Creating A Gesture Controlled Robotic Hand with Haptic Feedback

A Dissertation
Presented to
The Academic Faculty

By

Ahmed Elmi

In Partial Fulfilment
Of the Requirements for the Degree
BSc Computer Science

Swansea University
Prifysgol Abertawe

Swansea University
Department of Computer Science
Swansea, Wales

2018

DECLARATION

This work has not previously been accepted in substance for any degree and is not being currently submitted for any degree.

Date:

Signed:

STATEMENT 1

This dissertation is being submitted in partial fulfilment of the requirements for the degree of BSc in Computer Science.

Date:

Signed:

STATEMENT 2

This dissertation is the result of my own independent work/investigation, except where otherwise stated. Other sources are specifically acknowledged by clear cross referencing to author, work, and pages using the bibliography/references. I understand that failure to do this will amount to plagiarism and will be considered grounds for failure of this dissertation and degree examination as a whole.

Date:

Signed:

STATEMENT 3

I hereby give consent for my dissertation to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Date:

Signed:

ABSTRACT

This report discusses the design and functionality of a robotic hand that can be controlled through a sensor glove which is worn on a user's hand. In addition to controlling the robotic hand, the user will have the ability to almost feel what the robotic hand is touching due to sensor receptors on the fingertips of the robot. After some micro processing the user feels force in their fingers due to tension from a string. This gives the user the perception of feeling what the robot feels thus providing haptic feedback. Several concepts of a hand structure were evaluated and the most dexterous and convenient one was chosen as the subject of this report.

ACKNOWLEDGEMENTS

Firstly, I would like to thank my supervisor Dr Mabrouka Abuhmida for her unparalleled support throughout the year. She has given me advice, motivation, and direction which massively helped me in completing this project.

I would also like to thank my second marker Dr Xiuyi Fan for his feedback to my initial document and teachings of software testing.

Thank you to my parents for their continuous support and making sure I stay on track with my studies and ensuring I live a healthy lifestyle.

Thank you to my friends who kept me mentally healthy and stress free. Thank you for reminding me that we are all in this together and that sometimes a well needed break is necessary.

Finally, I would like to thank anybody who has participated in this project whether it was just testing/reviewing or simply giving guidance at any stage.

Project Outline

Introduction

In this chapter we will give an overview of the key concepts behind the project that we will expand upon later. Also will be included the project specification and my motivation behind doing this project.

Related Work

This section will cover past and current work that is related to this project. We will examine the work to gain a better understanding of how we can further the field.

Background

Here we shall explore the concepts behind the design of the project and the components that I will be using to make the robotic hand.

Project Management

In this chapter I discuss how I have managed my project and what software lifecycle model I followed.

Implementation

This section completely describes how I have gone about making the robotic hand and the results that I have gotten.

Evaluation

I discuss the problems and successes that I had and what limitations affected the end result.

Conclusion

I conclude the project by giving a summary of the project, my opinions of it, and what future developments could arise.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

The human hand is one of the most universal tools of nature. It is one of our main lines of interaction with the environment. So to have it replicated in a robotic sense would mean that our interactivity with the environment would be increased. In modern society, robotic hands are used in various fields such as industry and healthcare. In industry they allow us to grasp objects that can’t be carried in conventional ways. In the healthcare a robotic hand could serve as a replacement for suffering amputees. However based on aesthetics amputees tend to go for a prosthetic hand instead because it looks more human. In contrast to a prosthetic hand, a robotic hand would be focused more on functionality rather than aesthetics. The input method for the control of a robotic hand varies widely across the different problems that they are built for. For example some could be controlled by buttons and levers whereas others could be controlled by the electrical impulses that come from nerves in the brain. The robotic hand that I shall be creating will be controlled through a glove where the gestures are interpreted through a variety of sensors and presented through the robotic hand. To further the project I shall not only bring mobility to the robotic hand but also make it so that haptic feedback can also be returned from the hand to the sensor glove. (Gingichashvili, 2017).

1.1 Scenario

A man who works at a factory was involved in an accident whilst operating one of the industrial machines. He was sent to hospital and the doctors had told him that he would have to have his arm amputated due to a bad wound infection he had received. Based on the nature of his injury he was fortunately eligible to receive a robotic prosthetic hand which not only would restore hand mobility but also return haptic feedback based on force at the fingertips. The robotic hand receives input wirelessly from a glove that is worn on the other hand.

1.2 Project Aims/Specification

Keeping clear and concise specifications will ensure that I follow the intended direction of the project as linearly as possible. This will also give me milestones which I could use as checkpoints and measure the progression of my project. Below are the specifications that I believe the end product should have:

- Robotic hand must copy hand gestures of the user through sensor gloves.
- Robotic hand must be able to hold light objects such as a bottle.
- Sensor glove must be able to receive input from the force sensors on the robotic hand to be able to provide haptic feedback.

CHAPTER 2. RELATED WORK

In this section I will describe work that is similar to my project and provided me with useful information.

2.1 Marcus Sergius

Marcus Sergius was a famed roman general who led his legion in the Second Punic War in 218 BC. He is well known in the prosthesis circle because he is the first documented user of a prosthetic hand. The prosthetic hand was made out of iron and structured so that he could hold his shield in battle (Pliny the Elder, 1637) (Chadwick, 2003).

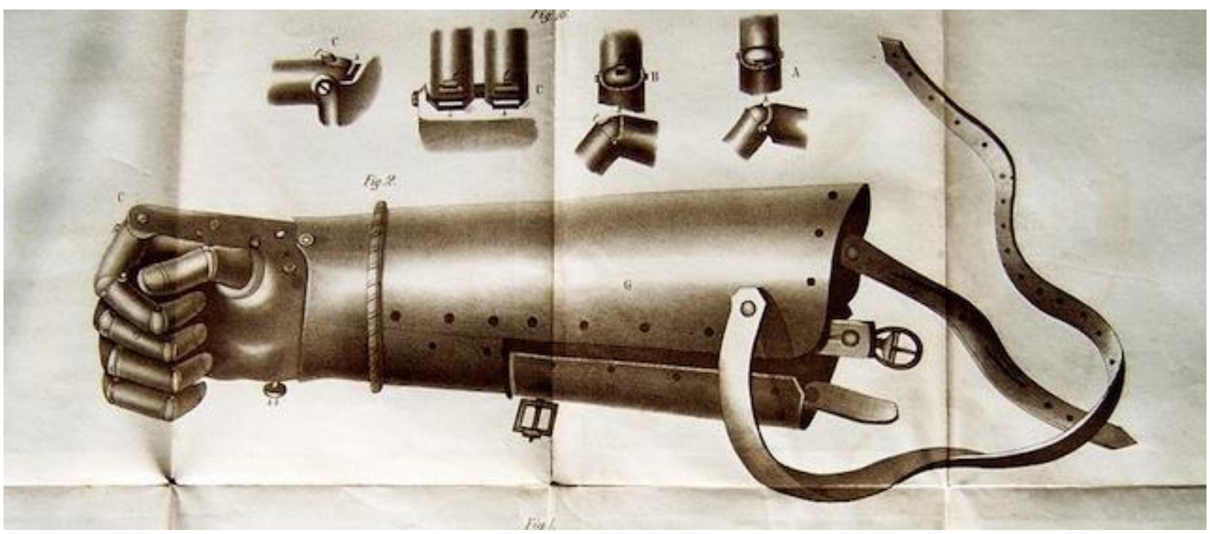


Figure 1 - Iron and leather prosthetic arm (Gizmodo, 2017)

2.2 3D Printing

3D printing is the process of creating a three dimensional object by using material that is solidified under computer control. It is mainly used for rapid prototyping. A 3D printer would typically need some sort of the 3D model file to print an object. It can also accept sources such as and Additive Manufacturing File. There are variations in the way that a 3D model can be printed.

Stereolithography (STL) is one such method that utilizes light to combine molecules together and form long chains of polymers. This reaction is called photopolymerisation. Each cross section combines as the model is printed layer by layer. (Hull, 1984)

Fused deposition modelling (FDM) is another variation where heated plastic is fed through a tube and a moving platform ensures it is deposited in the desired shape. Similarly, FDM also prints the model in a layer by layer order. (Wikipedia, 2018)

2.2.1 ABS vs PLA

Acrylonitrile Butadiene Styrene (ABS) and Poly Lactic Acid (PLA) are the two main materials used in 3D printing. It is important to use the right material for what you want to print. When deciding what material to use, one should review the desired properties of their model. ABS and PLA don't have any differences that are immediately apparent however I will investigate the characteristics of using both materials and conclude upon which one is most suitable for this project.

ABS is an oil based thermoplastic polymer which has a strong resistance to corrosive chemicals and physical stress. It is relatively inexpensive and very sturdy material. ABS has a high melting temperature and therefore must be printed on a heated plate to avoid cooling which causes undesired morphing. When printing with ABS the smell can be unpleasant like burning plastic therefore requires ventilation. (Digital Trends, 2018)

PLA is an organic thermoplastic polymer as it is derived from natural renewable sources such as starch and sugarcane. This makes it biodegradable and a sustainable material for prototyping. In contrast traditional plastic takes centuries to break down. PLA does not produce toxic fumes when heated unlike ABS. PLA is also strong however cannot withstand temperatures that are too high. (Biomass Packaging, 2018)

In conclusion both ABS and PLA have advantages and disadvantages. The robotic hand that I am creating will not require the property to withstand absurdly high temperatures. I believe PLA is the rightful choice of material as it will fulfil its functionality and also be cleaner for the earth.

2.3 The Shadow Hand

As time goes by the robotic replication of the human hand gets better and better. The shadow hand is a pretty good representation of how much engineering has advanced. The shadow hand was developed by The Shadow Robot Company in London and is one of the most dexterous examples of a robotic hand.

The standard description of a robotic arm would include their degrees of freedom. This is a number that highlights how many single-axis rotational points there are in the arm or hand (Paul, 1981). A normal human hand has 27 degrees of freedom including the wrist and 21 not including the wrist. Each finger has 4 degrees of freedom due to the two joints in between the distal, middle and proximal phalanges. Both of these joints allow for flexion and extension which accounts for 2 DOF. The other 2 DOF arise from the joint between the proximal phalange and the palm which allows for not only flexion and extension but also abduction and adduction.

Compared to the finger, the thumb has 5 DOF. There is only a single joint between the phalanges which provides only 1 DOF (flexion/extension). 2 DOF come from the joint between the proximal phalange and palm (flexion/extension, abduction/adduction). The final DOF comes from a joint inside of the palm between the metacarpal and trapezium. (Canadianer, 2015)

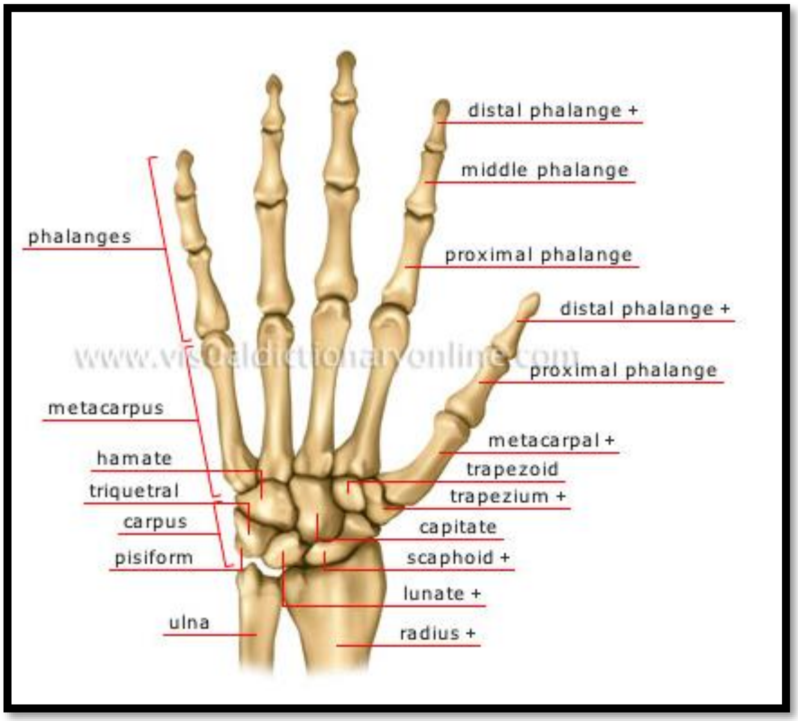


Figure 2 - Skeletal structure of a hand.

(Visual Dictionary, 2014)

The design of the Shadow Dexterous Hand is quite similar to that of the natural human hand. In fact it almost replicates it in the sense of degrees of freedom. The Shadow Dexterous Hand has 24 joints and 20 degrees of freedom in total which makes it pretty close to a natural hand which has 27 joints and 27 degrees of freedom.

Figure 3 - The Shadow C6M Smart Motor Hand in front of the Shadow C3 Dexterous Air Muscle Hand. (Shadow Robot Company, 2017)

2.4 Nintendo Power Glove

The power glove is a gaming accessory designed for the Nintendo Entertainment System to allow motion control of games. It was created to provide a more complete and engaged gaming experience compared to the traditional controller system. The power glove is based on the patented wired glove created by VPL research however many modifications had been made to end up with the power glove (Harvill, Zimmerman, & Grimaud, 1987). Various sensor technologies allowed the motion of the hand to be captured and processed. One of these sensors is a type of variable resistor called a flex sensor which measured the bend in each finger by its corresponding resistance. The power glove has 2 ultrasonic speakers that act as transmitters and the TV has 3 ultrasonic microphones placed around it which act as receivers. The transmitters each fire short bursts of 40 kHz sound waves and the system calculates the time it takes for it to reach the receivers. Based on this information the x, y and z positions of the power glove can be calculated. This means that the power glove can detect yaw and roll however it cannot detect pitch due to the hand being able to pitch without moving the location of the transmitters (Power Glove, 2018).

In comparison to my project, I shall also include the use of flex sensors to measure the motion control of fingers and use this data as input to my other subsystems. Since the days of the Nintendo Power Glove, robotic hand technologies haven't made the advancements that they deserve and I hope to gather all past and current knowledge to aid me in completing this work.

Figure 4 - Nintendo Power Glove. (Computing History, 2016)

CHAPTER 3. BACKGROUND

3.1 Hardware

3.1.1 Robotic Hand

For my robotic hand, I have chosen to make it myself from scratch. To do this I will need a 3D printer to print the parts. I have chosen an open source model developed by e-NABLE called the Raptor Hand. I picked this model out of the many other open source hand models because the Raptor Hand is designed for quick printing and easy assembly. Compared to the other hand models, this model doesn't require any other equipment such as screws, drills and other construction equipment. I decided that that hand model should be cost efficient but still not compromise the end product quality. These requirements fitted well with the Raptor Hand specifications. Printing the hand is important as it will be the foundation of my work to build on.

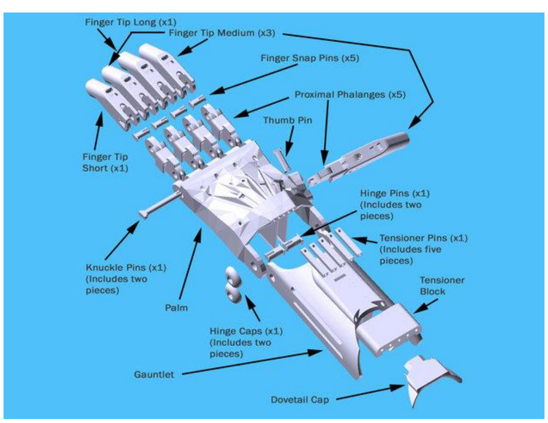


Figure 5 – The Raptor Robotic Hand. (e-NABLE, 2016)

3.1.2 Resistor

One of the first things that we learn about electricity is that all materials fall into categories if electrical conductor or electrical insulators. Conductors allow electricity to pass through them whereas insulators do not. After learning more about electricity, you learn that it is not that simple. Anything can conduct electricity if enough voltage is applied. This is where the concept of electrical resistance is derived. Electrical resistance; how easily a material will conduct electricity. At an atomic level this is dependent on the structure of the material. Metal easily conduct electricity due to its atomic lattice arrangements that allows easy access to for some electrons to move and carry electricity. In contrast, plastic has a polymer chain formation which has many bonds that occupy the electrons so that it's harder to carry the flow of electricity.

Along a circuit, the electromotive force that provides a current is called a voltage. The mathematical equation to calculate voltage is called Ohm's Law:

$$V = I \times R$$

Resistance is the amount of electromotive force required to produce 1 amp of current flow through a circuit. By rearranging the above equation we can calculate resistance with:

$$R = V/I$$

A resistor is an electrical component that is used to restrict current by bottlenecking the flow of electrons in a circuit and creating a potential difference. Resistors are made with precision to produce a specific amount of resistance. It works by forcing the flow of electricity through a thin copper wire that spirals around an insulating layer of ceramic. This makes it difficult for electrons to get around the circuit and thus diminishes its electromotive force (Woodford, 2016).

3.1.2.1 Flex Sensor

A flex sensor is a type of resistor, specifically a variable resistor that can change its electrical resistance depending on how much it is being bent. The principle that it uses to vary its resistance is based on the engineering concept of deflection. It has a layer of conductive ink on the outside which when stretched will reduce its conductive cross section and thus increasing the resistance by making it harder for electrons to flow through. It is similar to a potentiometer in the sense that its resistance can be controlled. However, sometimes the resistance may have to depend on outside stimuli such as force which the flex sensor accounts for.

Additionally, the resistance of a material is affected by its internal temperature. For every material, the higher the temperature the more energy its constituent atoms have. This causes the atoms to vibrate which makes it harder for electrons to flow through the material therefore increasing its electrical resistance (Woodford, 2016).

A typical flex sensor of standard issue at room temperature in its resting flat state will have a base resistance of approximately 30K Ω and a bend resistance range of 50K Ω to 70K Ω when bent to 90 degrees (Jimbo, Flex Sensor Hookup Guide). However, other factors can affect resistance such as temperature, width, length and depth of the flex sensor.

3.1.2.2 Pressure Sensor

A pressure sensor will be one of my necessary components in the circuit. It will be used to provide haptic feedback to the sensor glove by creating tension in strings attached to the fingers of the glove. The pressure sensor is a type of resistor technically called a force sensitive resistor. At resting state without any force being applied to the FSR, the resting resistance is greater than 1M Ω. The greater the force being applied to it, the lower the resistance. When force is applied to the FSR the resistive cross section decreases meaning delocalized electrons have an easier path to flow through. This causes a reduction in electrical resistivity. The FSR has a minimum and maximum weight threshold. If the maximum threshold is 10kg the system will not be able to differentiate between a 20kg weight and a 30kg weight.

Figure 6 displays the relationship between resistance and force for a FSR that has a maximum threshold of 1kg. For the first 20g the relationship has a steeper gradient which is decreased after 20g. This is due to turn on threshold meaning that a certain amount of force must be present for the FSR to decrease its resistance. Between 50g and maximum threshold the relationship is linear. The pattern of this graph would be the same for different FSRs that have different maximum thresholds (Jimbo, Force Sensetive Resistor Hookup Guide, 2003).

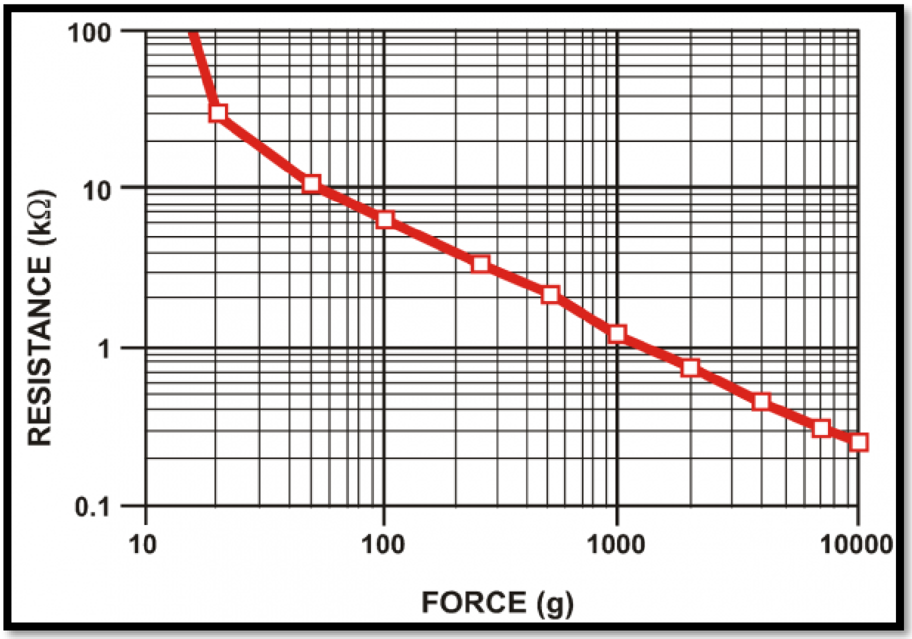


Figure 6 – Graph displaying relationship between Resitance and Force. (SparkFun Electronics)

3.1.3 Servomechanisms

A servo is an electrical component similar to a motor but uses a control loop feedback mechanism called a proportional integral derivative controller or PID for short. This feedback mechanism continuously measures the position of the servo and appropriately corrects it to a desired set point (Bob, 1995). The position is controlled by a potentiometer which is a variable resistor. Similarly, an encoder also measures position and speed however the potentiometer values can drift due to temperature affecting its electrical resistance (Baldor Electric Company).

Servos are essential in this project as they will be acting as the muscles for the robotic hand by providing movement. Each servo will rotate to a specific degree which in turn will pull upon a string that is connected to the respective fingertip in the robot. This then results in a contraction that pulls the finger allowing it to clench the hand. Furthermore, I will use servos to also provide haptic feedback in the sensor glove by allowing it to produce tension in strings running along the back of each finger hence creating a sensation of touch. In the robotic hand the degree to which the motors should pull the strings will be dependent on its mapping to the flex sensors. In the sensor glove the degree to which the servos should create tension will depend on its mapping to the pressure sensors. Ideally, I wish to use a servo that has a high enough torque value suitable for producing the necessary tension in the strings to cause the robotic finger to bend. I would also want the servos to have metal gears instead of plastic gears as this will greatly enhance the strength of the servos (ServoCity, 2018).

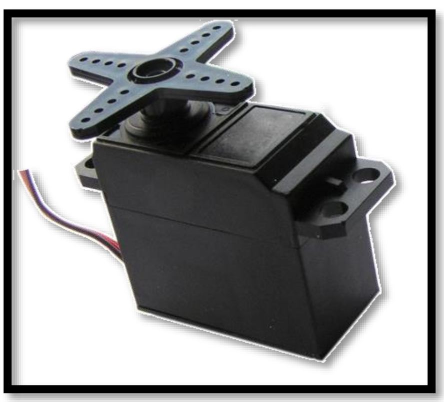


Figure 7– A small servo motor. (Chioka, 2016)

3.1.4 *Microcontroller*

A microcontroller is a small computer that is usually embedded onto a larger component so that it can control the inputs and outputs of the product. It serves as the logic for a system and will regulate its mechanical dynamics. For this project I have decided to use an Elegoo Uno R3 as it provides all the necessary properties that I require. The Elegoo

Uno R3 allows for digital input/output through the digital pins and analogue input/output through the analogue pins. Additionally, I can code the logic into the board via a USB port by using the Arduino integrated development environment. Within a microcontroller there exist three different types of memory. These are categorized as flash memory, static random access memory and electrically erasable programmable read only memory (Arduino, 2018). An Elegoo Uno R3 has an ATmega328 chip which has the following amounts of memory:

Flash Memory: 32k bytes of which 5k is used for the boot loader

SRAM: 2K bytes

EEPROM: 1k bytes

The code from the IDE is stored in flash memory on the Elegoo board. When the code runs it may create and manipulate some variables which are stored in the SRAM. EEPROM is memory that can be used by programmers to store long term data (Mittal & Vetter, 2015). Flash memory and EEPROM are both non-volatile memory which means that after the power is cut from the circuit it will still retain its data. In contrast, SRAM is volatile memory which will be lost and reiterated upon connection to a power supply. (Y, Y, & K, 1972)

Figure 8 – An Elegoo Uno (Average Maker, 2014)

3.2 Circuit

3.2.1 Data Flow and Subsystems

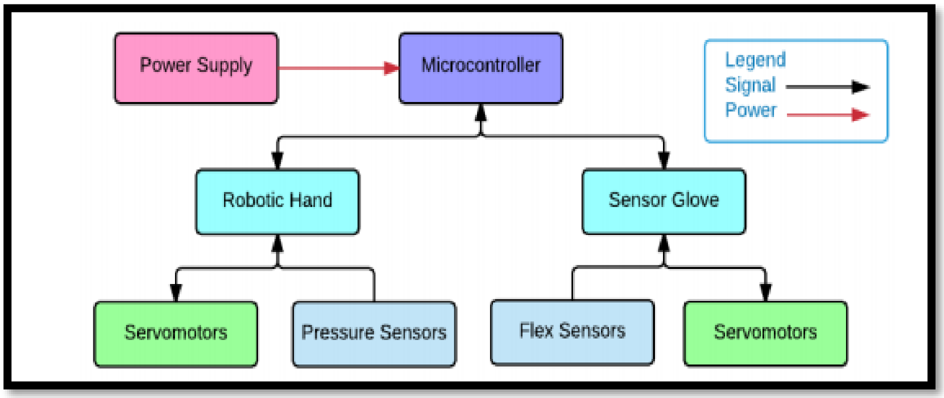


Figure 9 – Block diagram of the sensor controlled robotic hand (Hon, 2016)

Figure 9 displays the flow of data that shall occur in my system. For the power supply I shall provide the necessary voltage to the microprocessor through a USB connection to my laptop. The microprocessor will be my central unit of logic and will be responsible for controlling inputs and outputs. The system is mainly separated into two parts; the robotic hand unit and the sensor glove unit. The sensor glove will consist of flex sensors running along the back of each finger. This will collect analog data and feed it to the microcontroller which then maps this data appropriately to the servos in the robotic hand. This will result in the servo pulling strings to make the respective robot fingers bend. This subsystem covers the robotic hand imitating the gesture of the sensor glove.

Pressure sensors on the robotic hand will also collect analogue data and feed it to the microcontroller which maps this data appropriately to servos on the sensor glove. These servos will create tension in respective strings running along the back of each finger to create a sensation of touch. This subsystem covers the robotic hand producing haptic feedback to a user with the sensor glove.

3.2.2 *Electrical Circuit*

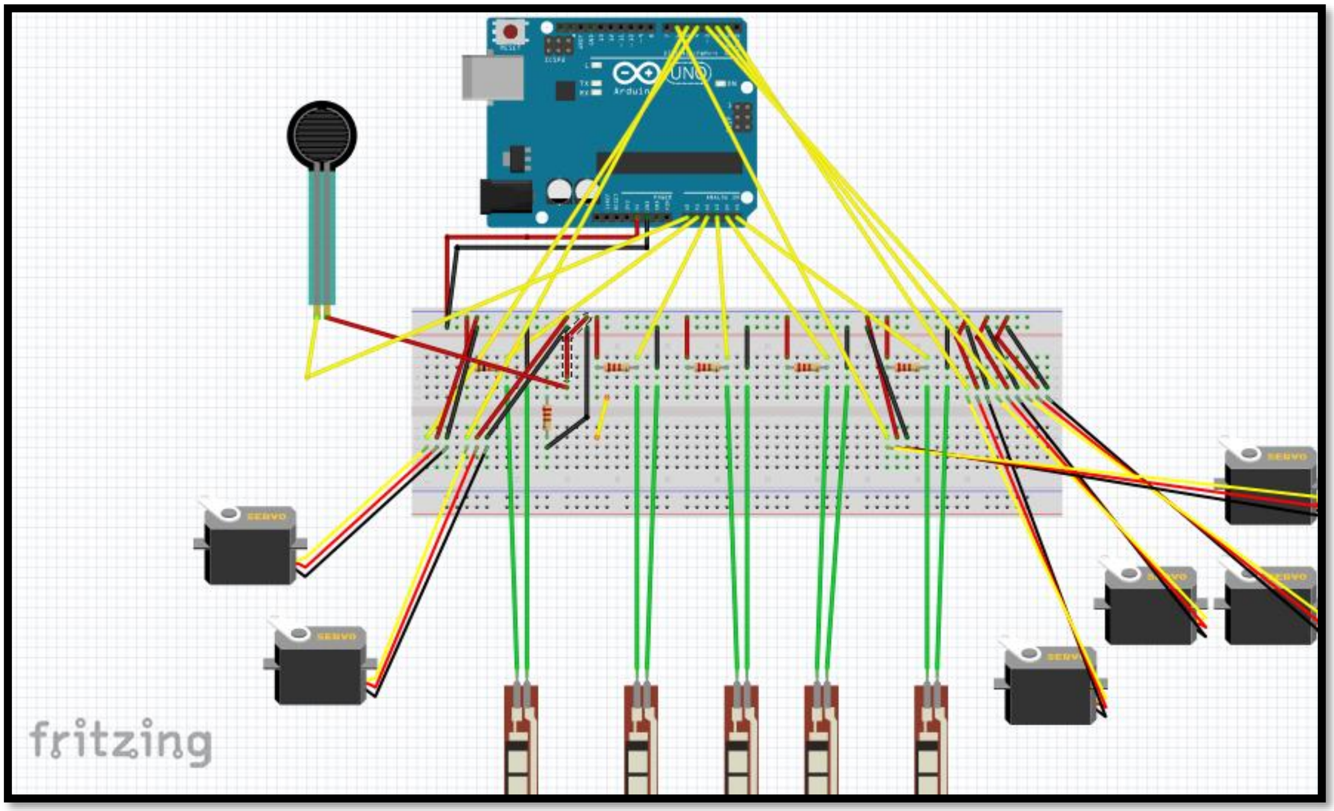


Figure 10 – Breadboard display of circuit routing in Fritzing.

I used an open source hardware displaying software called Fritzing to visualize and plan how I would connect all my system components together. Figure 8 displays the circuit that I ended up with. It shows the connections of my 6 servos, 5 flex sensors and 1 force sensitive resistor. The breadboard is one of the most fundamental pieces of equipment as it helps me create the circuit by providing a foundation upon which I can have the freedom to make connections that I wouldn't otherwise be able to make or it would be very messy and difficult.

Power will be supplied by the arduino 5V output port and the circuit will end at the anode. The force sensitive resistor and all the flex sensors are connected to 3 points through the circuit. First is the cathode which is the negatively charged electrode upon which electrons will enter the flex sensor. The anode, also referred to as the ground, is where the electrons will leave the circuit component. The final connection is the arduino pin which it would provide analogue data to.

The servo motors will also have the same connections as the flex sensor; the only difference being that instead of connecting to an analogue pin on the arduino, it will connect to a digital pin.

Initially, I found it difficult to think about how I was going to wire everything but after learning more about electrical circuits and how a breadboard worked, it became a lot easier and more systematic.

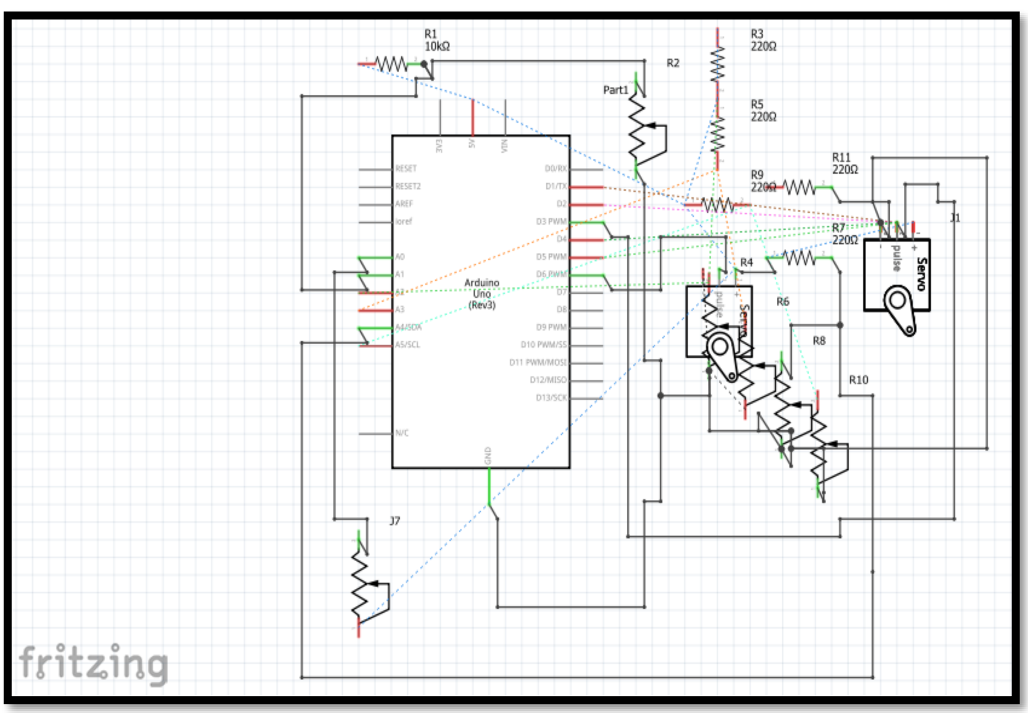


Figure 11 – Schematic view of circuit in Fritzing.

CHAPTER 4. PROJECT MANAGEMENT

In this chapter I will talk about my chosen software development methodology (SDM), why I chose it. Furthermore, I will reflect upon my timeline by comparing and contrasting with my original aims and milestones.

4.1 Software Lifecycle Method

The Software development method that I have decided to go with for my project is the prototyping model. I have chosen this because my project isn't heavily code and I believe I can produce quick working prototypes to help me test some of the most difficult parts and identify problems early on in the life cycle. Additionally, prototyping allows maximum user interaction between me and the target user base. Once a prototype is created, I can receive feedback for which I can use as improvement for the next prototype. I plan on repeating this cycle 3 times to strengthen the direction of my project. Another advantage is that users are a lot better at testing and reacting to a prototype than they are giving specifications. In contrast, the waterfall method would have users give specifications and later realise that it isn't what they had wanted.

4.2 Schedule

Task Name	Start Date	End Date
Initial Document	20/10/2017	01/11/2017
Gregynog Presentation	30/01/2018	03/02/2018
Student Project Demonstration Fair	15/05/2018	15/05/2018
Final Dissertation	15/03/2018	08/05/2018
Presentation Plan	15/12/2017	17/12/2017
Presentation rehersal	20/12/2017	20/12/2017

As part of my project management I created a Gantt chart to keep on track of my implementation and dissertation. It mainly was used to keep the project aims on focus and ensure delivery of my end product.

Figure 12 – Gantt chart showing schedule

The initial document is a subsidiary to this final dissertation and prepared me by giving me a preliminary investigation into my project. The Gregynog presentation gave me the opportunity to showcase my plan and at the time current development. It allowed me to defend my project and gain important feedback to use for future developments.

CHAPTER 5. IMPLEMENTATION

5.1 Preparation

I knew that I was going to use an Elegoo Uno R3 board which is a similar variation of an Arduino Uno therefore I would need to use the Arduino Integrated Development Environment to code my sketches for the functionality of my system. The Arduino compiler/IDE accepts C and C++ and the Arduino language is made up of functions from these languages. Therefore, I tasked myself in learning the arduino language and becoming comfortable enough to be able to code the logic for my system. The Arduino language isn't too complicated and due to my previous programming experience, it didn't take me long to get a grasp on it. The Arduino programming notebook (Evans, 2007) gave me a complete overview of how to code in the language and was very useful in teaching me the specifics of the code especially for controlling the servos.

5.2 Hardware

5.2.1 *3D Printing the Hand*

To make the robotic hand I needed to use a 3D printer to print the parts and pieces that I could use to assemble into a hand. The robotic hand model that I used was designed to work at scales between 100% and 170%. I decided that I would print the hand at 100% so that it can more closely resemble the size of a standard human hand. Printing it at 100% scale meant that the inside palm area would approximately measure out to be 55mm. I didn't use any supports for printing any of the hand component however two of the parts had supports built into the design. These two parts were the palm and gauntlet pieces. I decided to use poly lactic acid (PLA) as my 3D printing material for a more organic and smooth texture of the robotic hand. The settings that I used on the 3D printer were 0.2mm layer height, 35% infill and 2 outlines. The robotic hand took 13 hours to print and approximately 1 hour to assemble.

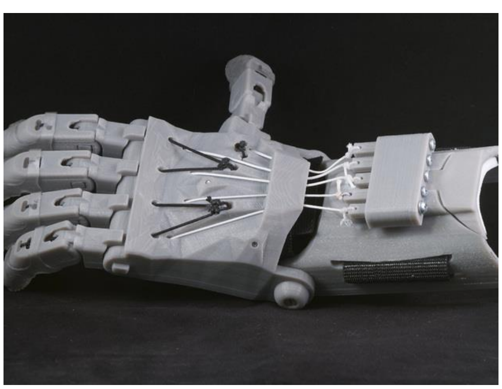


Figure 13 – 3D printed Raptor Hand. (e-NABLE, 2016)

5.2.2 *Making the Flex Sensors*

The flex sensors on the market were too expensive so I decided to understand the concept of how they work and try to recreate it using household materials to create an efficient and affordable alternative. The resistance of the flex sensor increases the more it is being bent due to a reduction in its conductive cross section. Following this concept I knew I would need a conductive material that is flexible and can handle the strain of being bent.

After doing some research on how others have overcome the same predicament, I found a video of a person making a flex sensor out of nothing but cardboard, paper, aluminium foil, glue and a pencil (Tech, 2017). I decided to follow this procedure to create my flex sensors. Firstly, I cut out a strip of paper and fully covered it in graphite by using a pencil. The graphite is used as a substitute for the conductive ink because graphite is a good conductor of electricity due to its many delocalized electrons. Being on the paper also means that it is flexible and will not be destroyed if deflected to maximum finger bend. The next step was to cut two strips of aluminium foil slightly thinner and smaller than the graphite strip. The two aluminium strips will be on either side of the graphite strip however they must not touch each other because the flow of electricity must be forced to go through the graphite and not be given a shortcut through the foil. Lastly, I cut two strips of cardboard to act as the outer layer of the flex sensor. An aluminium foil strip is stuck to each cardboard strip therefore the cardboard strip must be larger than the graphite strip. Afterwards, I used electrical tape to stick a frayed wire with a male end to each aluminium strip. This provides the connection points for it to integrate with the rest of the system. The graphite strip is placed in between both aluminium foil strips and the cardboard pieces are stuck together using electrical tape. This gives the flex sensor a more durable outer layer. The electrical tape is an insulator and therefore will not conduct electricity or interfere with the mechanics of the flex sensor.

5.2.3 Connecting the Parts

I followed the schematics that I provided in figure 10 and 11 to connect the components. The breadboard was small so I had to make everything a lot more compact. I found myself having to use female to female wires to extend some of the connections that I had due to the normal male to female wires being too short. I used resistors for each of the flex sensors and the force sensitive resistor even though the flex sensors and FSR are variable resistors itself. This is because using a secondary standard resistor will create a voltage divider which allows the output voltage to a fraction of the input voltage. This allows for better reading of the analogue data.

5.3 Software

5.3.1 Reading in Analogue Values from the Flex Sensors

After having the flex sensors connected, I wanted to test it out and see the range of values that I would get depending on how deflected it got. Firstly, I had to define the attributes for which analogue pins the flex sensors would connect to. I stated this

outside of the loop and setup. Secondly, within the loop I declared the variables to store the flex sensor value into by using an analogue read function. Now that the flex sensor value is stored in an integer variable I can display it to the serial by using a print function.

```
int thumbflexposition;

thumbflexposition = analogRead(thumb);
Serial.println("Sensor position");
Serial.println(thumbflexposition);
```

Figure 14 – Code for reding flex sensor data.

5.3.2 Controlling Servo with Flex Sensor

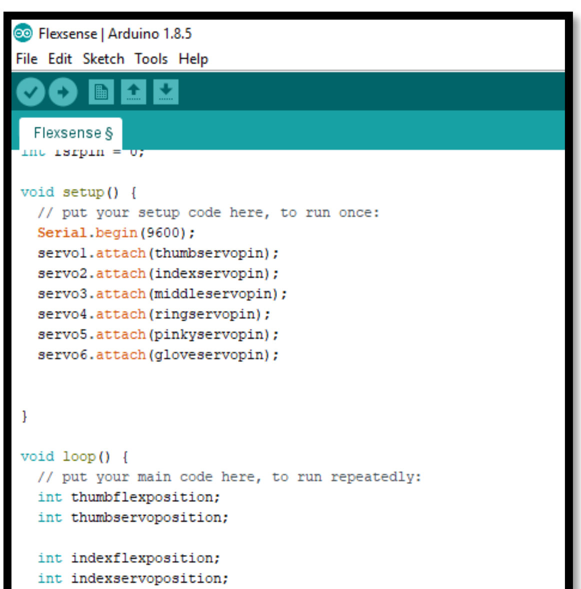
```
#include <Servo.h>
Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;
Servo servo5;
Servo servo6;

int thumbservopin = 9;
int thumb = 5;

int indexservopin = 8;
int index = 4;

int middleservopin = 7;
int middle = 3;
```

To control the servo I would need to use the Servo library which has 6 functions; attach(), write(), writeMicroseconds(), read(), attached() and detach(). Initially, I declared a servo object for each finger. I would also need to specify a variable for which pin each servo will connect to on the Elegoo R3. In the setup section of my sketch I needed to initialise the servos and attach each servo to its respective pin. In the loop section I declared a variable that I would use to store the servo positions. Next I mapped the value ranges from the flex sensor to the minimum and maximum of the servo. This would be 0 and 90 degrees. I constrained the servo position value to ensure it would go under 0 degrees or

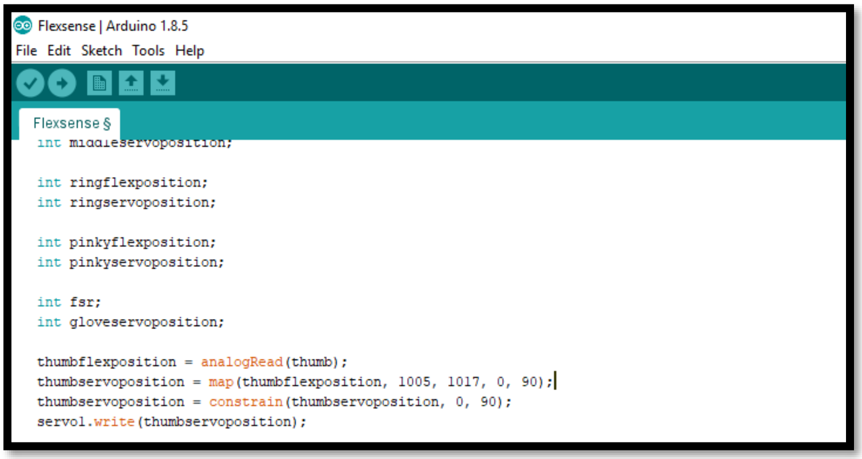


```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  servo1.attach(thumbservopin);
  servo2.attach(indexservopin);
  servo3.attach(middleservopin);
  servo4.attach(ringservopin);
  servo5.attach(pinkyservopin);
  servo6.attach(gloveservopin);


}

void loop() {
  // put your main code here, to run repeatedly:
  int thumbflexposition;
  int thumbservoposition;

  int indexflexposition;
  int indexservoposition;
```



exceed 90 degrees. Lastly, I assigned the servo position value to its servo by using the write() function.

Figure 15– code for controlling servo motors.

5.3.3 Implementing Haptic Feedback

Adding the haptic feedback feature was quite simple as all I had to do was read in the value from the force sensitive resistor then map this value to the servo with a range of 0 and 90. Finally, I would write this value to the servo that is attached to the glove and therefore pulling the string attached to the middle finger back to recreate the sensation of touch.

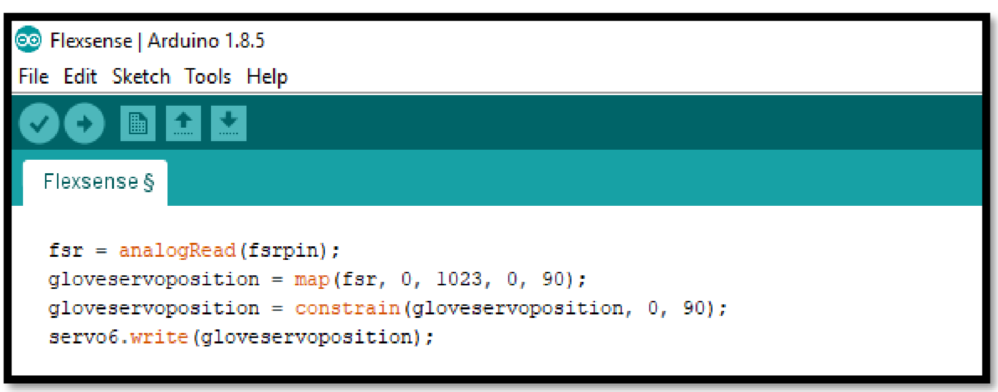


Figure 16 – Code for implementing haptic feedback.

5.4 Testing and Iterations

For my project development methodology I decided to follow a rapid prototyping method so that I could get early feedback and testing. The first prototype had the flex sensors working and the servos would react to the deflection in the flex sensor. I had asked some people to test it out and see what they thought. The feedback I received was that the flex sensor may have had too much noise and would not react in time or randomly cause the sensors to react. I subsequently did some research on why this was happening and tried many different ways to try and fix it. After much trial and error, I resolved the issue by remaking the flex sensors however this time using flat cardboard. I believe the issue came from the fact that the cardboard that I was initially using had inner folds which may have affected the inner aluminium foil lining of the flex sensor.

For my second iteration, I had rectified the earlier problem from the first iteration. After testing for a second time and receiving another round of feedback, I was told that the hand was a lot better but the problem this time was the haptic feedback. I was told that the servo on the sensor glove hand pulled too hard maybe I should lighten the tension in the string. Fortunately, this was an easy fix as the only thing that I had to do was alter the code a little bit so that the mapping of the flex sensor to the servo would be so narrowed.

Finally, for the third iteration I had multiple people test out my robotic hand. This time there were little to no problems with the hand and the users said they were happy to accept it as a completed product.

5.5 Results

I am glad with the product that I ended up with after several weeks of implementation. I believe the implementation was a success because looking back at my project specifications/requirements I can say that I have met all my goals.

- Robotic hand must copy hand gestures of the user through sensor gloves.
- Robotic hand must be able to hold light objects such as a bottle.
- Sensor glove must be able to receive input from the force sensors on the robotic hand to be able to provide haptic feedback.

The robotic hand reciprocates the gestures done using the sensor glove due to flex sensors and servo motors. The hand can hold light objects and they won fall out. Haptic feedback was implemented using force sensitive resistors and a servo motor on the back of the sensor glove.

5.6 Bob's Concise Coding Conventions

I followed Bob's concise coding conventions when writing my code to ensure legibility of my programming. Listed are the 3 main rules that applied to me.

Rule 1: Methods

Methods are 75 lines or less

• Method is visible on a single screen/page.

• Possible to see whole method from start to finish (without scrolling). Exception(s): Methods with case tables (switch statements) and perhaps main method.

Rule 2: Indentation

No methods shall use more than five levels of indentation. Exception(s): none

Rule 3: Line Length

No line of code exceeds 80 characters. It should not be necessary to expand code editor to entire screen width in order to read single line of code. Exception(s): none

(Laramee, 2017)

CHAPTER 6. EVALUATION

6.1 Predicaments

Throughout the project I encountered predicaments and ended up being in some sticky situations. Early in the implementation I bought six 360 degree continuous rotation servo motors instead of the 180 degree servos that I required. The 360 degree continuous rotation servos would not stop at any specified degree but would only change its speed and direction. Fortunately, I was able to return the continuous rotation servos and buy the correct servos.

Another predicament that I had was that I did all the inner implementation of the hand before the 3D printing. A better sequence would have been to do the 3D printing first so that I can base the inner functionality around the design of the hand. This would make the project look neater and would have saved me some time making adjustments.

6.2 Limitations

The quality of the hand is mainly dependent on one component and that would be the flex sensor. The better quality the flex sensors the better the hand would respond to the gestures. The flex sensors that I made were still functional however the professional ones would've made my hand more receptive.

The flex sensors were quite expensive and I didn't have the funds to purchase them. On the bright side, I learnt how to make my own flex sensors and the science behind it was very interesting to learn.

Initially, my plan was to have the haptic feedback in all fingers on the sensor glove. However, the Elegoo Uno only has 6 analogue input/output pins. Therefore, I could only implement the haptic feedback for only one finger. I decided this finger should be the middle finger as the servo would balance best on the glove for that finger.

CHAPTER 7. CONCLUSION

7.1 Project Outline

We began this project with only an Elegoo Uno R3 and through research and hard work; we have made many meaningful achievement to end up with a gesture controlled robotic hand with haptic feedback. The project was broken down into many small manageable pieces which would form a set of milestones. The first was creating the flex sensors. This was probably the most important milestone as it set the path for the rest of the project. The success of the final product was heavily dependent on whether or not I could succeed in creating the flex sensors. Next, I connected to servos and flex sensors to the Elegoo Uno R3 and programmed it so that the flex sensors could control the servos. This step was relatively straight forward and I enjoyed using the arduino IDE. The following step was to control one of the servos by mapping it to values from the force sensitive resistor. This step was quite similar to the previous step and I did not have any problems getting it sorted. Finally, I had to do the 3D printing of the hand and integrate the circuit into the robotic hand and sensor glove. The robotic hand is like the skeleton of the hand and the circuit is comparable to the muscles. All in all, I would conclude the project was a success thanks to the initial research, preparation and prototyping that was done. The results show a fully functional robotic hand that provides haptic feedback and is recognized by the user community through the previous testing and reviewing that I have done. It was an honour for me to undertake this project as I have been given a great insight into the world of robotics and its possible integration with industry, society and even possibly the human body.

7.2 Further Development

Compared to current research and developments of robotic hands, this project can be useful. The robotic hand I have created is more of a portable hand and would most likely be used in industry or hazard prevention. I believe that this project can be developed further. One idea that came to my mind in the early stages was to make the hand more useful to people who are missing a hand. Currently, the robotic hand is controlled through a sensor glove however I believe it can easily be adapted to be controlled in another way. For example, an EEG controlled robotic hand would be much more targeted towards the healthcare side of the user base. An amputee could wear the robotic hand and control it with their brain. This would be a great help for the suffering individuals.

Another way that I can improve my project would be to make a hand that has more degrees of freedom. This would make the hand a lot more dexterous and would make the robotic hand seem a lot more like a human hand. To do this would require printing many more components to simulate the joints inside a hand.

BIBLIOGRAPHY

Power Glove. (2018, April 17). Retrieved May 1, 2018, from Wikipedia: https://en.wikipedia.org/wiki/Power_Glove#cite_note-agetech-3

Arduino. (2018). *Memory*. Retrieved April 23, 2018, from Arduino: https://www.arduino.cc/en/Tutorial/Memory

Average Maker. (2014). *Elegoo Uno R3*. Retrieved 2018, from Average Maker: http://averagemaker.com/2018/03/elegoo-uno-r3.html

Baldor Electric Company. (n.d.). *Baldor Motor and Drives.* Retrieved April 21, 2018, from Baldor: http://www.baldor.com/Shared/manuals/1205-394.pdf

Biomass Packaging. (2018). *The Pros and Cons of Polylactic Acid (PLA) Bioplastic, the "Corn Plastics"*. Retrieved May 2, 2018, from Biomass Packaging: http://www.biomasspackaging.com/the-pros-and-cons-of-polylactic-acid-pla-bioplastic-the-corn-plastics/

Bob, P. (1995). *What's All This P-I-D Stuff, Anyhow?* Electronic Design.

Canadianer. (2015, 09 24). *Biology Stackexchange*. Retrieved 10 27, 2017, from https://biology.stackexchange.com/questions/30857/does-the-human-hand-have-27-degrees-of-freedom

Chadwick, P. (2003, 9 23). *Disability Social History Project*. Retrieved 10 25, 2017, from Disability History: http://www.disabilityhistory.org/timeline_new.html

Chioka. (2016). *How to move a servo with a keyboard*. Retrieved April 21, 2018, from Chioka: http://www.chioka.in/how-to-move-a-servo-using-a-keyboard/

Computing History. (2016). *PAX Power Glove for Famicom*. Retrieved 2018, from Computing History: http://www.computinghistory.org.uk/det/37392/PAX-Power-Glove-for-Famicom/

Digital Trends. (2018). *ABS or PLA: Which 3D printing filament should you use?* Retrieved May 2, 2018, from Digital Trends: https://www.digitaltrends.com/cool-tech/abs-vs-pla-3d-printing-materials-comparison/

e-NABLE. (2016). *The Raptor Hand*. Retrieved April 23, 2018, from Enabling The Future: http://enablingthefuture.org/upper-limb-prosthetics/the-raptor-hand/

Evans, B. W. (2007). *Arduino Programming Notebook.* Creative Commons.

Gingichashvili, S. (2017). *A New Robotic Hand*. Retrieved 10 27, 2017, from http://thefutureofthings.com/5605-a-new-robotic-hand/

Gizmodo. (2017). *The Fascinating Untold History Of War And Prosthetics*. Retrieved March 21, 2018, from Gizmodo: https://www.gizmodo.com.au/2014/05/the-fascinating-untold-history-of-war-and-prosthetics/

Harvill, Y. L., Zimmerman, T. G., & Grimaud, J. J. (1987). *Patent No. 5097252.* United State of America.

Hon, L. C. (2016). *Sensor Controlled Robotic Hand.* Hong Kong: The University of Hong Kong .

Hull, C. W. (1984). *Patent No. 4575330.* United State of America.

Jimbo. (2003). *Force Sensetive Resistor Hookup Guide*. Retrieved May 1, 2018, from SparkFun: https://learn.sparkfun.com/tutorials/force-sensitive-resistor-hookup-guide

Jimbo. (n.d.). *Flex Sensor Hookup Guide*. Retrieved May 01, 2018, from SparkFun: https://learn.sparkfun.com/tutorials/flex-sensor-hookup-guide

Laramee, R. S. (2017). *Bob's Concise Coding Conventions.* Retrieved May 2, 2018, from CS.Swansea: http://cs.swan.ac.uk/~csbob/teaching/laramee10codeConventionSlides.pdf

Mittal, S., & Vetter, J. S. (2015). *A Survey of Software Techniques for Using Non-Volatile Memories for Storage and Main Memory Systems.* IEEE.

Paul, R. P. (1981). *Robot Manipulators: Mathematics, Programming and Control.* MIT Press.

Pliny the Elder, H. P. (1637). *Pliny's Natural History.* London: Printed for the Club by G. Barclay.

ServoCity. (2018). *What Servo Should I Use?* Retrieved April 21, 2018, from ServoCity: https://www.servocity.com/what-servo-should-i-use

Shadow Robot Company. (2017). *Shadow Dexterous hand*. Retrieved April 19, 2018, from Wikipedia: https://en.wikipedia.org/wiki/Shadow_Hand

SparkFun Electronics. (n.d.). *Flex Sensor Hookup Guide*. Retrieved 10 28, 2017, from https://learn.sparkfun.com/tutorials/flex-sensor-hookup-guide

Tech, M. A. (2017, January 2). *Arduino Tutorial - Make a Flex Sensor for Robotic Hand (Cheap and Simple)*. Retrieved May 1, 2018, from Youtube: https://www.youtube.com/watch?v=b7zT94WV-Ek

Visual Dictionary. (2014). *Hand*. Retrieved February 24, 2018, from Visual Dictionary: http://visual.merriam-webster.com/human-being/anatomy/skeleton/hand.php

Wikipedia. (2018). *Fused filament fabrication*. Retrieved May 2, 2018, from Wikipedia: https://en.wikipedia.org/wiki/Fused_filament_fabrication#Fused_deposition_modeling

Woodford, C. (2016). *Resistors*. Retrieved May 01, 2018, from ExplainThatStuff: http://www.explainthatstuff.com/resistors.html

Y, T., Y, H., & K, N. (1972). *Electrically reprogrammable nonvolatile semiconductor memory.* IEEE.

APPENDIX A

Circuit Code

```
#include <Servo.h>

Servo servo1;

Servo servo2;

Servo servo3;

Servo servo4;

Servo servo5;

Servo servo6;


int thumbservopin = 1;

int thumb = 5;


int indexservopin = 2;

int index = 4;


int middleservopin = 3;

int middle = 3;
```

```
int ringservopin = 4;
int ring = 2;


int pinkyservopin = 5;
int pinky = 1;


int gloveservopin = 6;
int fsrpin = 0;


void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  servo1.attach(thumbservopin);
  servo2.attach(indexservopin);
  servo3.attach(middleservopin);
  servo4.attach(ringservopin);
  servo5.attach(pinkyservopin);
  servo6.attach(gloveservopin);
```

```
}


void loop() {
  // put your main code here, to run repeatedly:
int thumbflexposition;
int thumbservoposition;


int indexflexposition;
int indexservoposition;


int middleflexposition;
int middleservoposition;
int ringflexposition;
int ringservoposition;


int pinkyflexposition;
int pinkyservoposition;


int fsr;
int gloveservoposition;
```

```
thumbflexposition = analogRead(thumb);

thumbservoposition = map(thumbflexposition, 1005, 1017, 0, 90);

thumbservoposition = constrain(thumbservoposition, 0, 90);

servo1.write(thumbservoposition);


indexflexposition = analogRead(index);

indexservoposition = map(indexflexposition, 1005, 1017, 0, 90);

indexservoposition = constrain(indexservoposition, 0, 90);

servo2.write(indexservoposition);


middleflexposition = analogRead(middle);

middleservoposition = map(middleflexposition, 1005, 1017, 0, 90);

middleservoposition = constrain(middleservoposition, 0, 90);

servo3.write(middleservoposition);


ringflexposition = analogRead(ring);

ringservoposition = map(ringflexposition, 1005, 1017, 0, 90);

ringservoposition = constrain(ringservoposition, 0, 90);

servo4.write(ringservoposition);
```

```
  pinkyflexposition = analogRead(pinky);

  pinkyservoposition = map(pinkyflexposition, 1005, 1017, 0, 90);

  pinkyservoposition = constrain(pinkyservoposition, 0, 90);

  servo5.write(pinkyservoposition);



  fsr = analogRead(fsrpin);

  gloveservoposition = map(fsr, 0, 1023, 0, 90);

  gloveservoposition = constrain(gloveservoposition, 0, 90);

  servo6.write(gloveservoposition);

  delay(1000);

}
```