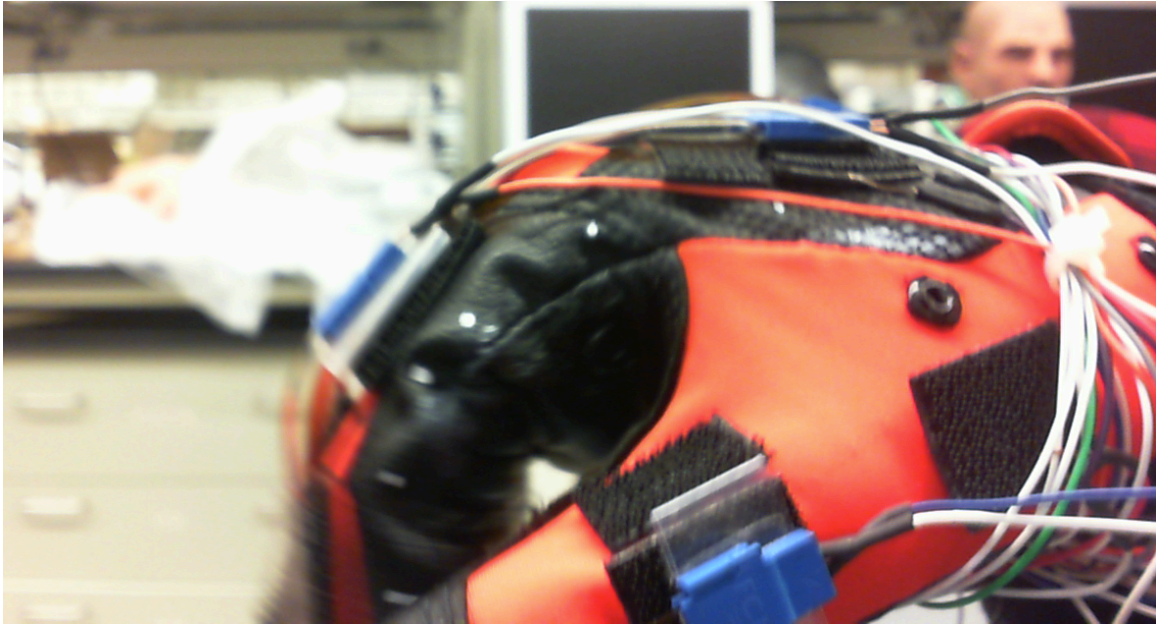


Figure 1-8: Tracking Points on the data glove. A net total of five points were needed to accomplish tracking, having one dot along the palm of the hand, the proximal knuckle, the proximal digit, the medial knuckle, and the medial digit.

The next step was to find a method for synchronizing the camera frames to the recorded data. Because the camera and voltage data were recorded separately, a new method must be used to synchronize the two sets of data. To accomplish this, the circuit had a small flip switch between the voltage divider and DAQ. By flipping the switch in front of the camera, the exact frame can be determined where the voltage jump corresponds to the switch being flipped

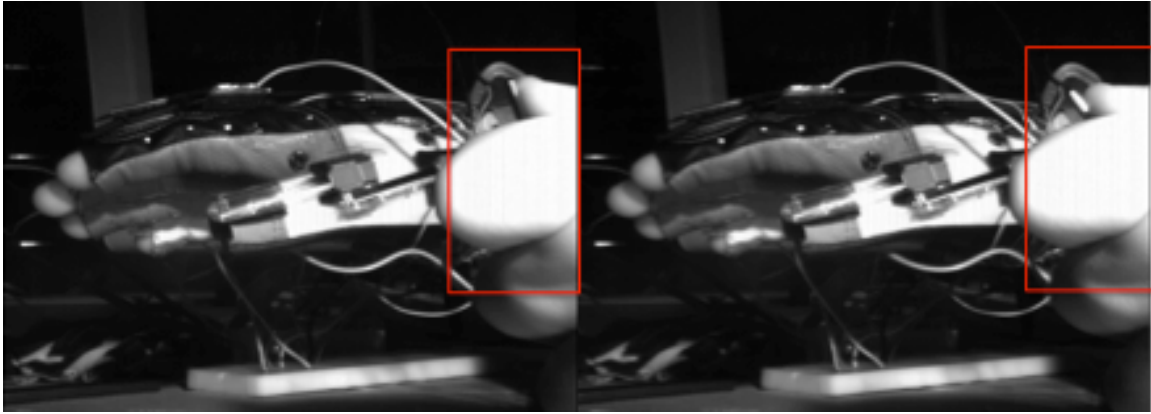


Figure 1-9: Two frames indicating the start of the switch being flipped.

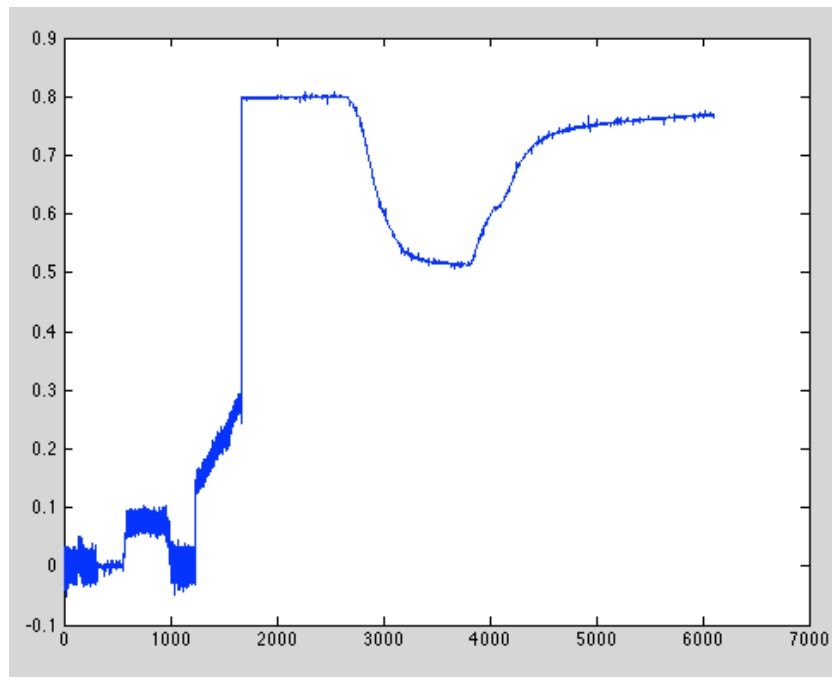


Figure 1-10: Voltage output before and after switch flipped.

Once the switch was flipped, the degree of freedom was gradually flexed from fully extended to fully flexed, held for half a second, and then returned to the starting position. Then, using a freeware image0processing program ImageJ, each of the dots along the data glove were manually marked in each frame to produce a series of

coordinates of the marked pixels. Then using these points to determine two lines and the respective angle between them gives the angle of flexion. Using these two data sets, correlation can be made between the angle of flexion and the voltage output from the voltage divider.

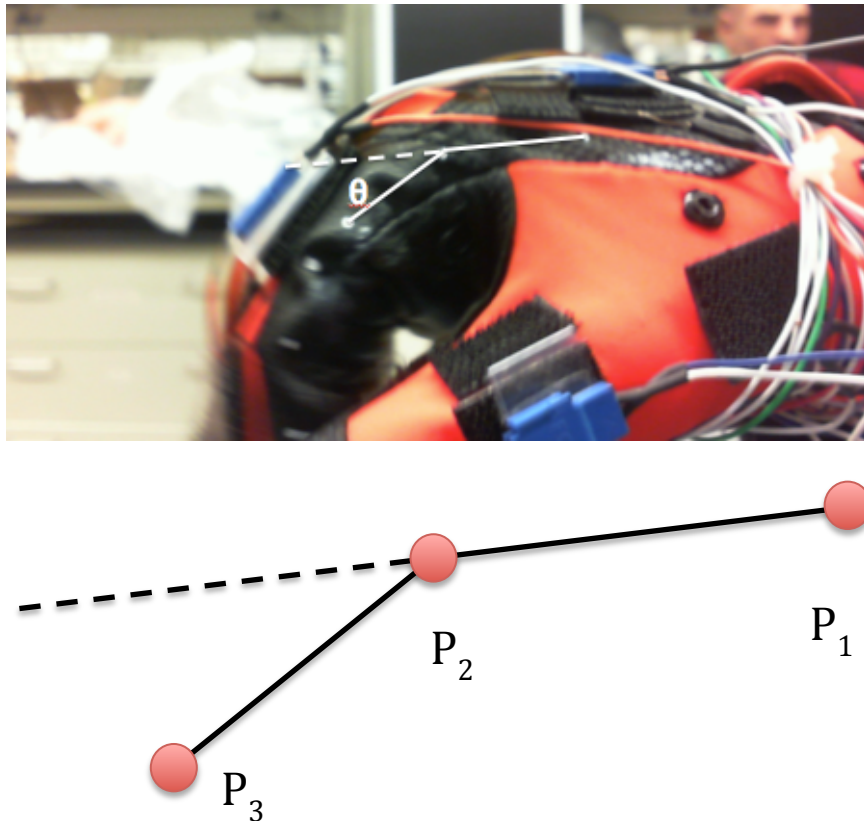


Figure 1-11: Data Glove with tracking points(above) and link segment model(below). Each dot is recorded as a series of x and y coordinate pixels, which were used to find the relative positions of each link.

$$\begin{aligned}
h_1 &= \sqrt{(P_{1x} - P_{2x})^2 + (P_{1y} - P_{2y})^2} \\
h_2 &= \sqrt{(P_{2x} - P_{3x})^2 + (P_{2y} - P_{3y})^2} \\
d_1 &= P_{1y} - P_{2y} \\
d_2 &= P_{2y} - P_{3y} \\
\theta &= \sin^{-1} \frac{d_2}{h_2} - \sin^{-1} \frac{d_1}{h_1}
\end{aligned} \tag{1.2}$$

1.9 Results

Using a linear fit to the data, a reasonably linear correlation between degree of flexion and voltage output was visible, with a residual norm for the distal joint at 0.21 and the proximal joint at 0.13. While ideally a residual norm of 0.05 or fewer would be preferable, there was little observed improvement in the fit by attempting higher polynomial fits. Even using a 4th order polynomial to fit to the data, the residual norm for the distal and proximal joints were 0.098 and 0.049, which did not account for enough of an improvement to justify the increased difficulty of future attempts at calibration for a user.

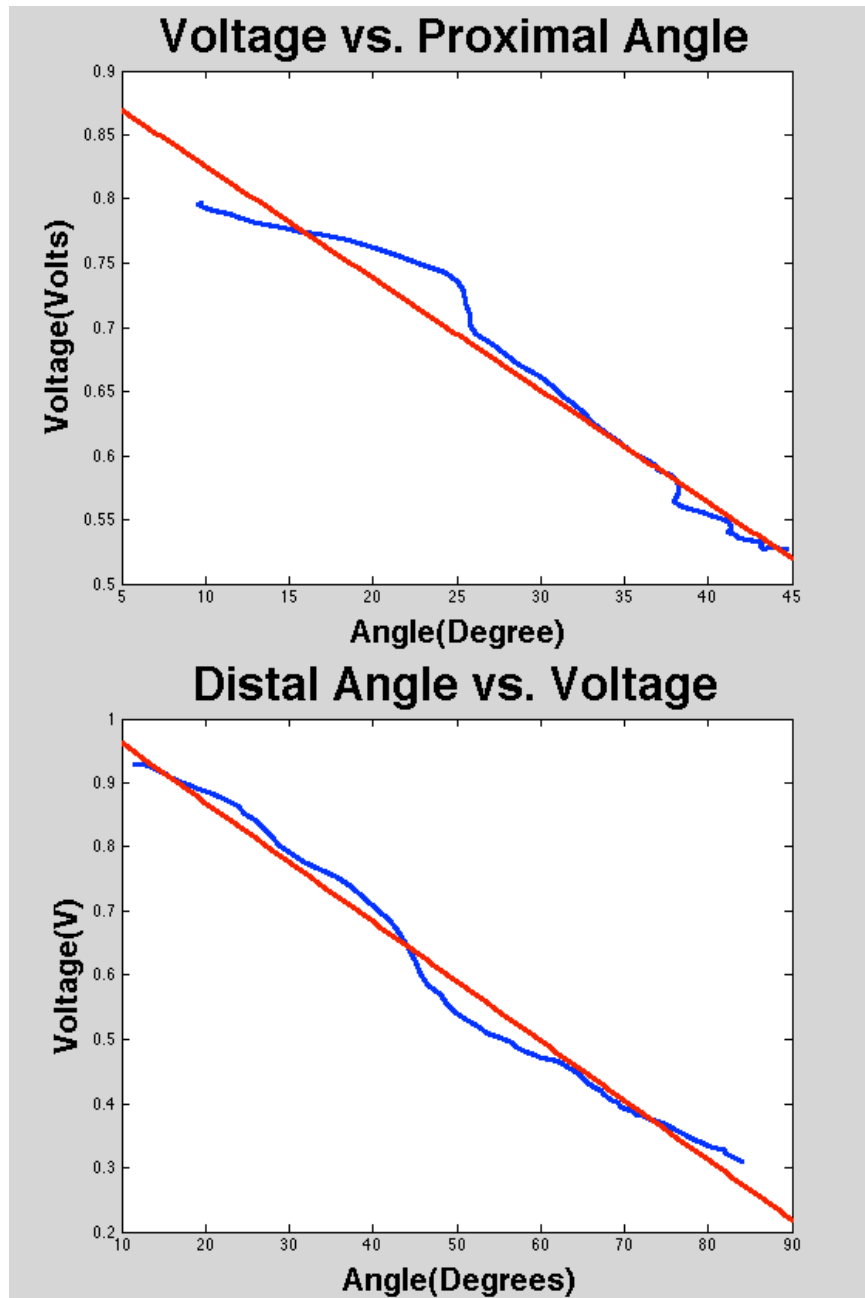


Figure 1-12: Proximal Angle vs. Voltage with linear fit(above), Distal Angle vs. Voltage with linear fit(below).

Of particular interest is the behavior of the proximal bend sensor. Upon observing the data plot, there appear to be two different linear slopes, which transition after the 25-degree flexion mark. Upon further investigation, there was a change in the bending of the sensor due to contact with the proximal knuckle. From 0-25 degrees of flexion, the sensor is not in contact with the proximal knuckle, 25 degrees on the sensor is in contact. This may explain the two distinctly different linear trends.

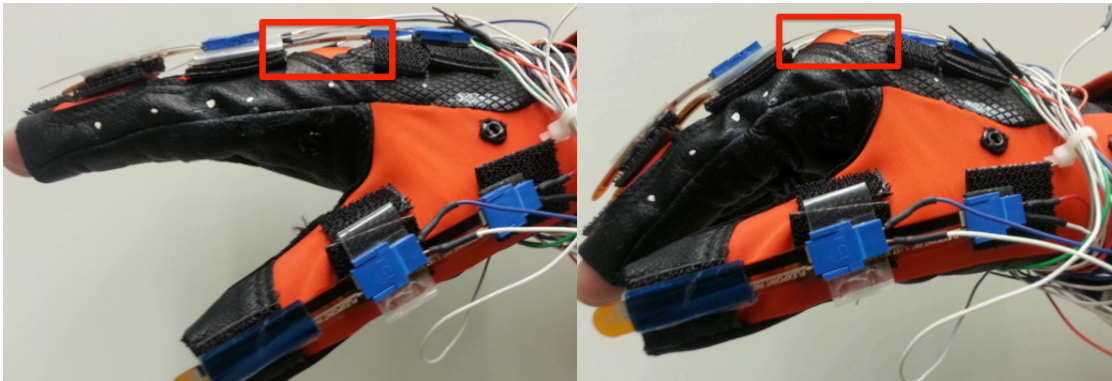


Figure 1-13: Data glove in the extended position(left) and at 25 degrees of flexion(right). Note the sensor is not in contact with the proximal knuckle in the extended position, but is in contact in the flexed position.

However, this experimentation method was not without error. Due to the flexibility of the glove around the user's hand, the tracked points do not always accurately reflect the true position of the user's joint. Typically, marking points for tracking assumes a rigid and unmoving marker, which is not possible with this glove. Furthermore, because the points were tracked manually, some error is to be expected during the process, especially since relatively small errors, such as clicking the edge of a dot rather than the center, can significantly skew data. The tracking data presented here is the result of several attempts, and appears to be adequate.

1.10 Noted issues and Future Fixes

While the data glove appears to perform well for the given investment, there are several issues, which should be accounted for in future versions. First and foremost, while the design of fixing sensors to the outside of the glove made troubleshooting and attaching sensors user friendly, it produced other concerns, such as sensor buckling and external influences to the sensors. The proximal digit sensors would occasionally be obstructed by the dual lock pads, resulting in the sensors producing a high resistance, which would be interpreted as a high angle of flexion. Ideally, the next generation of this glove would follow the methods of other data glove groups, and use an inlaid sensor design to mount the sensors directly to the hand, without the possibility for external interference.

Furthermore, additional degrees of freedom on the data glove would be necessary to track a user's hand for other hand positions. Most importantly would be to add a sensor to detect thumb rotation and abduction, which would allow for inclusion of a key or card grip. Finger abduction sensors would also be able to take advantage of additional degrees of freedom on the DASH hand. Also, it would be ideal to eventually track wrist position on the data glove. This could be performed with IMUs, but this would double or triple the cost of the glove, which would conflict with the goal of making the glove as cost efficient as possible.

Chapter 2 : Robotic Hand Feedback

2.1 Need for Robotic Hand Feedback

By adding the data glove to control the robotic hand, the robotic hand will be able to take on new tasks in manipulating objects as opposed to simply typing. The initial control scheme of the robotic hand only allowed for open loop control for the robotic hand. With this scheme, the robotic hand would go to several predetermined positions to touch various points on a keyboard. If there were failures in this system, it was due to the keyboard being out of position, and would be corrected by an operator. However, since the controls scheme using the data glove will allow for a significantly greater degree of control for a greater variety of situations. Thus, there must be additional methods of feedback to the operator.

With the current control scheme, robotic hand position can be observed based on servo position. This can be verified through visual inspection, either through a camera feed or through direct observation, although this is not ideal. In cases of an obstructed finger, motors will continually draw as much current as possible to try to force the finger into an appropriate position. This is concerning, since this kind of activity can be detrimental to the motors of the hand as well as the hand's physical structure. Furthermore, as the robotic hand progresses in strength and gripping capabilities, there is a possibility of the hand being unable to handle objects with suitable dexterity and gentleness. In this instance, it becomes concerning to handle objects without certainty that the robotic hand will not damage or otherwise poorly handle the object, particularly without a user in the immediate vicinity to observe the manipulation of the object. With this in mind, it is important to begin to determine a method to develop and implement

sensors on the hand that will be able to observe and quantify forces applied along the fingers, both for limiting factors of position feedback as well as force feedback for handling objects.

2.2 Methods of Feedback

There are numerous methods to obtain feedback from a robotic manipulator, some of which involve complex mathematical models that calculate external forces based on joint torques, while others involve using a variety of sensors to obtain large arrays of force feedback values to determine the precise location and application of force to the robotic hand.

One method is to determine applied force based on the output from the various motors. By knowing the baseline current draw for the motors, using a current threshold motors could be limited to a set amount current, preventing excessive motor torques. With a system like this, regardless of where an external force is applied to the joint, it will become clear from the source motor that excessive current is being drawn, and allow for the motor to release. This method would be particularly useful for a compliant mechanism in the finger to prevent excessive joint stiffness in response to obstructions or manipulation. However, with this method it would be difficult to determine the pressure applied by the joints on an object without direct visual verification, since it will be unclear which part of the joint is being obstructed.

Another option is to place sensors at various points of interest along the hand that can detect the pressure applied by objects at various points on the hand. This is typical in robotics, as sensing mechanisms on manipulators are able to determine the amount of pressure applied to an object, which can be used to determine how to best handle objects

with the appropriate amount of force. This can be done with a variety of sensors, assuming they are of appropriate size and sensitivity to observe the applied pressures. There are a number of these sensors available for commercial purchase and application, which makes this an ideal method for quick and effective implementation on the robotic hand.



Figure 2-1: Biotac robotic manipulator sensor. Image used with permission of Matt Borzage, Syntouch LLC, 2013.

A prime example of a commercial tactile sensor for use in the DASH hand is the BioTac sensor, which was developed by researchers at University of Southern California in 2006 for prosthetics, but has since evolved for a variety of robotic feedback purposes (Syntouch, LLC, 2013). The sensor is capable of observing texture, pressure, as well as temperature using a specialized conductive fluid, electrode arrays, and barometer with a sensitivity greater than that of a human finger (Fishel, 2012). While effective, these sensors come in predesigned plastic fingertips and skins, though custom sensors can be designed. However, the cost of this would be prohibitively expensive, as the primary

focus of this research is not only to perform gripping activities with the robotic hand, but to also perform such activities at a minimal cost.

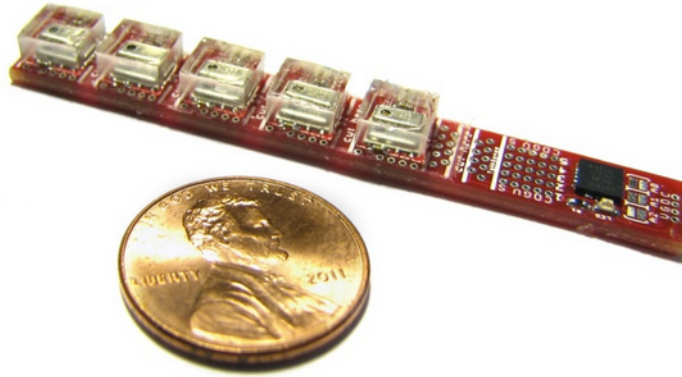


Figure 2-2: Takktile Sensor, with penny for size comparison (Takktile, LLC. (2013). *Takktile Products: TakkStrip*. Retrieved from Takktile: <http://www.takktile.com/product:takkstrip>). Image used as part of the Creative Commons License.

Another example of pressure sensing is the sensor, which is a rubber encased MEMS barometer designed by Harvard researchers. Using the barometer's sensitivity, the sensor is able to determine both pressure applied from an object, as well as texture based on fluctuations in the sensor signal (Tenzer, Jentoft, & Howe, 2012). When arranged in an array of sensors, a robotic hand is capable of obtaining a field of pressure values. However, the sensor is still prohibitively expensive, costing over \$100 for a single sensor. Furthermore, each sensing unit is 5 mm in thickness, while the desired thickness for the fingertip sensor would be less than 1 mm.

As an alternative, a common option for commercial pressure sensors are piezoresistive sensors, which operate using two conductive layers of material that are pressed together with increasing force, causing a decrease in the resistivity of the sensor. These are significantly less expensive, while also thin and easy to implement, making them ideal for application on the DASH hand.

2.3 Piezoresistive Attempts

As mentioned earlier, piezoresistive sensing is quite suitable and easy to implement for determining the forces applied by the robotic hand on the objects. The Next task was to determine exactly what type of piezoresistive sensors would be used. While it is possible to purchase a number of piezoresistive sensors, an important criterion in the selection was to identify a sensor that can be developed in the laboratory at a fraction of the cost of commercial sensors. With this objective in mind, custom piezoresistive sensors were produced using inexpensive conductive fabric from Less EMF.

Fabricating a piezoresistive sensor needs both a conductive as well as resistive element in order to create a separation of the two current sources. This sensor will use a stretch conductive fabric which has negligible resistance over its length, and a resistive element called Ex-Static, which has a resistance of $10^5 \text{ k}\Omega$ per square inch (Less EMF Inc., 1996). The piezoresistive sensor must have the resistive element between the two conductive layers, with the conductive layers unable to contact one another. As the force is applied to the sensor, the distance between the two conductive layers will decrease, and charge will be transferred across the two layers. When one lead is attached to one end and another lead is attached to the opposite end, the sensor acts like a resistor that varies with pressure.

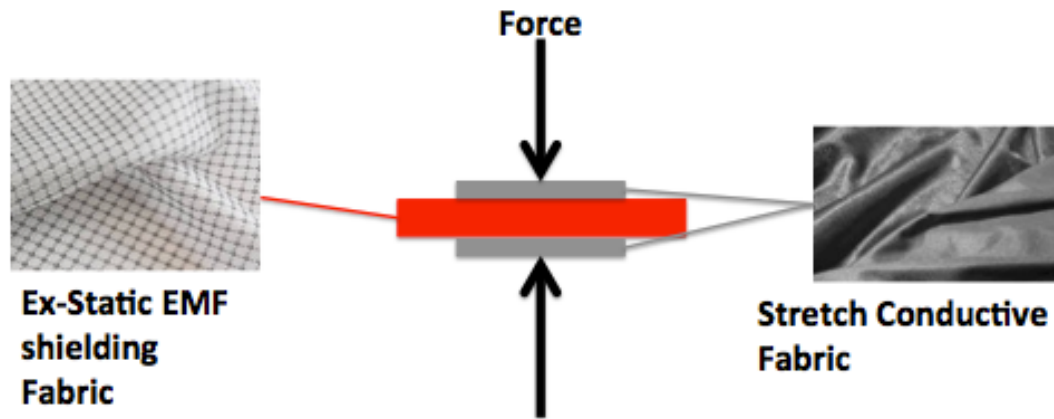


Figure 2-3: Schematic of fabric arrangement for pressure sensor.

An initial test was conducted using two 6 x 2 inch strips of stretch conductive fabric, and a 6.25 x 2.25 inch strip of Ex-static. It is important to note that the Ex-Static strip is larger than the conductive fabric in order to prevent contact between the two layers of conductive fabric. When the strips of fabric were in place, they were attached to each other using 3-inch wide strips of tape. Initially, spray-on adhesives were used, but the fabrics were held together too tightly and did not show a significant change in resistance with pressure. Instead, using tape to seal the fabric in place and attaching two wires to each side gave better performance for the piezoresistive pressure sensor.

Under no load, the sensor exhibited a resistance in excess of 20 M Ω , but under light loads such as 20 g, the resistance dropped to 1 M Ω . Under higher loads, the resistance continued to drop, and approached a resistance of under 1 k Ω with an applied load of 500 g. Thus, the sensor is very sensitive to light touches, but cannot distinguish clearly between degrees of pressure. To accommodate this, additional layers of Ex-Static

were added between the layers of conductive fabric. With this new composition, the light load resistance increased to 5 M Ω , while the heavier load resistance increased to 500 k Ω .

The current robotic hand used in this thesis was primarily 3D printed plastic, and as such had little compliance for gripping. Many manipulators have some kind of soft covering that aids in gripping, much like human skin, as the soft surface increases surface area with additional pressure. Because one of the objectives of this humanoid robot is to mimic human hand function, the EMDL lab has had significant experience working with a silicone based artificial skin called Dragon Skin developed by SmoothOn (Smooth-On Inc., 2011). By using the silicone mixture with an appropriate amount of thinner, Dragon Skin can be made to mimic the mechanical properties of human skin. With this in mind, it would be ideal to be able to place a sensor within a coating of Dragon Skin to act as a skin-embedded sensor for the robotic hand.

Taking the same 6 x 2 inch sensor developed earlier, sensor was placed in a slightly larger container cut from cardboard, and was then filled with a Dragon Skin mixture. Initial attempts were unsuccessful in embedding the sensor in the skin, due to the fact that the sensor started “floating” on the top of the container, and failing to remain in the middle of the mixture. However, further attempts were successful after the sensor was pinned in place using safety pins, which were removed after the silicone had hardened. The result was a 7 x 3 x .5 inch strip of silicone with an embedded sensor. After testing the sensor under the previous load conditions, the resistances for no load, 20 grams and 500 grams were virtually unchanged. It should be noted that while the sensor performed well when placed against a flat surface, even moderate amounts of bending quickly set the resistance to below 100 k Ω , which would indicate very high loads even

though none are present. As such, it became quickly clear that the sensors must remain against a flat surface in order to be accurate.



Figure 2-4: 6 inch fabric based sensor embedded in silicone.

While the larger sensor worked well as a proof of concept but, the actual sensor needs to be able to fit within a 0.75 x 0.5 inch fingertip. To accomplish this, the strips of Ex-Static and conductive material must be shrunk in dimensions. Though the sensor was able to fit on the fingertip, it was difficult to secure the wire leads to the conductive fabric, despite use of wire glue, and the sensors became too thick due to the necessary layers of Ex-Static to allow for better ranges of pressure values. A final attempt was made using the conductive fabric by making a very thin sensor, and using the conductive fabric itself as a lead, rather than a wire attached to the fabric. This sensor fit within the fingertip, and was not as fragile as in the earlier attempts. However, the sensor was too sensitive to bending along the length of the fingertip, resulting in noisy and difficult to process data. Furthermore, even with usage of a laser cutter to make very specifically sized fabric squares, further attempts at producing sensors resulted in each fabric based sensor being different, with little consistency between each sample. As a result, it was decided that it would be better to use a commercial pressure sensor, though using fabric

based pressure sensors may remain a viable option for larger sensors that cover a greater surface area.

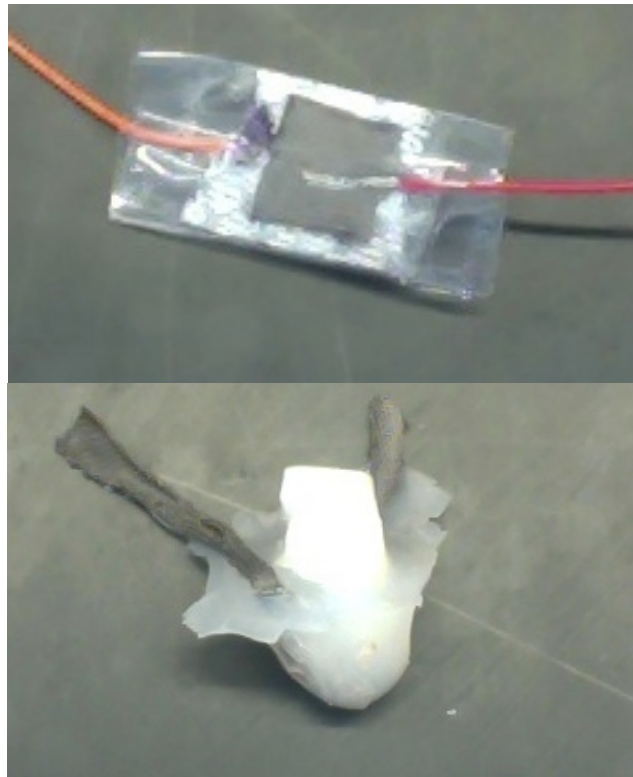


Figure 2-5: Fabric based sensor surrounded in tape(above). Early attempts at embedding fabric based sensor in silicone(below).

When deciding on a commercial pressure sensor, it was important to search for simplicity and robustness in order to obtain a sensor that could suit the project's needs and not require a sizable overhaul to the controls and resources of the lab's current robotics. Flexiforce sensors offered an ideal combination of simplicity, robustness, and cost effectiveness (\$20 per sensor). The commercially available sensor has a sensing pad, which operates using the same principle as the fabric sensors. It is composed of two parallel conductive pads with a separation between the two layers that decreases with

increasing pressure. However, unlike the fabric-based sensors, these sensors are a fraction of the thickness, and have a factory guarantee of consistency between each sensor, eliminating the need for individual calibration.

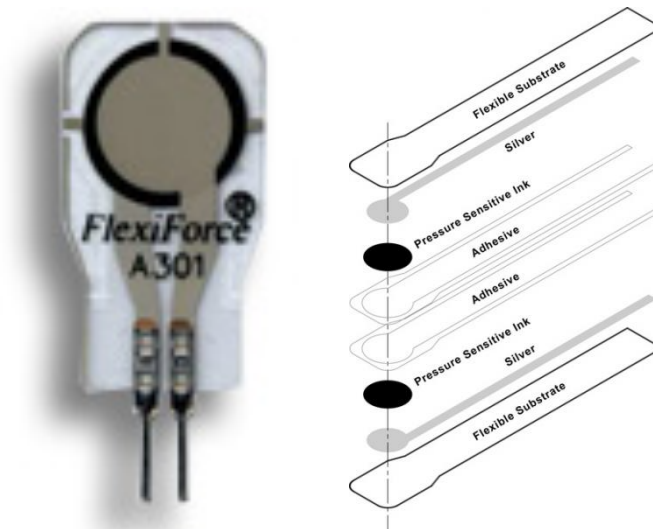


Figure 2-6: Flexiforce sensor 4201(left) and sensor composition(right). (Tekscan. (2013). *Tekscan: Flexiforce Sensors*. Retrieved from Tekscan Website: <http://www.tekscan.com/flexible-force-sensors>). Used under fair use, 2013.

2.4 Molding Process

With these new sensors, the next step was to find a method to attach them to the fingertip. While gluing the sensor directly to the fingertip appears to be a viable option, a better alternative was to add a compliant surface that allows for better gripping, such as Dragon Skin. Ideally, the sensor can be added to any fingertip, and can be easily replaceable in the event that the hand needs to be repaired. As such, it would be preferable to find a molding process that would allow embedding the sensor within a layer of Dragon Skin to form a fingertip that would go over the distal joints of the robotic hand. Adding these fingertips lays the groundwork for a system that both allows for

feedback to a user, while also enhancing the capabilities of the robotic hand by adding compliant gripping mechanisms.

Solidworks was used to produce a hollow, three part mold with an open top to be rapid prototyped. By making a recess in a solid block using a scaled up model of the fingertip, the mold for skin that is slightly larger than the fingertip can be produced. This block can be split into two separate portions, which would allow for easy removal of the sample from the mold, and make the mold reusable

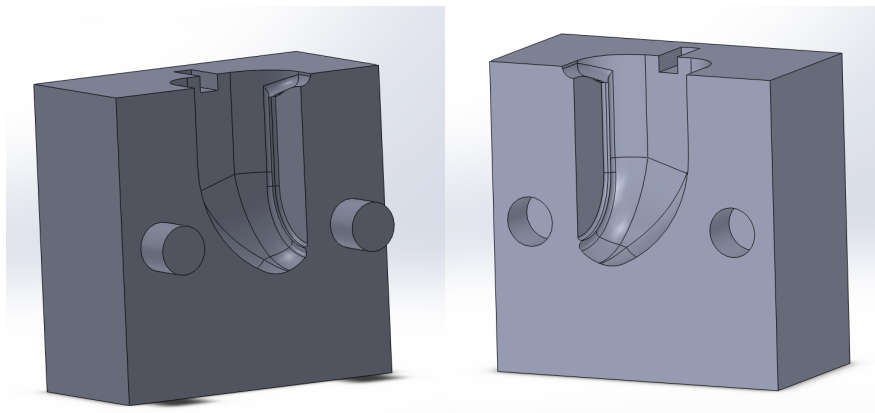


Figure 2-7: Left and right sides of the negative mold. Small circular extrusions were added to the left mold, while indentations of the same size were added to the right mold to ensure consistent placement of the two halves of the mold.

Because the objective was to create a silicone skin around the robotic fingertip, this process also needed a positive mold to insert into the main mold to produce the shell-like structure. This positive mold would also need to be able to hold the Flexiforce sensors in place during the molding process to ensure that the sensors maintain a consistent position between each sensor. With these needs in mind, a positive mold is produced that uses the original CAD from the robotic fingertip, and adds a 1 mm x 1 mm

cross-sectional bar that fits into a small slot on the negative mold to consistently place the positive mold. A small slot was also added into the negative mold with dimensions of .5 x 10 mm to ensure consistent placement of the sensor during the molding process. With this sizing, the metal leads of the sensor can fit out the top, while the remainder of the sensor cannot, which allows the sensor to be fixed in place. While the sensor is still able to sink down into the mold, a small amount of tape or paperclip can help keeping the sensor in place.

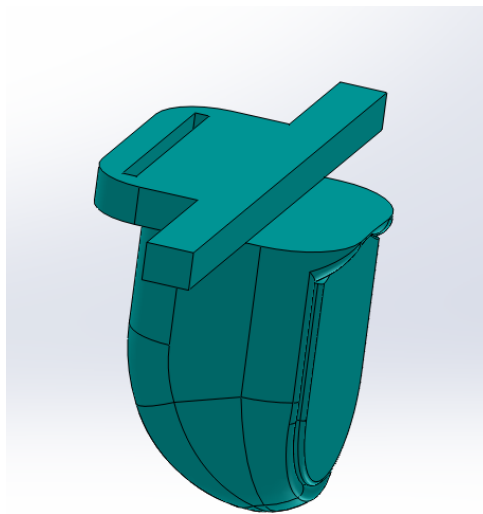


Figure 2-8: Positive Fingertip Mold. The slot was added to ensure consistent placement of the Flexiforce sensor with each mold, while the bar running along the top of the positive mold held it in place within the negative mold.

The final result was a three-part mold: a left and right negative mold, and a positive mold. To produce a fingertip, first the two parts of the negative mold must be firmly placed together and clamped using a c-clamp. Due to the small circular indentations on the molds, it should be clear on how to correctly orient the two parts.

Then, a small amount of silicone must be mixed and place within the mold. It is optional to add thinner to the mixture to make the resulting silicone more pliable, or to even add some sediment such as fine grain sand to allow for additional friction for grip. Soon afterwards, a sensor is placed along with the positive mold into the negative mold, before the silicone can harden. After snapping the rods in place, the mold should be ready to sit until hardened. It is important have correct placement of the sensor within the mold so that the sensor does not touch the edges of the negative mold, as this can create a mold that does not fully contain the sensor and can cause a split near the fingertip.

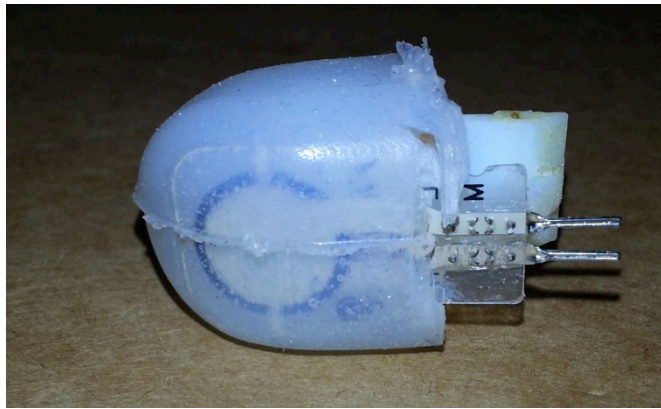


Figure 2-9: Fully molded embedded sensor attached to robotic fingertip.

Once the embedded sensor is complete, a robotic fingertip is printed to attach to the embedded sensor sleeve. Some more experiments were conducted to find an optimum way for easy removal of the embedded, including using setscrews or a reduced positive mold size to increase the tightness of the silicone on the fingertip. However, the embedded sensor still came off easily. Instead, it was easier to attach the sensor using Gorilla Glue by putting a thin coat of the adhesive around the rapid prototyped fingertip, and then attach the embedded sensor to the fingertip. While this makes removal of the embedded sensor very difficult, it is still removable, and a new sensor can be attached once the previous adhesive has been sanded away. If it is not preferred to strip the sensor

off of the fingertip, it is relatively easy to print a new fingertip and mold a new embedded sensor as well. However, if the robotic fingertip was not rapid prototyped, this would not be preferred, since milling or molding a metal fingertip would be significantly more exhaustive, making the fabrication of an entirely new fingertip prohibitively difficult.

2.5 Sensor Validation

According to the Flexiforce website, the sensor should have a non-linear response to forces, following a mostly logarithmic trend. Based on this information, using resistance measurement to directly correlate with the force will not be possible. Thus the data will require some form of post-processing or alternative circuit to observe the voltage.

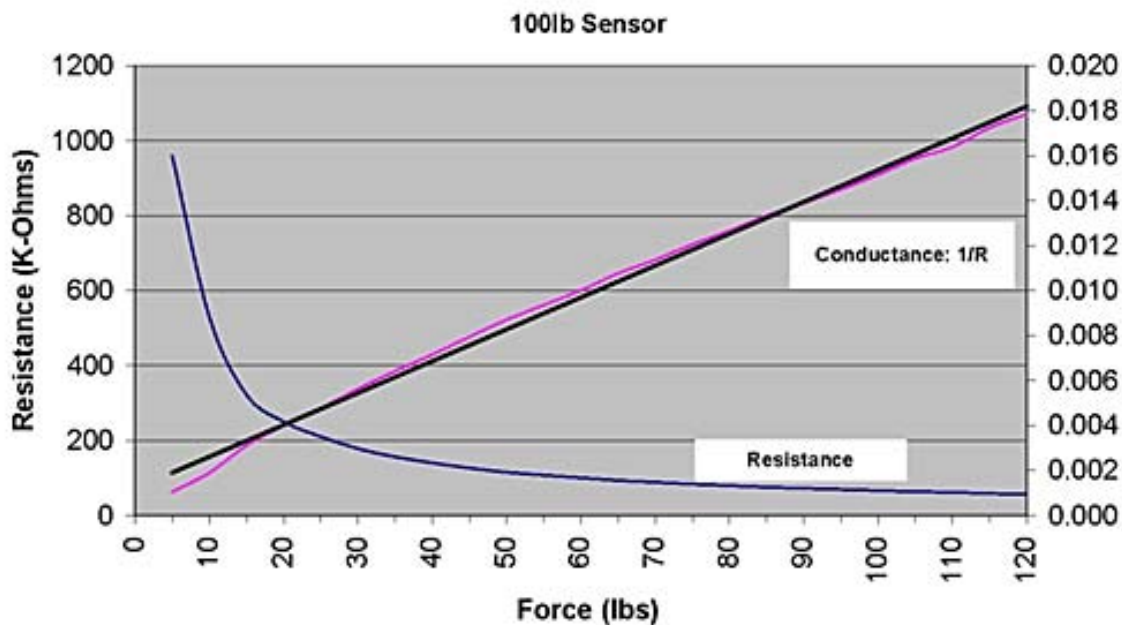


Figure 2-10: Theoretical response of 100 lbs sensor resistance in blue (Tekscan. (2013). *Tekscan: Flexiforce Sensors*. Retrieved from Tekscan Website: <http://www.tekscan.com/flexible-force-sensors>). Used under fair use, 2013.

The first goal is to test the response of the sensor to a variety of forces to observe the response of the embedded sensor, and check that the experimental response matches with the theoretical prediction. For this purpose, a test stand was developed that allowed applying a known force directly to the embedded sensor attached to the fingertip.

The basic design for this test stand is a flat plate with two or more posts running perpendicular to the plate. A second plate will be able to slide along the posts, and will be able to press directly onto the sensor. Additional weight can be added to the top plate to modulate the applied force. While it would be possible to mill and construct a test stand from stock aluminum, an alternative solution is to use a manual drill press, and apply weight plates to the top. Rather than using a drill bit to press on the embedded sensor a machine screw is used, which would apply force over the entire sensor, rather than a concentrated point.

For these tests, the sensor was attached to a fingertip, but not to the robotic hand. As a result, the sensor was free to roll and shift unless otherwise restrained. To alleviate this, three holes were drilled and threaded into a small piece of aluminum, and three 1/8 inch bolts were set into the plate. With these bolts in place, the fingertip can be slid in between them to be held in place from three directions without interfering with the application of pressure from the drill guide. Additionally, the fingertip was held in place from the rear using an 8.4 kg weight, in order to simulate the fingertip being attached to a rigid finger structure, and prevent the fingertip from sliding away during large pressures.



Figure 2-11: Aluminum plate with bolts to hold sensor sample in place.

To apply weight, four $\frac{1}{2}$ pound plates, one 1 pound plate, and one 6.39 pound cylindrical weight were available to add to the chuck of the drill guide, which weighs 0.75 pounds. By using these weights in combination, it was possible to apply 11 different forces to aid in characterizing the behavior of the embedded sensor against the expected response to various loads.

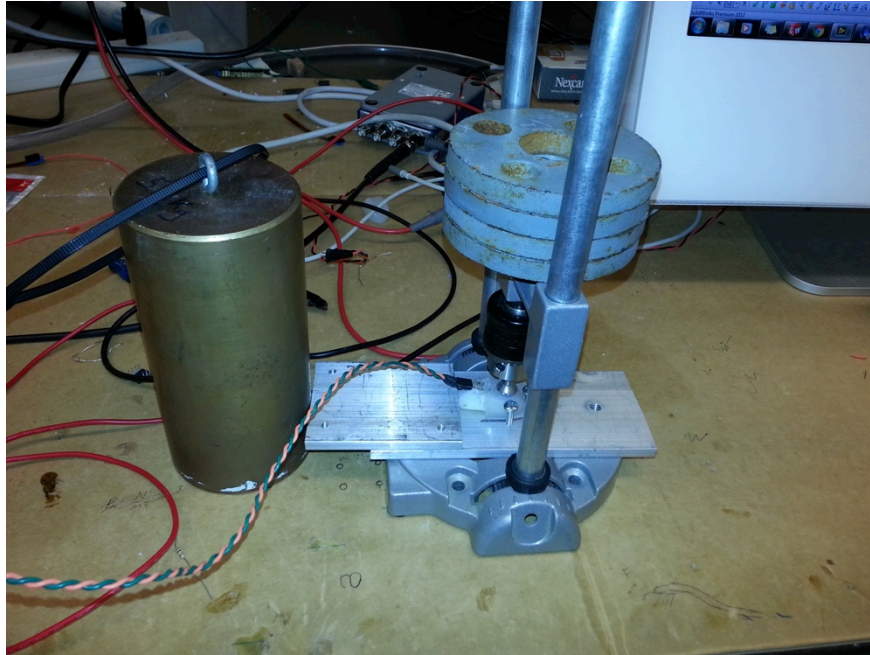


Figure 2-12: Experimental setup with 2.75 pounds of force applied with drill guide.

After applying a series of weights up to 8.39 pounds, the sensor appears to follow the expected response, which can be observed below.



Figure 2-13: Plot of sensor response due to weight in terms of resistance

However, it is clear that the relationship between pressure and resistance is non-linear. To deal with this, there is a recommended circuit from Tekscan to read from the sensor, which operates as an inverting amplifier.

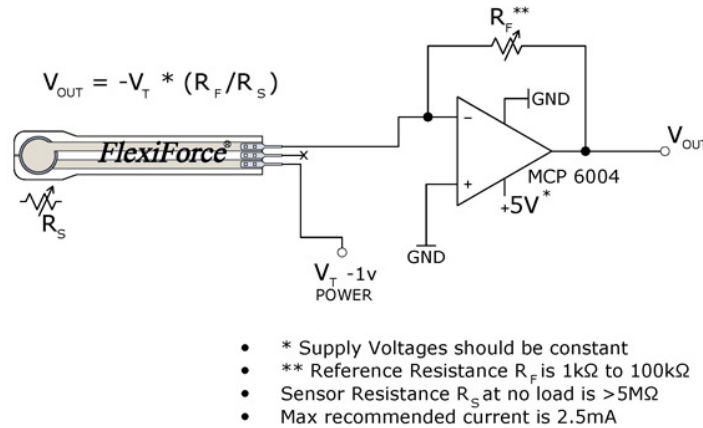


Figure 2-14: Recommended circuit and governing equation from Tekscan(Tekscan. (2013). Tekscan: Flexiforce Sensors. Retrieved from Tekscan Website: <http://www.tekscan.com/flexible-force-sensors>). Used under fair use, 2013

2.6 Results

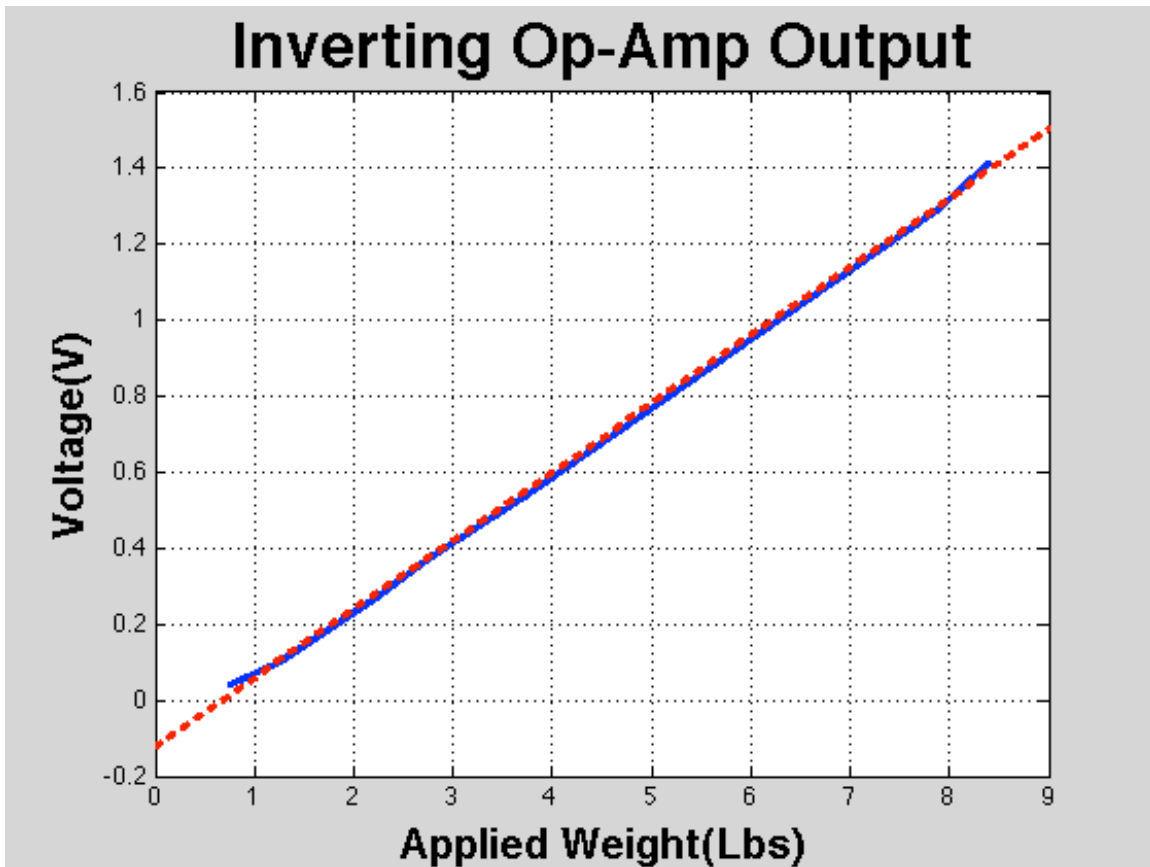


Figure 2-15: Inverting amplifier with linear fit.

The circuit used an input voltage of -5 volts, and a reference resistance of 22 k Ω . With these inputs, the same ranges of weights were applied to the fingertip, resulting in a linear trend, with a residual norm of .034, suggesting a strong linear correlation.

2.7 Conclusions and further work

The embedded sensor appears to have a suitable response that can be clearly observed and interpreted from a series of loadings between 0 and 10 pounds. The next step for the embedded sensors would be to observe the sensor responses during gripping tasks of the robotic hand. One test was performed where the index finger of the robotic

hand exerted a pressure on a flat piece of wood to observe the sensor response. However, as the sensor began to have a response equivalent to a ½ pounds load, the robotic fingertip was unable to produce further torque, as the fishing line tendons slipped out of the set-screws. Also, tests were performed to attempt to hold an empty aluminum bottle, resulting in the hand being unable to successfully accomplish a grip of the bottle due to limited torque and range of motion in the fingertips. Despite this result, based on the response of the sensor to the experiment, the embedded sensors appear to be a suitable feedback method to determine forces applied at the distal ends of the robotic finger joints.

Chapter 3 : Robotic Hand Control

3.1 Method to Control The Robot

With the completion of a data glove and fingertip sensors, the framework becomes available to begin controlling the robot for semi-autonomous purposes. There are numerous methods for controlling such a system, whether through an external server, or with imbedded processing on the robot. There are several necessary steps in order to go from obtaining a voltage reading from the data glove's sensors to controlling the robotic hand. First, the computer must obtain and parse a string of values from the Maple board corresponding to readings from the bend sensors, then take these values to relay an angular position interpreted from these values. These angular positions are then translated and relayed to the robotic hand,, which will then mimic these positions to give control using the data glove.

3.2 Deciding on a Controller

To fully analyze which system would be ideal to use to control the robotic system, several factors should be considered, including expected delays in the system, compatibility with hand and glove microcontrollers, size and processing requirements of the controller, familiarity of underlying programs, costs, and assessing university resources.

One option is to control the robotic hand directly from the Maple Board. This would be possible by using the Tx and Rx pins to simulate a USB port, which would be used by the robotic hand. This option has the benefit of requiring a bare minimum of resources, as it would not require a computer to act as a client to the processes. However,

using purely embedded code on these two microcontrollers would result in a lack of an GUI(Graphical User Interface), which significantly add to the user friendliness of the program. Without a graphic user interface, it becomes difficult to troubleshoot errors, particularly when determining a failure from either the glove or robotic hand. Furthermore, a certain degree of user interaction will be needed with the system in order to calibrate the data glove. The Maple Board has only two buttons on the board, which makes the inputs for calibration difficult.

Another alternative is to design custom code using java or C++ to take advantage of a computer to act as a server between the Maple board and the Pololu servo board. This would offer the ideal combination of both user friendliness as well as adaptability to various needs for the program, while also having low costs for the user, as the program would be made in house, and not require a third party program which can be expensive. The primary drawback to this option is that extensive efforts will be required to develop and test a robust computational program.

The third option is to use a commercial program in order to act as a server between the Pololu and Maple Board. This can be accomplished through programs such as Python, Labview, Matlab, or XCOM. On the whole, obtaining licenses to these programs is expensive, though Matlab and Labview are readily available to Virginia Tech. In fact, a large amount of robotics work at Virginia Tech, both between classes and research is performed using Labview or Matlab. As a result, there is a wealth of knowledge and experience at Virginia Tech pertaining to Labview, which makes it an excellent program to work with. Furthermore, Labview is well known for the user-friendliness of its GUI as well as programming language. It is also very extensive, and

allows for in-depth programming and customization without requiring writing an entirely custom code. In particular, Labview is commonly used to read serial ports, as well as to parse strings of data and apply post processing to achieve desired objects. As a result, Labview appears to be an ideal choice to design the program architecture around. As stated before a primary drawback of this language is that the license is expensive, and is not readily available to the wider world. However, for the early phase of this project and prototyping within a lab at Virginia Tech, it is an ideal starting point. In the future of the project, it will be necessary to develop a custom code for find a third party code that is able to accomplish the same objectives in order to make this product commercially available.

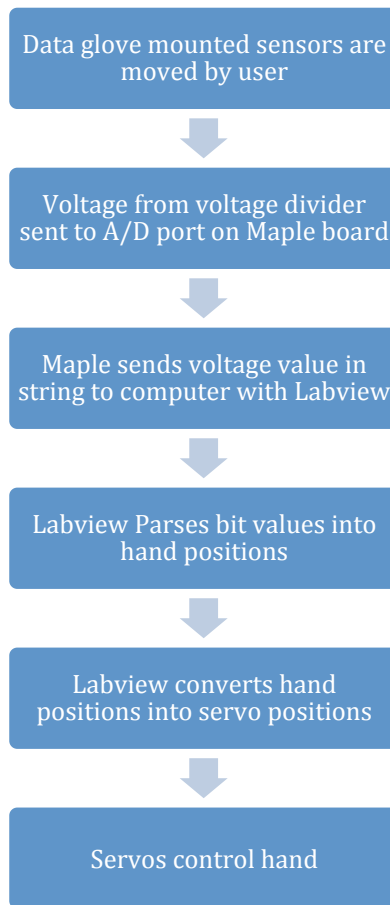


Figure 3-1: Simple flow chart explaining the process of obtaining, parsing and using signals to control the robotic hand.

3.3 Reading Signals from the Glove

Chapter 1 detailed the methods for creating a data glove to track 10 degrees of freedom were established, along with the necessary steps to take the resistance values from each individual sensor and obtain an analog voltage value, which can then be converted into a digital value on the Maple microcontroller. The maple accomplishes this using a 12-bit A/D converter. However, these signals must first be read and parsed using code embedded on the Maple. Because the Maple uses Arduino based code, it is necessary to use serial prints to send the data from the Maple to the PC. This is a fairly

basic command for a microcontroller to read an analog value, which will then send a digital value as a 12-bit digit between 0 and 4095. However, for a serial port to receive this information, all 10 values must be sent in a sensible arrangement that can be consistently read by a PC. Furthermore, the system needs to be capable of accounting for dropped bits, in the event of miscommunication over the serial port. To combat this problem, a simple system of headers and start and bytes was selected.

3.4 Parsing the Data Glove

Using a system of start and stop bytes is reasonably simple in concept. By sending a unique value which is unlikely to be replicated by the 12 bit voltage values being sent, and can be interpreted by the PC to a format that can be used to isolate the values. Without the usage of start and stop bits, the voltage values would be received as a long line of numbers, with no noticeable division between the values from different sensors. While some other parsing methods would involve using known bytes sent to determine the values that correlate to each sensor, this would not work for these values, since the values can range anywhere from a single digit to four digits, making tracking the values all but impossible.

The code begins each string with “A”, which indicates that the current code being sent is a newly sampled set of voltage data. The next value that is sent is an “a”, which indicates that the following digits are from the first sensor. After those digits, a “b” is sent, which indicates both the beginning on the next sensor, as well as the end of the first sensor. This process finds the digits between each letter, giving a series of parsed values. At the end of each string is a “KK”, which indicates the end of the given string, preparing the PC to receive the next string.

To give an example of this code implementation, a set of hypothetical values to show how the values are parsed and read from the PC are provided.

Sensor	12-bit Value
Sensor 1	2234
Sensor 2	1123
Sensor 3	132
Sensor 4	545
Sensor 5	3601
Sensor 6	3035
Sensor 7	804
Sensor 8	52
Sensor 9	1589
Sensor 10	2501

Table 3-1: Table of theoretical 12-bit values to be observed from the analog voltage ports.

With these values, the program on the PC would see the following string:

a2234b1123c132d545e3601f3035g804h52i1589j2501k

From these values, the program would notice the first value of the string is an “a”. The program would also notice that the sixth value in the string is a “b”. Based on this, the program can remove the second, third, fourth and fifth values to construct the 12-bit value for sensor 1. This also works for values that are not 4 digits, such as for sensors 3

or 8. In these instances, the code would recognize the 11th value as “d”, and the 15th value as “d”, allowing the code to observe the 3 digits of sensor 3 without issue.

However, there are other issues to receiving the signal. While ideally the string will be received in its entirety, this is unfortunately not always a reality. A common issue is that a string will be observed with only part of the values received, while the remainder of the string is sent in a separate package, typically either at the beginning of the next string. As a result, it is very possible for these errors to compound, and result in a fully unreadable system. To account for this, some small changes are made to the code. An “A” is added to the beginning of the code, and the “k” at the end of the code is replaced with a “KK”. With the inclusion of these two values, the Labview code receives the string, and first searches for an “Aa” as well as a “KK”, which would indicate that a full sensor reading is available. If an “Aa” is detected, but not a “KK”, the current string is placed back through a loop, and is concatenated onto the beginning of the next package, and will occur until a “KK” is observed at the end of the string. Below is an example of the code in operation, using the previous examples of sensor readings.

Here is an initial hypothetical reading:

Aa2234b1123c132d54

It is clear that the code cuts off halfway along the 4th sensor’s reading. The code recognizes that no “KK” is present, so the code is sent back through the loop to be concatenated on the beginning of the next string. The code receives the next string:

5e3601f3035g804

This is then concatenated onto the end of the previously received string, giving:

Aa2234b1123c132d545e3601f3035g804

Once again, no “KK” is present, so the code sends the string back through, and receives this package:

h52i1589j2501KK

After concatenation, becomes:

Aa2234b1123c132d545e3601f3035g804h52i1589j2501KK

Because “KK” is now present, the code will parse out the individual values as shown above.

3.5 Labview Code

Having obtained an array of 12-bit values from the Maple board, it is necessary to find a way to translate these values into an angular position that is interpreted from the user’s hand. The previous chapter established that while a linear fit is not a wholly accurate approximation of the behavior of the bend sensors with changes in finger position, general inaccuracies in the bend sensors mean that it is a suitable fit. As such, two points are needed to create a linear fit, which can be most easily accomplished by using the start and end position, which would correlate to 0 and 90 degrees of flexion. While a true humans hand has variations in degrees of flexion, 0 and 90 act as an approximation. This convention breaks each degree of freedom into a fully flexed(90) and fully extended position(0). Because the data glove encompasses proximal and distal flexion of both the fingers and thumb for a total of 10 degrees of freedom, it would not be ideal to calibrate each individual position. Instead, there will be a series of hand poses, which will be used to determine the fully flexed and extended positions. The initial hand

pose is the whole hand fully extended, also called “Neutral”, which acts as the first value for both the proximal and distal sensors for the fingers and thumb. In determining the additional poses, it is important to select sensors whose motion will not impede or otherwise affect another sensor that is being recorded. With this in mind, the next pose is titled “Distal Flex”, in which the index, middle, ring and pinky distal medial joints are flexed, and the values from the corresponding sensors are recorded. This is repeated for “Proximal flex”, in which the four fingers are flexed from the proximal joint. Because full range of motion in the thumb can impede the range of motion of other fingers, these portions were split into another set of poses, called “Thumb Distal” and “Thumb Proximal”. For each of these poses, a simple case structure with the desired sensors is detailed, allowing for a drop-down menu for poses to be selected. A button titled “Calibrate” commands Labview to save the calibration values for the corresponding sensors depending on the drop down menu option that has been selected. This allows the calibration options to be performed in any order, and allows for easy re-calibration.

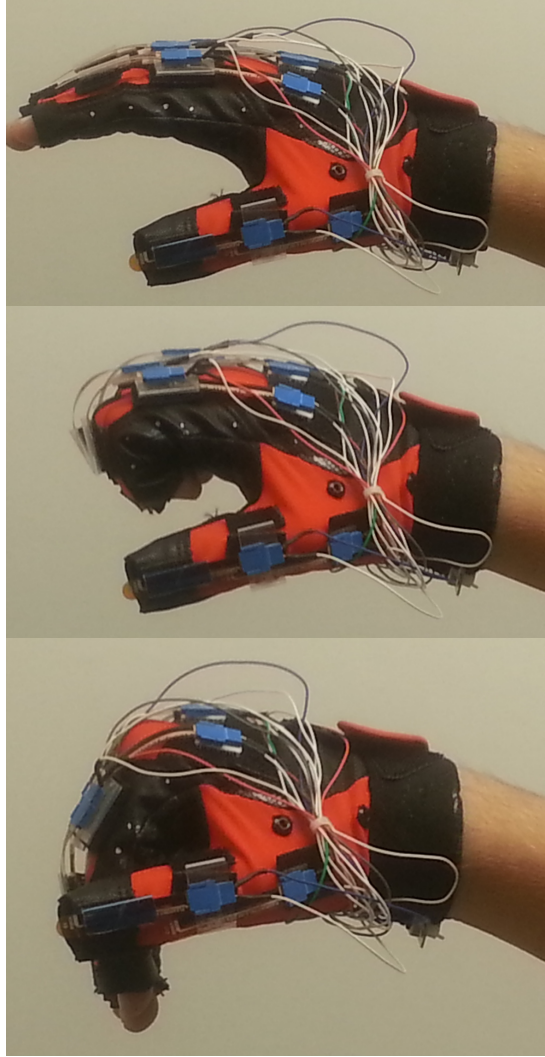


Figure 3-2: Neutral Hand Position(top), distal finger flexion(middle), proximal finger flexion(bottom)

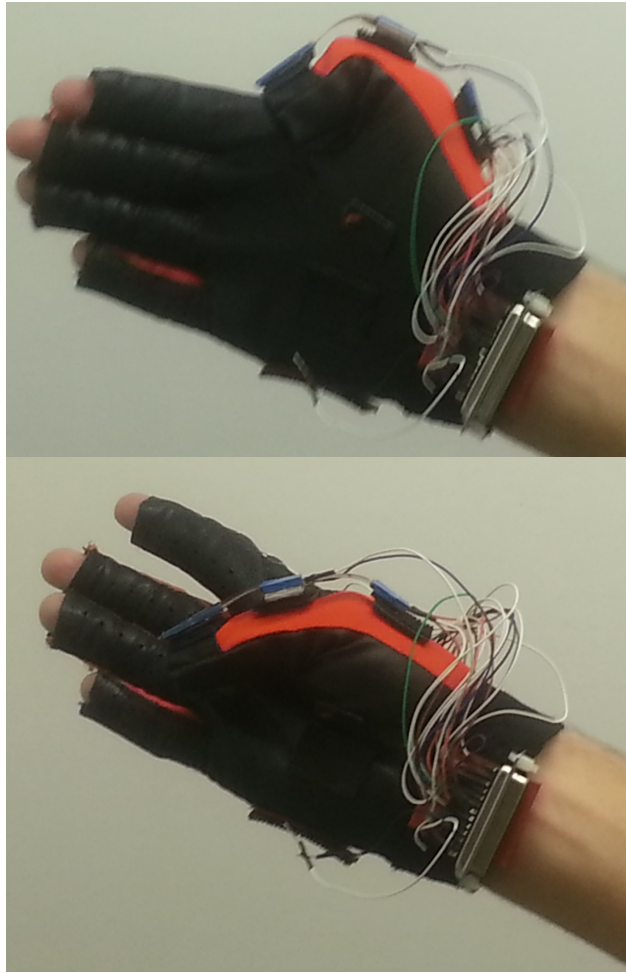


Figure 3-3: Thumb distal flexion(top), thumb proximal flexion(bottom).

Upon full calibration, Labview now has a 2x10 matrix of 12-bit integers, with one column being the “Neutral “ values, and the other being the corresponding values from the flexed position calibrations. With these values, a linear calibration will translate the values from the maple board into a value between 0 and 90, which will correlate to a flexed and extended set of positions.

$$P = \frac{90}{V_f - V_e}(x - V_f) \quad (3.1)$$

P = Position

V_f = Flexed bit value

V_e = Extended bit value

x = Current sensor bit value

However, noise and variation in the sensor resistances can result in values exceeding the calibration limits, so negative values and values over 90 are possible. In order to prevent these values, a simple case structure is added which takes any value over 90 and replaces it with 90, and takes any value beneath 0 and replaces it with 0. Once these calibrations are complete, a series of bars on the front panel will allow the user to see clearly the output of the data glove following these conversion, while a constant readout of the bit values coming from the hand prior to conversion allow for additional troubleshooting. Once the calibration is complete, the user can push a button titled “Control” on the front panel, which will initialize control of the robotic hand. With this control setup, the user is able to determine whether or not the data glove is sending the correct information to the robotic hand prior to actually controlling the hand, reducing risks of sending unexpected or erratic values to the robotic hand and causing potential damage to the motors or joints.

Extended Bit Value	Flexed Bit Value	Bit Value in	Translated	With Stopper
2632	1687	2011	59.14	59.14
2985	1917	2556	36.15	36.15
2500	651	1232	61.72	61.72
1526	752	702	95.81	90
2125	887	2021	7.56	7.56
2118	917	2098	1.50	1.5

2226	1185	2326	-8.65	0
2359	821	2278	4.74	4.74
2265	966	1203	73.58	73.58
2011	455	1956	3.18	3.18

Table 3-2: Series of flexed and extended 12-bit values, observed bit values from the maple, translated values from governing equations, and values after stopper program.

The robotic hand that is being controlled operates using a Pololu mini-Maestro 18, which receives 8-bit values for servo positions over serial communication, and then relays them to the attached servos. This communication can be easily accomplished using a min-USB port on the hand, though it could also be done through Tx Rx pins on the board. However, because a computer with Labview acts as a server, the resources to use a direct USB port is preferred, since Labview already has the communication protocol for USB ports well documented.

By opening a COM port, it is very simple to send a position to any of the given servos. The motors are limited to a set range of 8-bit values, which operate within a 12-bit series of limits that is programmed into the Pololu board. This is typically done using Maestro Control center, which uses sliders to send positions to the servos, but also allows an operator to program in limits to each servo. The servos have large range of motion, but only a portion of this is needed for controlling the robotic hand. In fact, using the full range of motion within the motors causes the servos to draw excessive amounts of current at the limits of the servo range. By setting the limits to the mid-range of the servos, power draw is lowered in the servos. Furthermore, by limiting the servos to the desired operation range, the motors can be more sensitive. The Pololu board receives the servo position as an 8-bit value, and then assigns a servo position along the predetermined

range of 12-bit values. Also, the robot's joint flexion depends on the servo orientation, so some servos cause flexion with a positive servo position, while other servos create flexion with negative servo positions.

The next step is to convert the values from the 0-90 flexion values to 8-bit values to be received by the robotic hand. The Pololu receives a servo position as part of a string that includes a start byte of FF, a hexadecimal servo number, and a hexadecimal 8-bit number. If a value exceeding 255 is received, the servo board will become overbuffered, and will need to be reset. To avoid this, the 0-90 value will be converted to a value between 0 and 254 to control the robotic hand. Depending on which direction the servo must go to create flexion, the equation to convert the flexion degree will be slightly different.

If 254 is full flexion:

$$S = \frac{254}{90}(P) \quad (3.2)$$

If 0 is full flexion:

$$S = -\frac{254}{90}(P) + 254 \quad (3.3)$$

S = 8-bit servo value

P = 12-bit sensor value

This full system allows usage of the data glove to control the robotic hand. However, there have also been instances of failures in the data glove, namely when

sensors become detached, or in cases where the motion of the sensor may be impeded. To deal with these issues, the front panel of the controller can switch the individual degrees of freedom of the robotic hand to control through a series of Labview sliders. This also allows a user to produce hand poses that are typically difficult to produce or otherwise maintain. Furthermore, several degrees of freedom that are not tracked by the data glove can be controlled using this method, including thumb rotation, pinky abduction and index abduction.

3.6 Future Work

While the current data glove controller appears to be effective, there are numerous improvements that could be implemented. Namely, being able to perform network operation with the data glove and a remote computer would be ideal. This was the initial proposed objective, as it would act in line with the original goal of the robotic hand, which was to fully perform tasks remotely, with the user able to perform the task from miles away. However, for the purposes of troubleshooting, it was ideal to remain in close vicinity to the robotic hand, as delays from sending the control data from sizable distances would make controlling the robotic hand difficult. Though it has not been implemented yet, TCP/IP communication is still very viable, as Labview has a client/server program.

Also, it would be useful to be able to include readings from the fingertip sensors as part of the Labview code. Because of the robustness of the Labview front panel, adding code to receive and parse the signals from the fingertip sensors would be very similar to the sensors from the data glove, and allow for additional forms of feedback to a user. As this code would develop, semi-autonomous control could also become

developed, such as setting in force limits on the fingertips, or creating routines to maintain a grip on an object without input from a user. However, until the robotic hand becomes robust enough to grip objects, this coding would be difficult to effectively design without the opportunity to observe the behavior of the robotic hand with fingertip sensors attached.

Chapter 4 : Brain Computer Interface Application

4.1 Beyond Data Glove Controllers

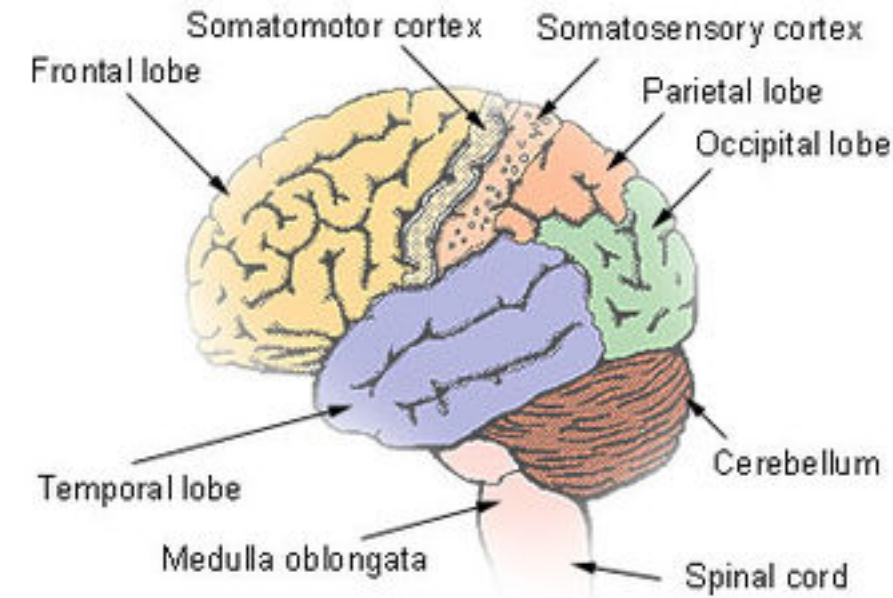
While the methods to control robotic hands detailed in the previous chapters have proven to be effective, further advances can be made by using neural interfaces. As detailed in the beginning of the data glove chapter, there are many methods to finding a multi-degree of freedom controller that can be used to control a robotic hand. To really push the limits of human-machine interface, alternative methods for control input should be considered.

Of particular interest are methods that utilize the human biology to control robotics as naturally as a human controls a limb. While a data glove is a good option to accomplish this goal, control can also be accomplished by obtaining signals directly from a user's muscles using Electromyography(EMG). This method involves using a series of electrodes that observe electric membrane potentials created in a human's muscles during contraction, which in turn allows for direct observation of a subject's muscle contraction, typically for experiments involving biomechanics (Soderberg & Cook, 1984). EMG can come in two forms, surface EMG and intramuscular. In surface EMG, leads are placed onto the surface of a subject's skin, just above the muscle being observed. This typically works better with larger muscles, as it is very susceptible to cross talk, which is where action potentials in other muscles interfere with the expected signal. Another option with EMG is intramuscular EMG, where small gold wires are surgically attached directly to the muscle fibers. This method, while slightly invasive, allows for excellent readings from muscles with limited cross talk. Using leads attached into an amputee's forearm or

pectoral, an amputee can learn to control various upper limb positions using slight muscle activations and movements (Englehart & B., 2003).

4.2 Brain Computer Interfaces

In this chapter, an attempt was made towards directly utilizing the user's brain, in a concept called Brain Computer Interface (BCI). There are numerous purposes for BCI studies, such as neural prostheses for hearing (Gantz, Turner, Gfeller, & Lowder, 2005), seeing, and even for robotic arms (Velliste & al., Neuronal Ensemble Control of Prosthetic Devices by a Human with Tetraplegia, 2008). While the subject seems to be from a science fiction source, some BCIs have become a relatively common occurrence, such as in Cochlear Implants for certain hearing impaired individuals. BCI work has been a popular topic, as it offers exciting insights into human brain function, while also opening new avenues and concepts for how the brain can be used as a direct input into various programs (Haas, 2003).



Lobes of the cerebrum

Figure 4-1: Locations of various brain functions. Of interest to this project is the Somatomotor cortex, also called the motor cortex, which controls a majority of muscle movement. National Cancer Institute. (2013). *Brain*. Retrieved from National Cancer Institute: Seer Training Modules:

http://training.seer.cancer.gov/images/brain/cerebrum_lobes.jpg

To perform Brain Computer Interface studies, first there must be a method to observe brain signals in some capacity. This can be done both with invasive as well as noninvasive methods. Invasive methods involve cutting into the skull of a patient in order to implant a device, whether a series of electrodes or a standalone chip, to take readings directly from brain tissue. These methods, while notably difficult as well as potentially dangerous, offer some of the best data in terms of both spatial and temporal resolution. Noninvasive methods do not require surgery or other techniques that involve cutting in order to obtain readings, at a cost of data resolution. Some of these methods will be discussed here with respect to robotics control.

4.3 Non-Invasive BCI Methods

As early as the 19th century, scientists have observed a correlation between brain activity and electrical signals that are observable from the scalp of a subject. In 1924, Hans Berger took some of the first electrical signals from a human subject to observe brain activity (Haas, 2003). These early studies led to the establishment of Electroencephalography (EEG). This methodology involves using small electrodes planted directly to the scalp of a subject to observe small fluctuations in electrical activity in the brain as relayed through the surface of the subject's skin. In typical clinical applications, 21 electrodes, including ground and reference, are attached to a patient's scalp according to a standardized 10-20 system. Clinically, this system has applications for determining brain death, characterize seizures, and monitor brain activity for patients in comas, either induced or non-induced. Compared to other options for observing brain signals, EEG is relatively inexpensive, does not require a large amount of additional technology to maintain the system such as coolants or power, and has good temporal resolution. On the other hand, the data has a poor signal to noise ratio, as well as poor spatial resolution. In addition to these issues, EEG can only record surface brain signals, while deeper brain signals come with significant noise (Wolpaw & al., 2002).



Figure 4-2: Subject with high density EEG electrode array cap. Myers, D. (2010).

Wikipedia. Retrieved 2013, from Electroencephalography:

http://upload.wikimedia.org/wikipedia/commons/b/bf/EEG_cap.jpg

The mechanic for observing brain signals based on electrical pulses created by brain activity can be measured by another method, called Magnetoencephalography (MEG). This method measures electrical signals produced within the brain, but unlike EEG, MEG observes brain activity by observing the changes in magnetic fields due to electric currents on the order of 10 fTesla. This method has both excellent temporal and spatial resolution for surface brain signals, allowing for revelation of high details of brain function (Cohen & Halgren, 2004). Because magnetic fields decay exponentially with distance, this method only works for brain activity close to the skull, as other signals cannot be observed. However, this method is very sensitive to external noise, and must be

performed in a controlled room with strong magnetic shielding, while the subject must remain entirely immobile (Vrba & Robinson, 2001). Also, because the magnetic field produced by brain activity is so small, the SQUID(superconducting quantum interference device) sensors require significant cooling, needing a vacuum tube of liquid helium to obtain suitable cooling (Vrba & Robinson, 2001). Another important issue is that MEG sensing is subject to an Inverse Problem, which states that any given series of signals can have an infinite number of possible solutions for their source and magnitude within the brain. Even with a sizable array of sensors along a subject's brain, the sensors cannot fully determine the exact location and strength of each magnetic signal. For example, a single reading from a sensor could suggest a near source with a weak field, or a further source with a strong field, which would be virtually indistinguishable between one another without a planned theoretical model (Uutela, Hamalainen, & Somersalo, 1999).



Figure 4-3: A subject using a MEG interface. Note the large device size and cooling necessary to use this method. National Institute of Mental Health. (2009).

***Magnetoencephalography.* Retrieved 2013, from Wikipedia:**

http://upload.wikimedia.org/wikipedia/commons/thumb/e/e6/NIMH_MEG.jpg/398px-NIMH_MEG.jpg

Another method to recording brain activity is using functional Magnetic Resonance Imaging. This method uses a high power magnet on the order of 1 tesla to obtain detailed imagery of the brain itself by magnetizing a subject's brain tissue. A gradient field is applied to the tissue and subsequently removed. The magnetized tissue releases energy that can be observed and used to construct images. Different tissues are

magnetized differently, which allows doctors to distinguish between different types of tissue and structures. This can be applied to arterial and venous blood flow, due to the differences in the blood oxygenation levels. Arterial blood is oxygen rich blood that is used to resupply oxygen in tissues, while venous blood is de-oxygenated after having been used within tissues. This fundamental difference of oxygenated versus de-oxygenated fluid allows fMRI to observe venous and arterial blood separately; de-oxygenated blood is more magnetic than oxygenated blood, allowing technicians to make distinctions between the two flows. As neurons are active, the tissue sees an influx of arterial blood, which is observable as a change of flow rate (Logothetis & al., 2001). The primary drawback of this method is that there is very low temporal resolution due to the physiology that is being observed, and is phenomenally expensive, costing on the order of millions of dollars. Additionally, because this method uses powerful magnets, users should not have any metal objects, which includes metal implants such as pacemakers, catheters, or joint pins or replacements (Lebedev & Nicolelis, 2006).

4.4 Invasive Methods

Another option to obtain brain signals is to implant electrodes directly into a patient's brain. This option, while notably dangerous due to necessitating brain surgery in order to implant the device into a patient's brain, offers unparalleled spatial resolution, as the device is placed close to the signal and has little noise.

One of these methods is called Electrocorticography (ECoG), which operates using the same concept as EEG, using a series of electrodes in an array attached just below the Dura Mater, in order to read action potentials between various points of the brain without interference from the scalp and skull. To place this array, a craniotomy is

necessary, where doctors remove a portion of the patient's skull in order to implant the electrode array to the surface of the cortex. This method is considered one of the prime options for evaluating severe epilepsy patients' conditions in preparation for surgery, due to the high signal to noise ratio along with high temporal resolution (Kuruvilla & Flink, 2003). However, this method is only employed in human patients in preparation for surgery, due to already needing to perform a craniotomy for the surgery. This method has been employed on rats (O'Keefe & Dostrovsky, 1971), as well as primates for brain computer interface studies. In one prime example, monkeys with an ECoG interface were given control of a robotic hand to obtain some desired treats with a joystick. However, after some time the monkeys began to realize the act of wishing to control the hand was sufficient to obtain the treats, producing a full BCI (Velliste, Perel, Spalding, Whitford, & Schwartz, 2008).

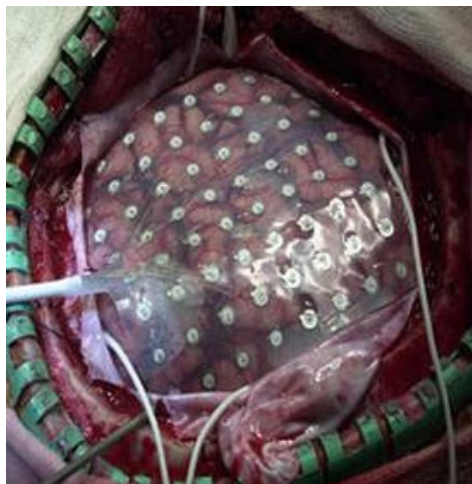


Figure 4-4: Electrocorticography array attached to skull of a patient prior to surgery(Future Predictions.com. (2011). *Mind Reading by ECoG as Brain Computer Interfaces Will Become Common*. Retrieved from Future Predictions.com:

<http://futurepredictions.com/wp-content/uploads/2011/05/ecog.jpg>). Used under fair use, 2013.

A far more invasive method that offers unparalleled spatial and temporal resolution is implanting microelectrodes directly into the brain of a subject. Compared to ECoG, this method is significantly more invasive, as the process involves penetrating the subject's grey matter with small electrodes, which can observe electric signals directly at the source, allowing incredible spatial resolution, such as observing the activity of a single neuron (Quiroga & al, 2005). Depending on the desired task, the electrode array is placed into the appropriate portion of the brain, which in the case of neuroprosthesis would be motor cortex. In some of the more recent research, it has been applied to patients with Locked in Syndrome, where the patient has little or no motor function below the neck, making simple tasks such as drinking or eating all but impossible without assistance from a caretaker, despite the patient having full cognitive awareness. In several studies from Brown, using one of these electrodes implanted into the head of several human trials, subjects have been able to successfully control multiple degrees of freedom robotic hands mounted to their wheelchairs to accomplish simple tasks (Hochber & al, 2006). However, because this method requires surgical procedures, it is considered a last-option. Numerous concerns exist, between complications during surgery, chip deterioration, or further complications due to lack of human testing. Thus this option is limited to all but patients with very severe disabilities, such as those with tetraplegia.

Method	Created	Advantages	Disadvantages
EEG	1924	Good temporal resolution, inexpensive	Poor spatial resolution, only able to observe surface signals at noise threshold
MEG	1968	Good temporal and spatial resolution	Unable to read deep brain signals, Inverse problem, subject must remain very still
fMRI	1990	Excellent spatial resolution, deep brain information	Low temporal resolution, Large magnets needed, subject must remain very still
ECoG	1950	Good spatial resolution, good temporal resolution	Involves partially invasive surgery
Micro-Electrode Array	1972	Excellent spatial and temporal resolution into deep brain signals	Requires intense brain surgery

Table 4-1: Table of advantages and disadvantages of various BCI methods.

4.5 BCI Studies in this thesis

Upon examining existing options for robotic control, several methods appear as particularly feasible. It is clear that invasive methods would not be investigated as part of

this experimentation, as the risk greatly outweighs the rewards of robotic control. While the concept of remotely controlling robotic hands is an appealing goal, few persons would be willing to risk long-term brain damage to accomplish this. As mentioned before, this technology has been used in quadriplegics, where the potential benefits are significantly greater than for an average, functional human. As such, only non-invasive options will be considered for this thesis.

One of the foremost options when choosing a non-invasive option to read brain signals is Electroencephalography(EEG). Because of the relatively simple setup, this is a popular method for brain computer interface, as it does not require prohibitively expensive equipment or medical operations, while also maintaining a high temporal resolution necessary for real-time control. EEG is a well-versed subject, which gives additional resources and references to draw from in order to accomplish control of a robotic hand using exclusively brain signals (Gernot & al, 2008). However, the low spatial resolution and mechanisms of EEG technology prevents researchers to obtain in-depth information about human brain function. While this research is applicable in some fields, further research using a different method that would yield more details about human brain function during BCI control would open many more doors into the learning techniques and practices to implement and control with BCIs.

This leaves fMRI and MEG as possible options to control the robotic hand. While MEG has the appeal of a high temporal resolution, the issue of not being able to read deep brain signals makes it a methodology on par with EEG. However, the high spatial resolution to the surface brain signals would make it preferable over EEG. This is a moot point though, because the facilities at Virginia Tech do not include an MEG scanner,

which requires a magnetically shielded room and a very powerful, super cooled magnet to be possible.

Virginia Tech Research Center at Carillion has a Neuroscience department with access to two 3 Tesla fMRI scanners, along with a very capable staff of programmers and technicians. This facility was utilized for the research conducted in this thesis.

4.6 Experimental Setup

For control of the robotic hand, it is important to find a robust model of brain activity that will allow for a new user with little brain computer interface skills to control the robotic hand. It has been noted that BCI work is a skill for an operator, and must be honed and developed to demonstrate complex controls. As such, the user will be using simple left and right hand activity in order to clearly draw a distinction between two different states. Because the left and right hand are controlled by two different sides of the brain, this will allow for a very clear distinction between either left brain or right brain activation, which in turn will allow for early phase control of the robotic hand.

The Carillion lab has previously performed experiments that train left and right brain activation along with left and right hand usage, so a previously developed code will be used in order to train an individual and build a mathematical model around their brain activation. To develop this model, the subject is given cues from within the fMRI to press some kind of controller with either their left or right hand, while the blood flow in the subject's brain is observed. By plotting the expected response with the actual response, after several repetitions the program is able to recognize a clear pattern between the left and right motor cortex activation. As the program observes this distinction, the written program will give increasingly positive values for right side activation, and negative

values for left side activation. This way, large positive values indicate strong right side brain activity, while large negative values indicate strong left side brain activity.

These values are observed from a Linux machine running python, which must be sent to the robotic hand to achieve control. It is relatively simple perform serial communication between the Linux machine and an external Windows machine using a USB to RS-232 Serial interface, which will be used until distance TCP/IP communication is established. As mentioned in chapter 3, it is significantly easier to control the robotic hand using programs that are familiar, so this experiment will use the Labview running a modified control program from the previously described data glove program. This code will move the robotic hand between three different hand positions: a neutral position, a American Sign Language(ASL) A, and an ASL B. Using a case structure, when the code receives an 8-bit value between 0 and 2, the robotic hand responds with a corresponding preprogrammed robotic hand position. However, because the Python code operates using positive and negative values, the Python code was also modified to send the appropriate values to the robotic hand. For all positive values, a 1 was sent to the robotic hand, while for negative values, a 2 was sent to the robotic hand, while a value of 0 was cycled in between these values in order to ensure the hand was moving from neutral into the desired hand position.

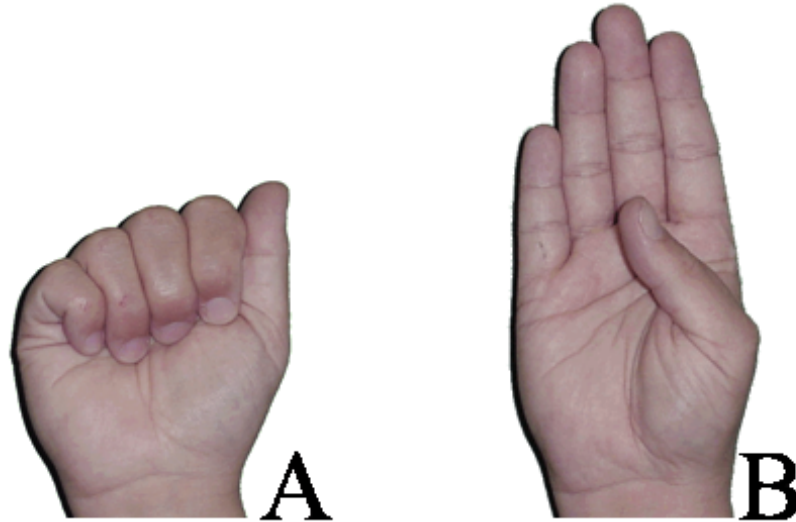


Figure 4-5: American Sign Language A(left) and B(right) which are imitated by the robotic hand during different stages of brain activity.

4.7 Experimental Process and Results

To obtain the necessary data, a subject was placed within the fMRI with no metal. The person was laid on a table with a magnetic coil around their face to help get higher resolution images of the brain. Due to the loud noises of the active scanner, the subject also wears ear protection for the full duration of the test. The subject was also given two fiber-optic button boxes, one for each hand, which can be used to transmit button press information without using metal wires. The room is then pressure sealed, while the controls operators work in an adjoining room that is magnetically shielded. The fMRI operators can then speak to the subject in order to give directions through speakers in the room, and also receive voice from the subject through pressure microphones.



Figure 4-6: A subject being prepared for fMRI readings. Pictures for this must be taken from a distance, in order to prevent the magnetic field from destroying the camera.



Figure 4-7: High-resolution image of the subject's brain.

The subject then uses a mirror placed on the head apparatus to view a computer screen that is projected onto the back of the fMRI, allowing the subject to view visual cues for the experiment, as well as view the evoked potentials they are projecting to better track their progress. Once the user is fully situated, a 5 minute long scan takes place to obtain a high resolution image of the subject's brain, giving a resolution of about 1 mm^3 of the user's brain in a series of slices, which is used to overlay the lower resolution blood flow images. Once this is complete, the subject begins conditioning the brain to respond to set stimuli paired with specific motor function. In this case, the subject will be cued to press the buttons in either the left or right hand, which will allow for the Python program to build a model to which portions of the brain are activating.

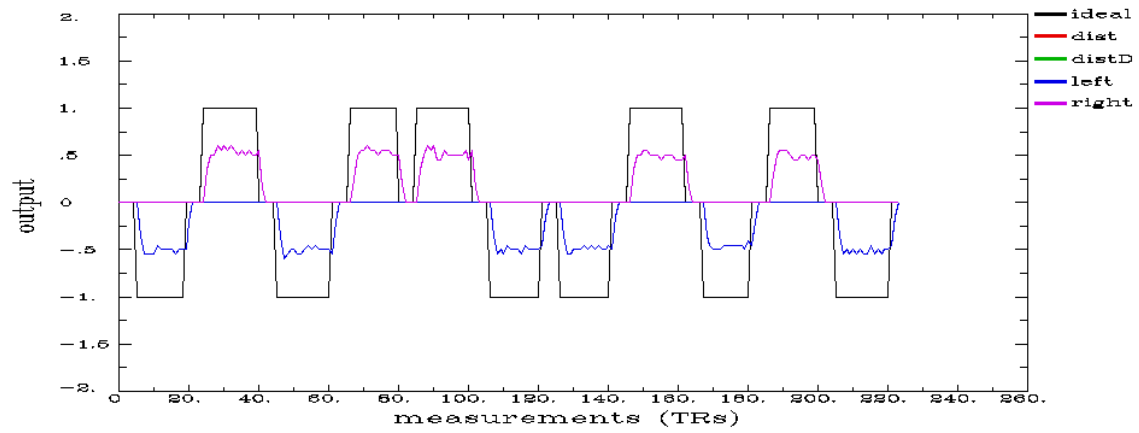


Figure 4-8: Plot showing the training phase of the experiment. The black lines indicate the ideal response, while the blue lines indicate left hand button presses and pink shows right hand button presses.

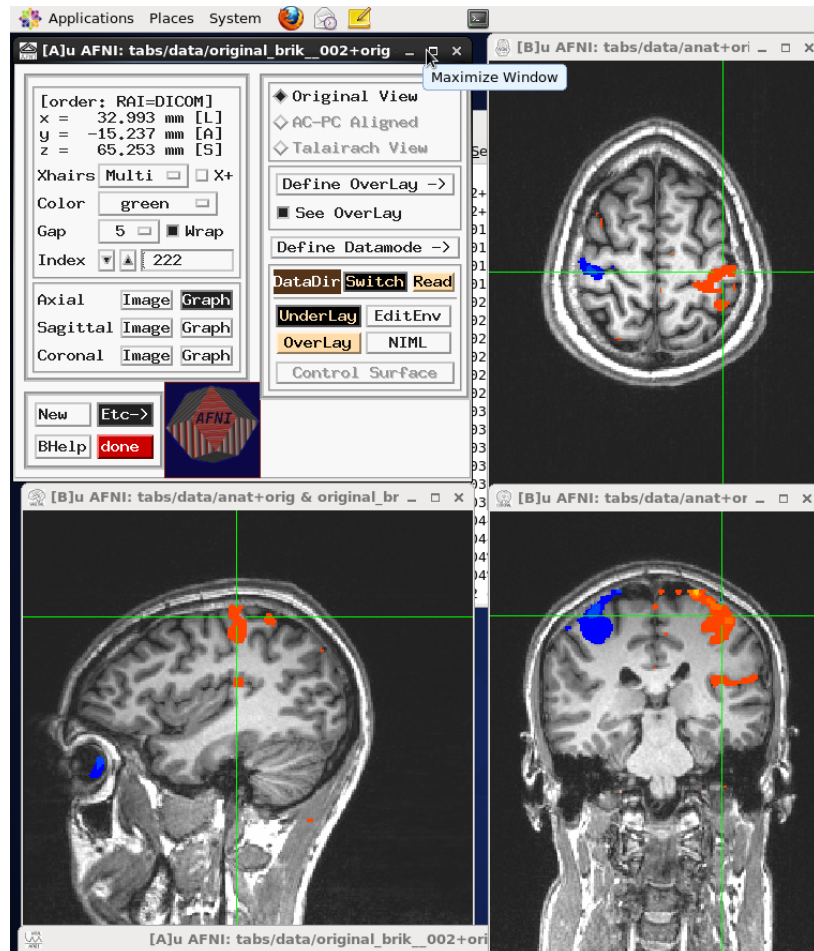


Figure 4-9: Image showing subject's brain, specifically blood flow during gripping action of the right hand.

Once a suitable model has been built, the robotic hand can be controlled using evoked potentials. To build a robust model, the subject is still given the same cues, and continues to press the same buttons. However, the buttons no longer have any link to the computer, and the end result for robotic hand control is the user's brain. The user receives the same cues as before, being told to press either the left or right buttons, though now a slider bar shows the results of the brain activity, sliding further to the left for highly negative values, and to the right for highly positive values. This can be seen in the figure below, which plots the expected values in red against the actual values in black. A

camera feed of the robotic hand is also shown to the user, which allows the subject to confirm that the robotic hand is giving the appropriate response.

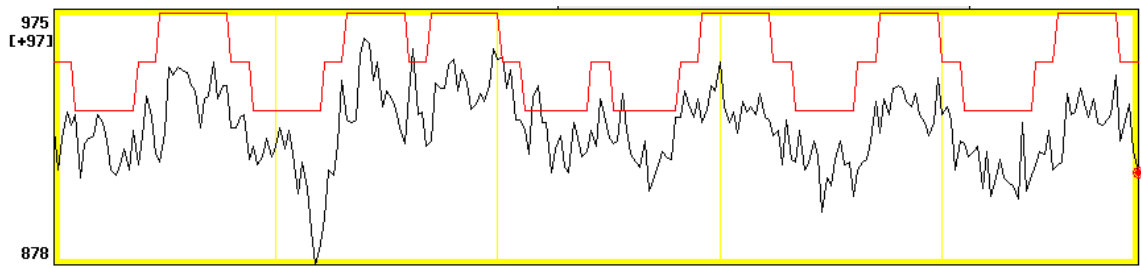


Figure 4-10: Image showing the ideal response of the brain in red, and the actual response of the brain in black, which is seen trending with the ideal response.

The results are as expected, showing consistent activation for the contralateral motor cortex for activity with each hand, allowing for successful control of the robotic hand using evoked potentials. However, the expected delay was also present, creating a 2 second delay between the changes of state for the robotic hand control following a change of cues to the subject. Despite this, the created model was sufficient to control a robotic hand, which was the primary purpose of this experiment.

While there is a general trend of the evoked potential following the ideal response, there was significant noise in the signal compared to an ideal response. This is indicative of the subject being a novice with brain computer interfaces, and would be expected to improve with additional training.

This method was effective, but it is important to acknowledge that using fMRI as an industry standard of BCI work is not feasible. In particular, any BCI subject must remain very still during testing. Significant movement will cause data to become skewed, as the brain will shift relative to the expected model, causing blood flow in other sections the brain to be incorrectly attributed to the trained model. Furthermore, it takes at least 2

seconds for a subject's blood flow to indicate a noticeable evoked potential, which would make a control system relying entirely around fMRI input very ineffective. Regardless of the improvements in fMRI technology, this time delay cannot be improved, as the limitation is caused by the physiology of the action. However, a major benefit to using the fMRI is to further the understanding of the human brain, particularly with respect to learning the use of an external machine-brain interface. In the next section, I will discuss the potential benefits of following this method of Brain Computer Interface.

4.8 Future Applications of Research

While proving that an operator can control a robotic hand using only their brain from an fMRI is commendable and appealing to a wide audience, this has been known, as brain activation related to gripping and other large motor tasks have been examined before (Lee, Ryu, Jolesz, Cho, & Yoo, 2009). However, by utilizing this early simple test set up, researchers the VTCRI are able to begin to open doors into very interesting questions.

While currently the controls are very simplistic, providing either one hand position or the other, future models and training may allow for a user to begin forming additional, more complicated motions. For example, instead of making one hand position or the other, trained users may be able to consistently control individual fingers, or even develop their own hand posturing to apply to the robotic hand. As familiarity builds between the subject and the hand, researchers may be able to observe signs of phantom limb syndrome, where the subject associates physical stimuli with specific forces applied on an object.

One of the most intriguing applications is finding more complex control algorithms to apply in tandem with brain control. For example, using a semi-autonomous control system that allows the robotic arm to perform various tasks at the user operation. With this system, a user could use even an abstract idea in their head, rather than a direct motor cortex interpretation, to perform a variety of tasks. What is particularly exciting about this concept is how this methodology would mimic a human controlling their limbs through the peripheral nervous system. While the main control center for a human is the brain, a large amount of the tasks that are performed are hardwired into the peripheral nervous system. Take, for example, flipping a coin in the air. A human wants to do this task, and takes some time to learn to consistently flip the coin. However, once the task is trained, the majority of the task is known subconsciously through the peripheral nervous system. Small details of the task, such as sensory feedback of the coin's position, suitable muscle torques to accomplish desired joint velocity, and hand position during the follow through to catch the coin are all trained, and then delegated to the peripheral nervous system. By emulating this concept, a human operator could plan a desired task, while the robot will be pre-programmed with a variety of tasks. While the human operator builds up the brain potential to select a desired task, the robot could perform autonomously, or wait for additional feedback from the user. This concept could also provide interesting insights on human-machine learning. For example, a human learning to perform a task in tandem with a machine that is also learning how to perform the task could provide an interesting dynamic for cooperative human-machine learning, as well as understanding the process for a subject to become familiar with BCI work. This research could be used

to provide better machine learning algorithms, as well as for developing methods for teaching and maintaining BCI skills.

Chapter 5 : Summary and Conclusions

Over the course of this thesis, the methods to implement a biomimetic approach for controlling a robotic hand have been developed through the creation and implementation of a data glove, grip feedback sensors, and through brain computer interface controls. All the characterization was conducted on a 3D printed robotic hand. The data glove was designed using commercial sensors and showed suitable sensitivity and linearity for control purposes while costing only a fraction of other commercially available data gloves. Despite some issues with linearity of the data across all degrees of freedom, the current data glove has an excellent potential towards becoming an useful product. The robotic grip sensors had success in exhibiting excellent linearity and resolution for a suitable range of applied forces while also providing an esthetically pleasing and effective silicone skin that imitates the human skin in some degrees. Also, this thesis described some of the early steps taken to implement the new brain computer interface. The results showed a preliminary success in controlling a robotic hand using a subject's brain.

In future studies, there are improvements that can be made for each of the topics. For the data glove, additional degrees of freedom would help to shorten the gap between the capabilities of commercial data gloves. The robotic hand feedback has not been tested in a practical setting due to the need for a redesign of the robotic hand. However, once this redesign is complete, there is a very exciting realm of possibilities for new control schemes and methods that present themselves for a robotic hand with both user input as well as feedback on gripping strength for the user. Using brain computer interface we can begin to search for new and exciting applications by beginning to train subjects over a

long period to develop acute skills to produce more hand positions and begin to perform teleoperation experiments with their brains alone. These new control methods will eventually bridge the gap between human and robotic function allowing for complex teleoperative tasks.

Appendix A: Molding Process



Figure A- 1: Negative molds(left) with .75 scale and .5 scale positive mold.

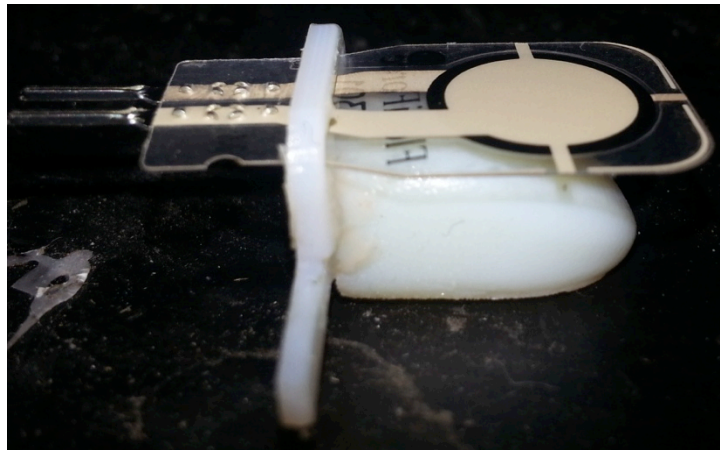


Figure A- 2: Positive mold with flexiforce sensor fitted into slot.

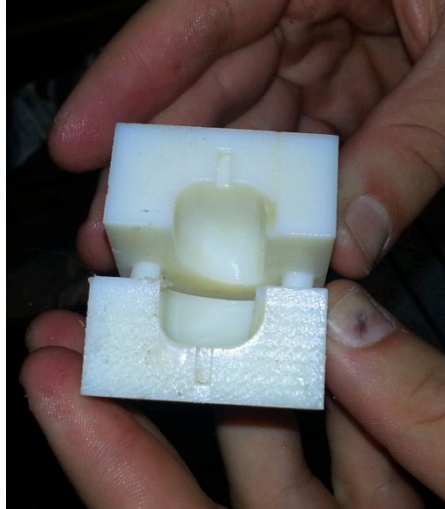


Figure A- 3: Left and right negative mold attached using guide holes.

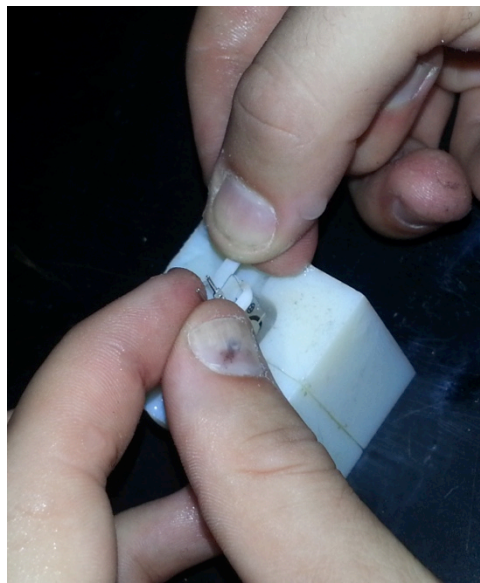


Figure A- 4: Positive mold a sensor placed into silicone in full negative mold.



Figure A- 5: Full mold assembly clamped to cure.

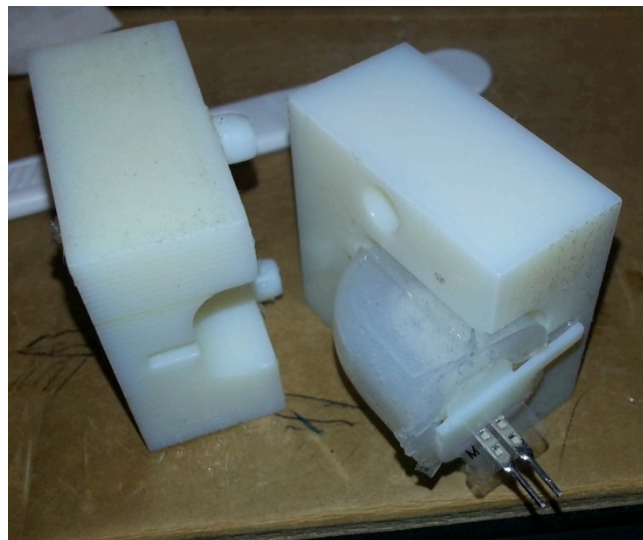


Figure A- 6: Pulling the left and right mold apart.

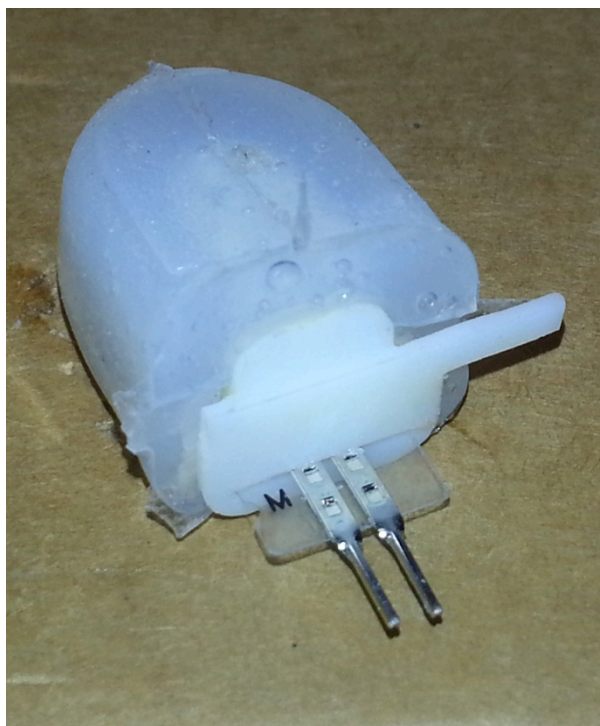


Figure A- 7: Silicone mold with sensor and positive mold inside.

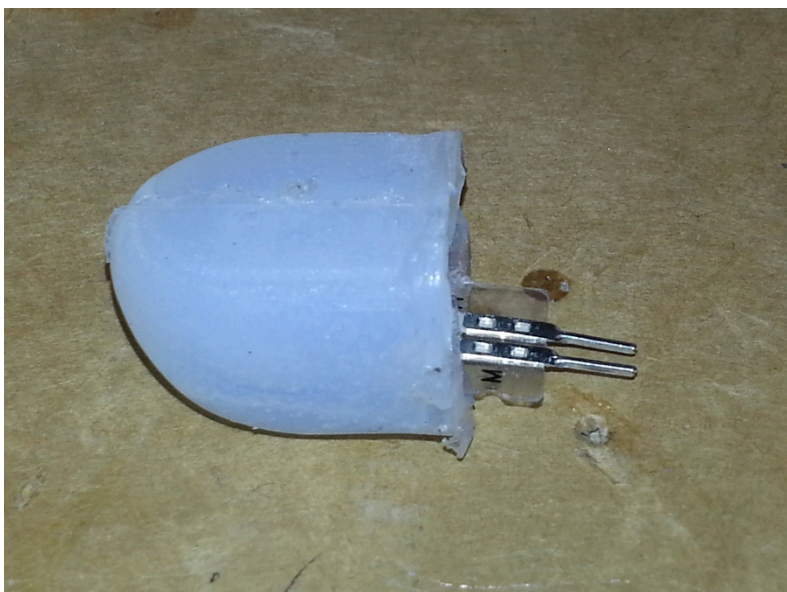


Figure A- 8: Final silicone skin with only the sensor inside.

Appendix B: Hand Control Code

```
int analogPin = 1; // Ind Prox

int analogPin2 = 0; // Middle Prox

int analogPin3 = 11; // Ring Prox

int analogPin4 = 15; // Pinky Prox

int analogPin5 = 20; // Ind Dist

int analogPin6 = 2; // Middle Dist

int analogPin7 = 10; // Ring Dist

int analogPin8 = 12; // Pinky Dist

int analogPin9 = 19; // Thumb Prox

int analogPin10 = 18; // Thumb Dist


int val = 0;

int val2 = 0; // variable to store the value read

int val3 = 0;

int val4 = 0;

int val5 = 0;

int val6 = 0;

int val7 = 0;

int val8 = 0;

int val9 = 0;

int val10 = 0;
```

```

void setup() {
    Serial1.begin(19200);

    pinMode(analogPin, INPUT_ANALOG); // set up pin for analog input
    pinMode(analogPin2, INPUT_ANALOG); //
    pinMode(analogPin3, INPUT_ANALOG);
    pinMode(analogPin4, INPUT_ANALOG);
    pinMode(analogPin5, INPUT_ANALOG);
    pinMode(analogPin6, INPUT_ANALOG);
    pinMode(analogPin7, INPUT_ANALOG);
    pinMode(analogPin8, INPUT_ANALOG);
    pinMode(analogPin9, INPUT_ANALOG);
    pinMode(analogPin10, INPUT_ANALOG);
}

```

```

void loop() {
    val = analogRead(analogPin); // read the input pin
    val2= analogRead(analogPin2);
    val3= analogRead(analogPin3);
    val4= analogRead(analogPin4);
    val5= analogRead(analogPin5);
    val6= analogRead(analogPin6);
    val7= analogRead(analogPin7);
    val8= analogRead(analogPin8);
}

```

```
val9= analogRead(analogPin9);  
val10= analogRead(analogPin10);  
SerialUSB.print("A");  
SerialUSB.print("a");  
SerialUSB.print(val);  
SerialUSB.print("b");  
SerialUSB.print(val2);  
SerialUSB.print("c");  
SerialUSB.print(val3);  
SerialUSB.print("d");  
SerialUSB.print(val4);  
SerialUSB.print("e");  
SerialUSB.print(val5);  
SerialUSB.print("f");  
SerialUSB.print(val6);  
SerialUSB.print("g");  
SerialUSB.print(val7);  
SerialUSB.print("h");  
SerialUSB.print(val8);  
SerialUSB.print("i");  
SerialUSB.print(val9);  
SerialUSB.print("j");  
SerialUSB.print(val10);
```

```

SerialUSB.print("KK");

delay(100);

}

```

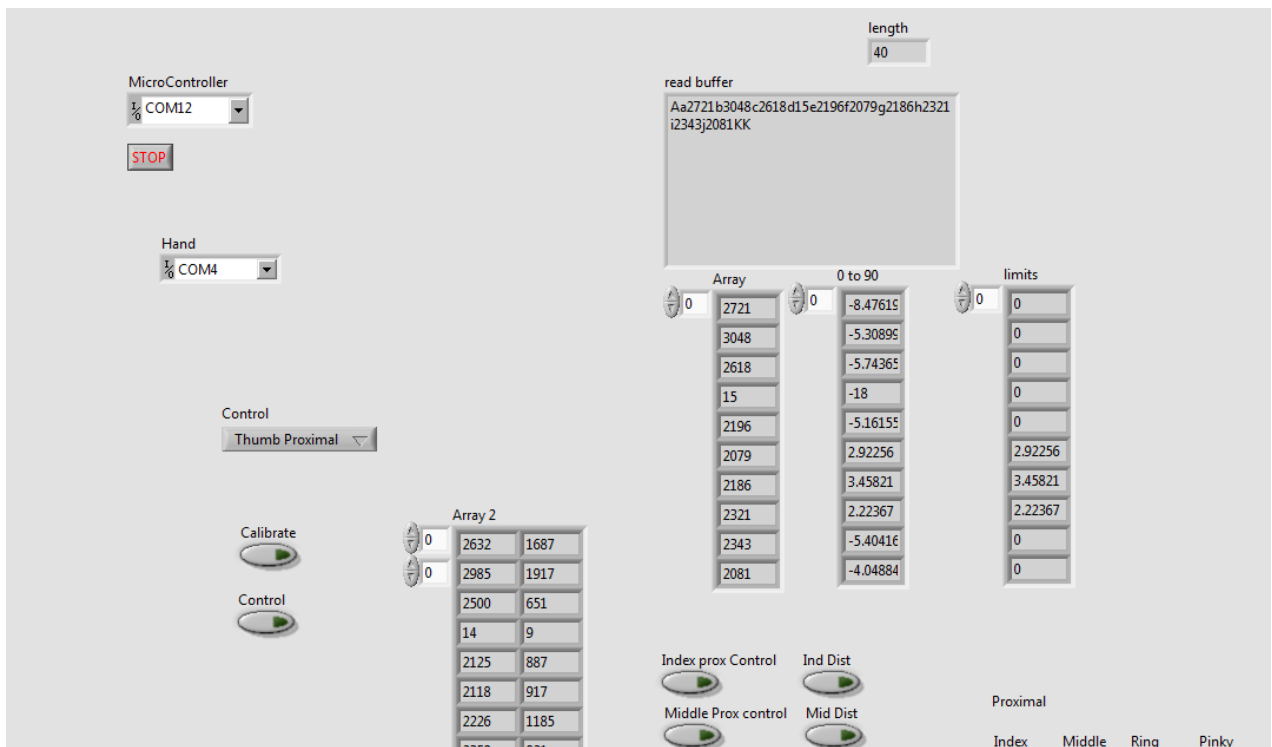


Figure B- 1: Front Panel of the data glove controller. Note the text box showing the input string from the Maple board. These values are collected in the table titled “Array”

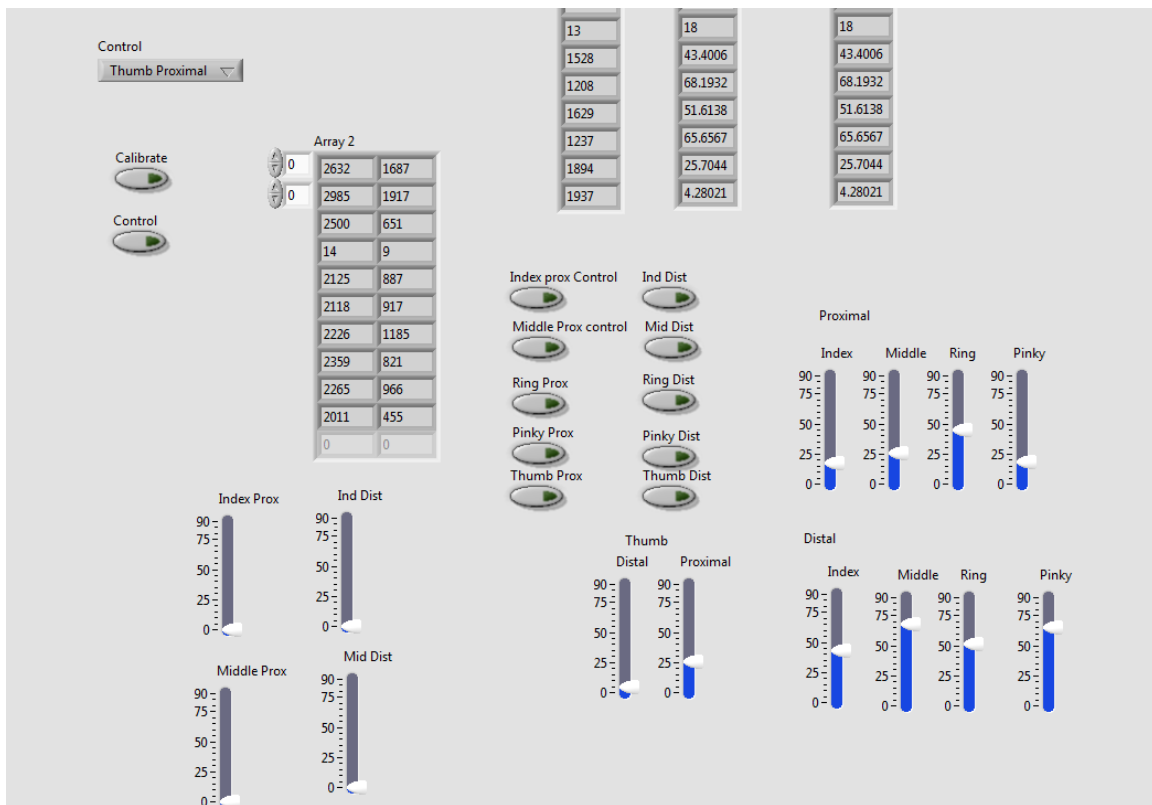


Figure B- 2: Labview panel with sliders visible. On the left are the direct control sliders, while on the right are the indicator sliders for the relayed value from the date gloves.

References

Atwell, C. (2012, Oct. 15). *Are gesture-sensing interfaces set to replace your mouse and keyboard? ms digit and leap motion say yes*. Retrieved from

<http://www.element14.com/community/community/news/blog/2012/10/15/are-gesture-sensing-interfaces-set-to-replace-your-mouse-and-keyboard-ms-digit-and-leap-motion-say-yes>

Baek, S., Jeong, Y., & Lee, B. (2002). Characteristics of Short-Period Blazed Fiber Bragg Gratings for Use as Macro-Bending Sensors. *Applied Optics* , 41 (4), 631-636.

Biswas, K. (2011). Gesture recognition using Microsoft Kinect. *Automation, Robotics and Applications(ICARA)* (pp. 100-103). Wellington: IEEE.

Carlson, W. (2003). *A Critical History of Computer Graphics and Animation*. Retrieved April 10, 2013, from Ohio State Department of Design:

<http://design.osu.edu/carlson/history/lesson17.html>

Cohen, D., & Halgren, E. (2004). Magnetoencephalography. In B. S. G. Adelman, *Encyclopedia of Neuroscience*. Amsterdam: Elsevier.

Cyber Glover Systems, LLC. (2011). *Cyber Glove III*. Retrieved Sept 2011, from Cyber Glove Systems web site: <http://www.cyberglovesystems.com/products/cyberglove-III/overview>

Cyberworld Inc. (2013). *P5 - Virtual Reality Data Glove*. Retrieved Sept 2011, from Cyber World Store web site:

http://www.cwonline.com/store/view_product.asp?Product=1179

Data Connect Enterprise. (2013). *Fiber Optics: The Basics of Fiber Optic Cable*.

Retrieved Feb 18th, 2013, from Data-Connect Web Site: http://www.data-connect.com/Fiber_Tutorial.htm

Englehart, K., & B., H. (2003). A Robust, Real-Time Control Scheme for Multifunction Myoelectric Control. *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING* , 50 (7), 848-854.

Fifth Dimension Technologies. (2005). *5DT Data Glove 5 Ultra*. Retrieved Sept 2011, from Fifth Dimension Technologies web site:

<http://www.5dt.com/products/pdataglove5u.html>

Fishel, J. (2012). Sensing tactile microvibrations with BioTac - Comparison with human sensitivity. *Biomedical Robotics and Biomechanics(BioRob)* (pp. 1122-1127). Rome: IEEE.

Flexpoint Sensor Systems, Inc. (2005). *About the Bend Sensor*. Retrieved Dec 2011, from Flexpoint flexible sensor systems:

<http://www.flexpoint.com/companyInfo/bendSensor.htm>

Future Predictions.com. (2011). *Mind Reading by ECoG as Brain Computer Interfaces Will Become Common*. Retrieved from Future Predictions.com:
<http://futurepredictions.com/wp-content/uploads/2011/05/ecog.jpg>

Gantz, B., Turner, C., Gfeller, K., & Lowder, M. (2005). Preservation of Hearing in Cochlear Implant Surgery: Advantages of Combined Electrical and Acoustical Speech Processing. *Laryngoscope* , 796-802.

Gentner, R., & Classen, J. (2009). Development and evaluation of a low-cost sensor glove for assessment of human finger movements in neurophysiological settings. *Journal of Neuroscience Methods* , 178 (1), 138-147.

Gernot, R., & al, e. (2008). Control of an electrical prosthesis with an ssvep based BCI . *IEEE Trans. Biomed. Eng* , 55 (1), 361-264.

Haas, L. (2003). Hans Berger(1873-1941), Richard Caton(1842-1926), and Electroencephalography. *J. Neurol., Neurosurg. Psychiatry* , 74 (1), 9.

Han, Y. (2010). A low-cost visual motion data glove as an input device to interprety human hand gestures. *Consumer Electronics* , 56 (2), 501-509.

Hochber, L., & al, e. (2006). Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature* , 442, 164-171.

Joiner, R., Messer, D., Light, P., & Littleton, K. (1998). It is best to point for young children: a comparison of children's pointing and dragging. *Computers in Human Behavior* , 14 (3), 513-529.

Kang, I., Schulz, M., Kim, J., Shanov, V., & Shi, D. (2006). A carbon nanotube strain sensor for structural health monitoring. *Smart Materials and Structures* , 737.

Kuruvilla, A., & Flink, R. (2003). Intraoperative electrocorticography surgery: useful or not? *Seizure* , 12, 577-584.

Leap Motion, Inc. (2013). *Leap Motion: Our Device*. Retrieved Feb 16th, 2013, from Leap Motion Web site: <https://www.leapmotion.com/product>

Lebedev, M., & Nicolelis, M. (2006). Brain-Machine Interfaces: Past, Present, and Future. *Trends Neurosci* , 29 (9), 536-546.

Lee, J.-H., Ryu, J., Jolesz, F., Cho, Z.-H., & Yoo, S.-S. (2009). Brain-machine interface via real-time fMRI: Preliminary study on thought-controlled robotic arm. *Neuroscience Letter* , 450 (1), 1-6.

Less EMF Inc. (1996). *Less EMF*. Retrieved Dec 5th, 2012, from EMF Shielding and Conductive Fabrics: <http://www.lessemf.com/fabric.html>

Logothetis, N., & al., e. (2001). Neurophysiological Investigation of the Basis of the fMRI Signal. *Nature* , 412 (6843), 150-157.

Microsoft. (2013). *Kinect*. Retrieved Feb 16th, 2013, from Xbox Website: <http://www.xbox.com/en-US/kinect>

Myers, D. (2010). *Wikipedia*. Retrieved 2013, from Electroencephalography: http://upload.wikimedia.org/wikipedia/commons/b/bf/EEG_cap.jpg

National Cancer Institue. (2013). *Brain*. Retrieved from National Cancer Instute: Seer Training Modules: http://training.seer.cancer.gov/images/brain/cerebrum_lobes.jpg

National Institute of Mental Health. (2009). *Magnetoencephalography*. Retrieved 2013, from Wikipedia: http://upload.wikimedia.org/wikipedia/commons/thumb/e/e6/NIMH_MEG.jpg/398px-NIMH_MEG.jpg

O'Keefe, J., & Dostrovsky, J. (1971). The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Res.* , 34, 171-175.

Quiroga, R. Q., & al, e. (2005). Invariant visual representation by single neurons in the human brain . *Nature* , 435, 1102-1107.

Smooth-On Inc. (2011). *Dragon Skin® High Performance Silicone Rubber*. Retrieved March 14th, 2013, from Smooth-On: https://www.smooth-on.com/index.php?cPath=1115_1129

Soderberg, & Cook. (1984). Electromyography in Biomechanics. *Physical Therapy* , 64, 1813-1820.

Sturman, D., & Zeltzer, D. (1994). A survey of glove-based input. *Computer Graphics and Applications IEEE* , 14 (1), 30-39.

Syntouch, LLC. (2013). *BioTac History*. Retrieved September 15th, 2013, from Syntouch: <http://www.syntouchllc.com/Products/BioTac/BioTac-History/>

Takktile, LLC. (2013). *Takktile Products: TakkStrip*. Retrieved from TakkTile: <http://www.takktile.com/product:takkstrip>

Tekscan. (2013). *Tekscan: Flexiforce Sensors*. Retrieved from Tekscan Website: <http://www.tekscan.com/flexible-force-sensors>

Tenzer, Y., Jentoft, L., & Howe, R. (2012). *Inexpensive and Easily Customized Tactile Array Sensors using MEMS Barometer Chips*. Cambridge: Harvard School of Engineering and Applied Sciences.

The Human Story: The Hands. (2011). Retrieved from Online Digital Education Connection: http://en.wikipedia.org/wiki/File:EEG_cap.jpg

Uutela, K., Hamalainen, M., & Somersalo, E. (1999). Visualization of Magnetoencephalographic Data Using Current Estimates. *NeuroImage* , 10 (2), 173-180.

Velliste, M., et. al. (2008). Neuronal Ensemble Control of Prosthetic Devices by a Human with Tetraplegia. *Nature* , 453, 1098-1101.

Velliste, M., Perel, S., Spalding, C., Whitford, A., & Schwartz, A. (2008). Cortical control of a prosthetic arm for self-feeding. *Nature* , 1098-1101.

Vrba, A., & Robinson, S. (2001). Signal Processing in Magnetoencephalography. *Methods* , 249-271.

Wolpaw, J., & al., E. (2002). Brain-Computer Interfaces for Communication and Control. *Clinical Neurophysiology* , 113, 767-791.