A low-cost Human-Robot Interface for the Motion Planning of Robotic Hands

Alice Miriam Howard[1], Emanuele Lindo Secco[2]

[1] Robotics Laboratory, School of Mathematics, Computer Science & Engineering, Liverpool Hope University, Hope Park L16 9JD, UK
19000025@hope.ac.uk, seccoe@hope.ac.uk

Abstract. This paper outlines the design and development process of producing the prototype for the Robotic hand and glove controller with a focus on the design and construction process and emphasis on potential use and future projects. The primary concept of the project is that a user with the control glove can control the prosthetic arm and although the prototype has limited use it creates a good foundation for understanding the basic principles of complex prosthetic arms in terms of design and control, as well as the flex measurement glove demonstrating a wearable control system with further application use.

Keywords: low-cost HRI, wearable sensors, Arduino.

1 Introduction

The first mechanical arms were created in 1959 with the purpose of doing tasks that were harmful to humans and later mass produced for factory automation however prosthetics have been used as far back as Ancient Egypt, two toes that carries at least 40% of the body weight (Mechanical arm, 2017), arms both mechanical and prosthetic have changed vastly over years but their purposes have stayed the same. Due to their uses and implementations these arms have been heavily studied and researched over the years pushing their potential, this producing industrial arms that can work precisely to the millisecond and prosthetics that could one day replace lost limbs, with more and more research pertaining to limb attachment right onto the nerves.

The project we have designed is for the purpose of learning more about Robotic Arms and Prosthetics. We wanted to understand the processes of Arms and Prosthetics and the means of making them alongside how they are controlled. Although our prototype is not an accurate representation of arms it has however enabled better understanding the basic workings of robotic arms both mechanical and prosthetic and by designing and creating our own arm we believe we gained a better understanding then relying on standard research alone as it actively makes me think more about the design and development side which has led to thinking about further projects and innovative ideas.

2 System design

The proposed architecture is made of 3 main elements: a wearable set of sensors or a glove, an open source Arduino platform and a robotic system or hand which allows performing some finger movements mimicking the ones of the aforementioned glove. Figure 1 shows a block diagram of the proposed system.

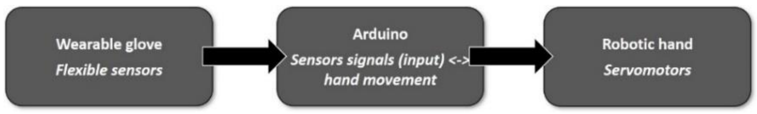


Fig. 1. Block diagram of the proposed system.

2.1 Selection of the parts

In order to design and integrate the system, the following components are detailed (Figure 2).

Arduino board - An Arduino Uno board was used: the board is built around an ATMEGA micro controller which will controls our system. The programming code is uploaded to the micro-controller through a USB port on the board and the Arduino IDE software environment.

Flex sensors - A set of 5 2.2 inch flex sensors model by SpectraSystem are integrated in a wearable fabric or glove (Figure 2): these sensors perform variable resistors vs their flexion since the resistance increases as the sensor is bent. In our project we use these on the glove controller to measure the flex of an end-user's hand.

Foam board - The foam board is a denser type of polystyrene and was used as the primary material in prototyping a robotic human hand. Foam board made a suitable material due to the easy nature of shaping it into the parts required, namely the phalange segments, the palm and the forearm.

Servo motors - Unlike standard DC motors servo motors do not move freely but has an actuator or liner actuator that allows for better control and precise positioning. In this system 5 mini servo motors model from Fitec control the movement of the fingers on the robotic hand.

Other components - Other components are needed in order to finalize the design and assure a proper performance, namely:

- A *nylon wire* attaches to the servo motors and finger so the digits can bend as soon as the servomotors are activated.
- A *thin winter glove* embeds the flex sensors onto them and it is worn by the user to control the system.
- A *set of elastic band* secures the flex sensors down and keep the sensors attached to the human hand phalanges when the subject is moving his/her hand

- A *0.25 mm Gauge steel wire*, which is used to crate channels from the fingers to the servos for the nylon wire
- An *Amphenol FCI Clincher Connector* (2 Position, Female) is also used to connect the pins of the flex sensors to the board

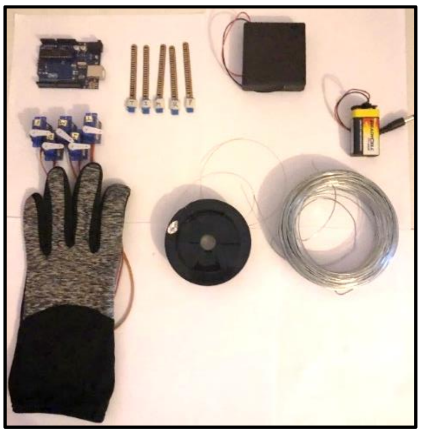


Fig. 2. Overview of the main elements of the system

2.2 Requirements and specifications

Prior to starting the design of the prototype, research was necessary to gain better understanding of the project by looking at previous solutions and understanding of the main drawbacks and advantages of certain technical choice in order to finally obtain a more successful end product.

A. Prosthetic or robotic hand

The first thing that was researched was prosthetic arms, with the way to control the arm already thought out with the use of flex sensors, but it was unknown what the design process of a prosthetic arm is or how it works. It was essential to understand this so when creating this prototype, it could work effectively. It was found that one of the most common ways to control a prosthetic was via *Electromyography* (EMG) (Electromyography & Nerve Conduction Studies, 2018) where the sensor would record the electrical activity of the muscles, by using this control method a more accurate movement could be created alongside the programming to understand better how the muscle was moving, many commercial arms and prosthetics use this method like the Hero Arm by Open Bionics (Secco, 2016-2020).

Design took on two key approaches, one being fully mechanical and use actuators to control the moving parts or using a pulley system with servos and a wire. Both these methods can give the same result but one is simpler than the other, having servo motors pull on a wire to create the contraction is the simpler and cheaper method, having some sort of actuator can become expensive as an actuator will be required at each joint.

Being a prototype using servos in combination with a wire would be the better option for this project.

B. Power supply

It was the desire for this prototype to work fully independently, so it required an external power source, the most common of which was a 9 [V] battery plugged into the Arduino's 5.5 mm / 2.1 DC barrel Plug. An Arduino can operate on a power supply of 6-20 [V] but when the board is supplied with less than 7 V the Arduino's performance can fluctuate and could altogether function incorrectly, this can similarly happen when the board is supplied with 20 [V] as excess voltage will dissipate as heat which is also inefficient. The recommended voltage range is 9-12 [V] this is known as a nice middle ground as the board can easily regulate excess voltage and sufficiently power the I/O pins.

C. Flex sensors

The research needed to involve, how to wire the flex sensor and how to code them correctly but also expand on basic knowledge. Flex sensors are variable resistors so when they are bent it creates a resistance value for example at a bent angle of approximately 45^{o} the resistance value increases to approximately 65 KΩ, and in the terms of this project that resistance value will tell the servo motor how far it should move to pull the arms fingers. The right most pin connects to the ground and the left most pin is for power with the use of a 10 KΩ resistor, connecting the flex sensor to the board we used Amphenol FCI Two Pin Clincher Connector with jumper cables connecting to the board. To implement a flex sensor in code it needs to be established on an analogue pin and with the use of 'analogRead' the information from the flex sensor can be read and the utilized further.

D. Servo motor

There are many differ types of servos to suffice different solutions, the basics of a servo motor is that it is they are rotary actuators that allow precise control over a linear or angular position, acceleration and velocity. Similarly, to the information needed to use flex sensors we needed it for servo motors, the understanding of how to use them correctly was important, the pin setup is simple, servos can cause strain for the Arduino so its recommended to give servos an external power source as it could potentially damage the servo motor. The *Servo.h* library is required to use servos in a program, once the library is imported a servo object is needed (similarly to establishing variables) alongside setting its control pin, *map()* and *constrain()* functions to align value set ranges and parameters which will then be written to the servo motor that will move to its position.

3 Design

Going straight into creating the prototype caused failure, by not having clear thoughts and direction penned out before starting no progresses was made and the project had to be side-lined, going back to the research staged helped as it led to the discovery of the *Engineering Design Process* (EDP) which outlines a series of steps that help to find a solution to a problem and by following this process gave the prototype new and clear direction. Starting again with the help of the Engineering Design Process the prototype had new vision, the prototype was meant to be a learning experience in the design and creation process but as well as learning more about prosthetic.

After a stable foundation of what the end-product should be the design stage could occur, here sketches and measurements were created for a rough idea to how things would look and to where components would fit. This stage was extremely helpful, with a rough idea planned out more accurate measurements of the foam board so not too much of the material would be taken off which could potentially reduce the integrity of the whole piece.

We outline the measurements of the key components by having the measurements of the key components the dimensions of the arm could be established and a more refined arm could be created without excessive material. This also lead to the fingers being more proportional to the arm so not to look too out of place.

When creating the designs for the arm prototype specifically with the design for the fingers the use of a real life hand as well as google images of hands to create a more accurate design. By doing this more thought about the individual aspects of the phalanges, their weight, height and length as well as methods of simulating joins and contractions were considered. A simple spring could function well as a joint in between each phalange and after experimentation with string, a spring and some excess material a successful finger contraction was created by situating the spring in between two small pieces of foam board and attaching string from one end though the other and pulling on it the spring would bend creating the contraction.

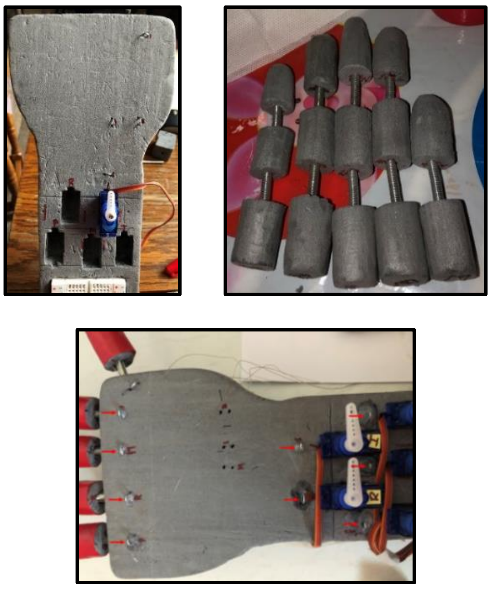


Fig. 3. The arm with servo (top left panel) and fingers (top right panel) and the wire channels for the pull wires (bottom panel)

When designing the arm prototype, a material needed was something that was light weight, malleable and sturdy, many materials were considered but only three were tested; Balsa wood, Polymorph and foam board. Balsa wood was very light weight which was a positive as it meant the springs and pull wire wouldn't have too much resistance from the weight. The downsides of this material was that the Balsa wood was too brittle, when shaping the wood and drilling the wood started to splinter and could not hold the spring in it due to that fact.

The second material that was tested was he biodegradable polyester, Polymorph (Polycaprolactone). Once heated in hot water and left to steep the material becomes very malleable, the Polymorph could be shaped into all the pieces needed and once cooled was very hard and sturdy however, the material became very heavy and once a drill hole was made for the spring the pull wire found it very hard to contract due to the weight.

The last material tested was the one which was used, foam board is light, sturdy and could be carved into the shape required. Foam board is commonly used in modelling, sets and props as it is akin to polystyrene but denser so the use of this material for the prototype seems suitable. Due to the density of foam board more than just basic utility knives were required but these additional tools required to shape and carve the foam board did not need additional purchasing due to already having access to them.

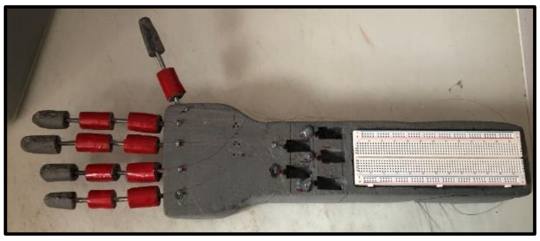


Fig. 4. The full arm design and first prototype.

The arm design

As previously mentioned prior to the construction of the arm a sketch was designed to gain the measurements of the arm, the measurements were calculated from the sized of the key components that needed to fit onto it. After the sketch was recreated on the large piece of foam board the general shape could be carved out of it alongside each phalange segment for all the fingers. After each of pieces of the arm and fingers were carved out slots for each of the five servo motors needed to be made along with the slot for the breadboard, by carving out the slots for these components it would reduce the bulkiness of the arm. the servo motors had a lip on each end of them but choosing not to fit them into the arm to that lip was to reduce the risk of its wire becoming damaged.

To build each of the fingers three phalange segments (only two for the thumb) and three spring (only two for the thumb) were used, each segment had springs to attach it to each other and the spring at the very bottom to attach it into the arm, the springs would act as the join between each phalange segment.

The mechanism to create the contraction was simple, by using a thin wire attached through the finger and once pulled on the wire would contract. When mounting the fingers to the arm some of the fingers didn't line up correctly with their respective servo motors, to correct this by cutting off small lengths of metal wire and shaping them in like a 'U' they could be stuck in the arm to create guide points for the wire to follow to the servo motor.

While researching it was found that the servos power could not come from the Arduino unlike other components could, the servos could potentially damage the Arduino and would cause overheating due to there being too much current on the board, so a separate power source had to be added for the servos to draw off. The Arduino also had a 9V power source connected to it via the DC barrel plug on the side of the controller, and a 6V battery pack was plugged into the bread board for the servo motors. During testing the power was not getting to the servo motors it was unknown why this was but after further exploration into the external power sources and powering servo motors it was found that a 'common ground' was needed (Common ground and why you need one, 2019), after the common ground was created the servo motors no longer has issue.

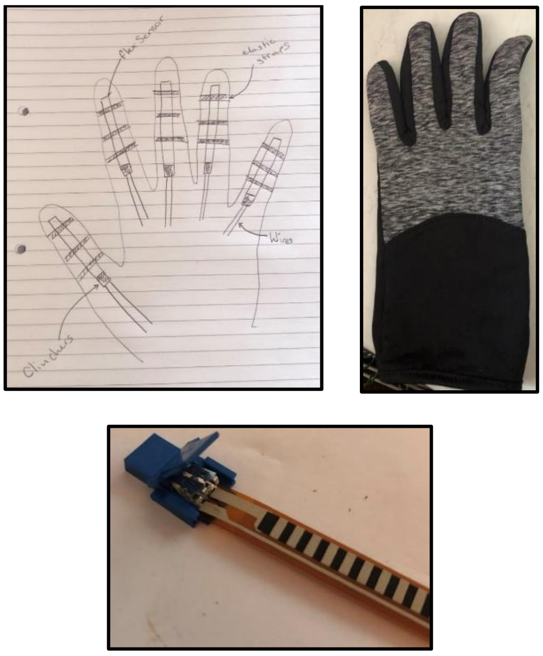


Fig. 5. The control glove sketch (top left panel), the fabric glove (top right panel) and the flex sensor as integrated with the Amphenol FCI Clincher Connector (bottom panel).

The glove design

The glove controls the whole system, by utilizing the flex sensors it can measure the resistance of the wearers hand and mimic those movements on the arm and from the struggles of making the arm the same Engineering Design Process approach was utilized to support a successful outcome. A sketch out design was created similarly to the arm so key components placement can be planned on the glove and decide ways to hold other components.

Choosing the right glove was essential, when choosing the glove different factors came into play, tightness, thickness and material. The glove needed to be comfortable to wear as once everything would be build it could become heavy and not knowing how long the user would be wearing the glove there had to be longevity in the comfort. A thicker glove like a ski glove could maintain comfort and could also comfortably hold all the components on the back. The downside to this however is that the glove itself is thick and bulky and would prohibit clear and precise movements. The glove used in the prototype is a soft winter glove which is thin and flush to the skin, it is both comfortable and functional for the user.

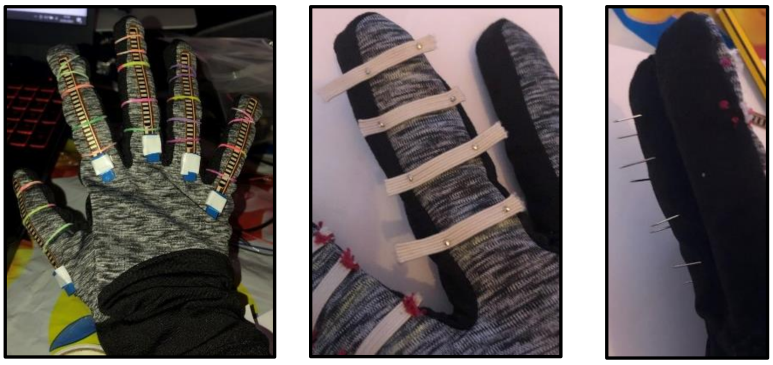


Fig. 6. The flex sensors are attached by using small elastic bands (left panel); the elastic pinned to the glove (central and right panels).

The main component to the glove prototype are flex sensors, these sensors are long thin resistors, when the resistor is bent it creates the resistance, that resistance is converted into an angle that the servo motors move to on the arm prototype creating the mimic response. The flex sensors are attached to the back of each finger and using two position *Amphenol FCI Clincher Connector* to Jumper Pin wires to the breadboard, this was done as access to soldering equipment's was not available but the desire to not limit these sensors to solely this project was the primary factor for this.

To secure the sensor to the finger changed multiple times the main reason being that the methods of attaching them to each finger could be uncomfortable, not secure enough or even dangerous for the user to wear. Cable ties were the first method used but due to the tightness of them caused blood flow restrictions which is dangerous over a period of time. A more effective method of securing the sensors was by using small elastic bands at intervals on the sensor, these elastic bands were both comfortable for the user and would keep the sensor mostly in place. The downside to this method that with excessive movement the sensors has the potential to dislodge out of place or even pop out.

The successful method of attaching the flex sensors to the glove was to use strips of elastic to the back of the fingers and the back of the hand. Attaching them this way meant that the glove was still comfortable to wear as nothing was wrapping around the user's hand. The process of attaching these strips of elastic was to position the flex sensors on the fingers using small elastic bands to hold them in place to mark with tailor's chalk where the small elastic bands were holding the sensor down and then pinning the elastic to the glove ready to be sewn.

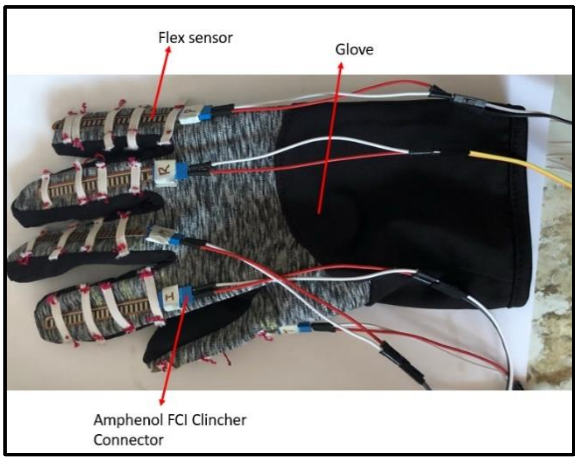


Fig. 7. The wearable prototype of the glove with the embedded flex sensors.

From there the elastic can be cut and sewn to the glove itself so the flex sensors can slide into place, the sensors stay firmly in place even with excessive movement, but to fix any sensors popping out an extra piece of elastic was used below the elastic at the tip.

4 Software integration

The system is controlled by two factors, the implemented code and the user's physical inputs from their and movements. After the immediate setup of the servos and flex sensors the code repeats for each servo with its respective flex sensor translating the resistance value from the flex sensor to the servo as an angle. The coding language chosen was Arduino, Arduino is a free open source software system with its own software tools and compiler. The compiler version of Arduino is 1.8.13 (Windows Store 1.8.42.0). Libraries are used as extensions for a program allowing extra functionality to a sketch which can range from working with hardware to data manipulation.

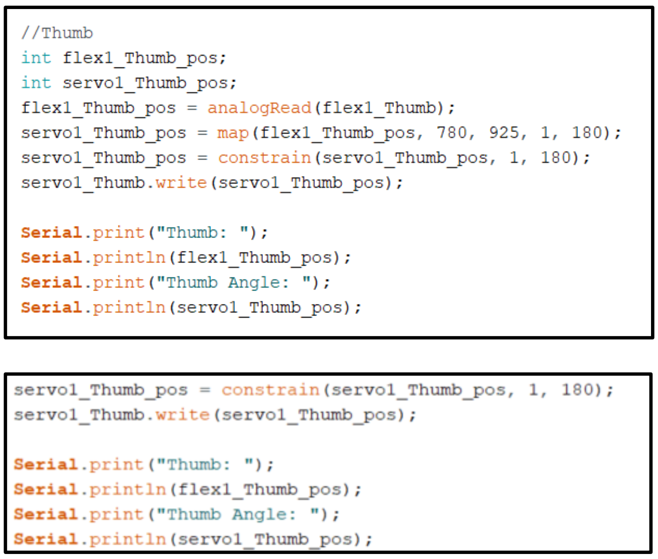
For the servo motors to work it required the "Servo.h" library, this library enables the Arduino board to use up to twelve RC servo motors on one timer. In the prototype the servo motors control the contraction of the fingers, when the user contracts their finger using the glove controller the servo motor will turn to a corresponding angle thus contracting the finger. Objects are the instances of a class, when initialized only then can memory be allocated. Objects are created using the syntax of *ClassName*

ObjectName. Each of the five servo objects are assigned to a digit of the hand, the naming choice is to keep better track of each servo, as the servos have been established data can be assigned to them in the setup and loop functions where they will be attached to a specific pin on the Arduino board and within the loop be used to give the component movement.

Flex sensors run on analogue instead of digital, to set their pin to analogue the syntax of *A* is put before the pin number on the board so to distinguish between from the digital pins. Each flex sensor on the glove is appropriately named to its corresponding position on the glove controller, the data type assigned is integer as the resistance value that they produce is numerical and thus be calculated into the final position for the servo motors.

According to the Arduino Programming Language, anything in the *void setup()* is ran one in comparison to *void loop()*, which is ran until a condition is met. In this setup the serial monitor is enabled so the resistance values from the flex sensors can be viewed the speed of which has been set to the value of 9600 bits per second.

On the contrary, *loops* will run a sequence of events until the specific condition is met, if it is not met then the loop will run forever. For the instance of this program no end condition so the loop will go on forever or until the power supply is cut, the reason for this is that the sensors and servos are constantly updating with new information that the prototype arm is outputting.



```
//Thumb
int flex1_Thumb_pos;
int servo1_Thumb_pos;
flex1_Thumb_pos = analogRead(flex1_Thumb);
servo1_Thumb_pos = map(flex1_Thumb_pos, 780, 925, 1, 180);
servo1_Thumb_pos = constrain(servo1_Thumb_pos, 1, 180);
servo1_Thumb.write(servo1_Thumb_pos);

Serial.print("Thumb: ");
Serial.println(flex1_Thumb_pos);
Serial.print("Thumb Angle: ");
Serial.println(servo1_Thumb_pos);
```

```
servo1_Thumb_pos = constrain(servo1_Thumb_pos, 1, 180);
servo1_Thumb.write(servo1_Thumb_pos);

Serial.print("Thumb: ");
Serial.println(flex1_Thumb_pos);
Serial.print("Thumb Angle: ");
Serial.println(servo1_Thumb_pos);
```

Fig. 8. The loop function for reading the flex sensor (top panel) and the instructions where we constrain the values to the servo motors and display them to the serial monitor (bottom panel).

For each flex sensor and servo motor a variable to hold their position was needed, this was initialized within the loop instead of before with other variables as they were only required there and needed no prior information in the variable. Each of the new variables, *flex1_Thumb_pos* and *servo1_Thumb_pos* (number and digit is changed for each one) hold the position value for the respective component.

After the two new variables are initialized the *flex1_Thumb_pos* will be set with the resistance values from the flex sensor, from there two functions are used, *map()* and *constrain()*. The *map()* function re-maps numbers from one range to another using lower and higher bounds, sometimes lower bound values are larger than higher bound values, what this means is that the function takes an input range and then scales it to an output range for example if the input value was 2 and the rage was between 0 and 10 and mapping it to a new range of 0 and 100 the value retuned would be 20. For returned values not to return higher than the ranges the *constrain ()* function is required.

After *servo1_Thumb_pos* value has passed though *map()* the value is constrained with *constrain()* function so it is not to exceed negatively or past 180 (the maximum value of the servo motors). Once the final value has been calculated then it's given to the *servo1_Thumb*, which controls the servo.

Finally, to see the values, change across time this data can be viewed in the serial monitor. This function of the program was originally used to check the values of the flex sensor so to find the ranges.

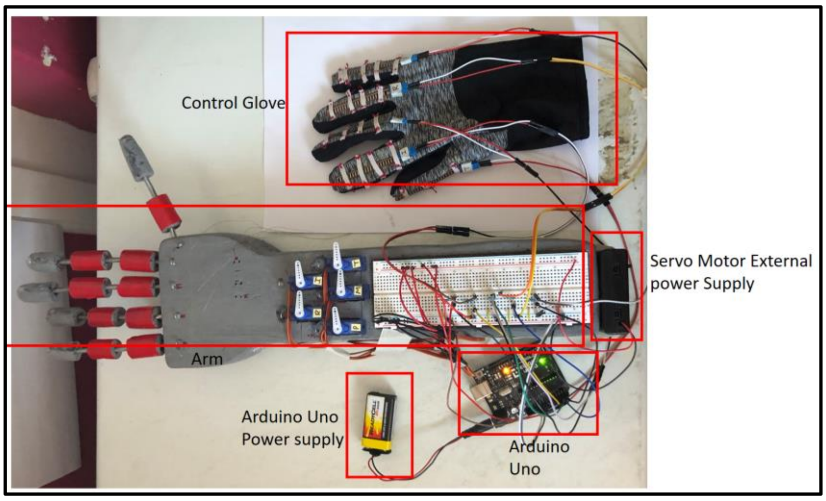


Fig. 9. The overall system completed of the wearable glove with the flex sensors and the robotic actuated device with the 5 servo motors.

5 Testing

To test the prototype, the decision was made to keep the Arduino connected to the computer so the values from the flex sensor to the servo motors could be monitored, the servo motors kept their independent power, a 6 [V] power pack. To begin testing the code was uploaded to the Arduino and power supply established to the servo motor, at this stage of testing what was being looked for was that things appeared in working order, could the servo motors move, could the flex sensors work and was the code correctly mapping and containing values for the servo motors.

The first test was mostly unsuccessful, all the servos moved with the exception of the servo connected to the thumb, this movement happened in what appears like random directions, some moved 180^0, and some moved 90^0 but after that initial movement they stopped, this could be due to a few factors, power problems, the code itself could be wrong and they were moving to their reset position at this stage it was unknown. As the Arduino was still connected to the computer the serial monitor could be viewed, the serial monitor was outputting were constant even when the sensor was not moving and the information had not been updated, the values also looked stage as no matter how much the hand was flex the values stayed between 940 and 960.

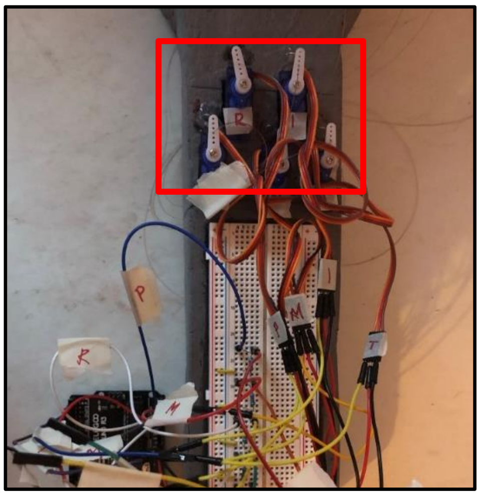


Fig. 10. Servo motors embedded in the arm with insignificant movement.

The 2nd, 3rd and 4th tests yielded similar results, the servo moved again for a brief moment and when tampering with the wires there was a little more movement but it could not be recreated. The servo motor for the thumb still wasn't working. The Arduino was tested without connection to the computer using a 9 [V] battery connected via the DC port, the Arduino lights were on but again still no movement. On the fourth test it was found that the power supply to the thumb servo motor had become disconnected, once reconnected it moved just like the other servo motors.

A further 5th test reviled that the flex sensors were not correctly connected to the analogue pins corresponding to what they have been set at within the code, from this it was clear that the code needed a closer look as one of the flex sensors was completely unconnected, but the serial monitor was producing a high value for it. After this fifth test it was noted that all the flex sensors were not correctly connected to the Arduino analogue pins and had to be corrected. After all visible issues and wire connections had been fixed the prototype still was not functional so each component had to be looked at individually to find the source of the issues.

To check the system reliability and robustness, the following were all tested individually or within partial collaboration with a known good breadboard:

- Arduino
- flex sensors (x 5, namely each one)
- servo motors (x 5)
- power supply
- code

Individually the servo motors and the flex sensors worked and when they worked together individually (one flex sensor and one servo motor together) the desired outcome occurred, but the Arduino Uno board was damaged and two digital pins did not work (all others were in working order) and there was an issue with the external 6 [V] power supply causing an issue on the Arduino. More research into external power supplies was necessary and it was discovered that a common ground was needed as not doing this made the Arduino board buzz and no power provided to the servo motors once this was corrected power to the servos was achieved and in the testing rig the sensor and the servo worked together as intended. This buzzing sound was most likely due to there being a current issue, without correction could cause the regulators in the board to break.

Throughout testing the components, it was clear that the 'map ()' values were incorrect after further research to find the optimal values the flex sensors resistance value had to be values and the higher and lower values could be collected by simply bending the flex sensor. At rest (no bend) created the value of 780 and once bent upwards of 950. The reason to why the high end value is set to 925 and not all the way to 950 was because the servos needed to reach the 180° mark but not instantly creating a full fist to the slightest movement, but due to the low weight strength to the hand reaching the 180° mark was necessary.

The last phase of testing before integrating this with the arm was do incrementally put everything back together testing it along the way. This entails attaching one servo and flax sensor and testing it, and adding the second and then testing them both, leading all the way to testing all five at once to ensure that the board, sensors, servos, power and code all worked.

6 Discussion and Conclusion

The goal of the project was to create a prototype which could be controlled via a glove controller. Despite the prototype arm not function the way as intended, due to a weight miscalculation in the fingers, we feel that this project was still a success as it has taught us an effective design process, researching techniques, bug testing and other knowledge about the components and devices used in the creation of this prototype.

The programming introduced new functions we had not used before and added another layer of interesting challenge the project needing the flex sensors connected to obtain the right values in which to use to the translate the user's movement to the servo motors, this was interesting as it was like live testing and trial and error to obtain the best values for functionality.

Finally, the testing showed a set of bugs but mainly found the main issue of the weight miscalculation so due to this a proof of concept experiment was devised to be able to prove that the system could work despite the weight miscalculation of the fingers holding the prototype back.

Acknowledgements

This work was presented in coursework form in fulfilment of the requirements for the BEng in Robotics for the student Alice Miriam Howard under the supervision of E.L. Secco from the Robotics Lab, School of Mathematics, Computer Science and Engineering, Liverpool Hope University.

References

1. Amazon.co.uk, (2020). [online] Available at: https://www.amazon.co.uk/ [Accessed 12 Dec. 2020].
2. Learn.sparkfun.com. 2016. Flex Sensor Hookup Guide - Learn.Sparkfun.Com. [online] Available at: <https://learn.sparkfun.com/tutorials/flex-sensor-hookup-guide/all> [Accessed 27 January 2021].
3. Store.arduino.cc. n.d. Arduino Uno Rev3 | Arduino Official Store. [online] Available at: <https://store.arduino.cc/arduino-uno-rev3> [Accessed 14 December 2020].
4. Kitronik Ltd. n.d. Mini 180 Degree Resin Gear Servo SG90. [online] Available at: <https://kitronik.co.uk/products/2565-180-mini-servo?_pos=3&_sid=0abae1a03&_ss=r> [Accessed 12 December 2020].
5. Science Buddies. n.d. The Engineering Design Process. [online] Available at: <https://www.sciencebuddies.org/science-fair-projects/engineering-design-process/engineering-design-process-steps> [Accessed 16 December 2021].
6. Forum.arduino.cc. 2019. Common ground and why you need one. [online] Available at: <https://forum.arduino.cc/index.php?topic=653831.0> [Accessed 25 February 2021].
7. EN. n.d. Amphenol FCI Clincher Connector (2 Position, Female). [online] Available at: <https://robosavvy.com/store/amphenol-fci-clincher-connector-2-position-female.html> [Accessed 25 February 2021].
8. En.wikipedia.org. 2017. Mechanical arm. [online] Available at: <https://en.wikipedia.org/wiki/Mechanical_arm> [Accessed 25 February 2021].
9. 2016. Replica of a prosthetic arm. The original probably dates from between 1550 and 1600. [image] Available at: <https://en.wikipedia.org/wiki/Mechanical_arm > [Accessed 25 February 2021].
10. Open Bionics. n.d. Bionic Heroes - Open Bionics. [online] Available at: <https://openbionics.com/bionic-heroes/> [Accessed 25 February 2021].
11. Forum.arduino.cc. 2014. [online] Available at: <https://forum.arduino.cc/index.php?topic=236116.0> [Accessed 25 February 2021].
12. Ee.ic.ac.uk. n.d. [online] Available at: <http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf> [Accessed 25 February 2021].

13. Hobby Components. n.d. nrf24l01 2.4ghz wireless radio transceiver module. [online] Available at: <https://hobbycomponents.com/wired-wireless/156-nrf24l01-24ghz-wireless-radio-transceiver-module> [Accessed 25 February 2021].
14. Thingiverse.com. 2016. Ada Robotic Hand by openbionics. [online] Available at: <https://www.thingiverse.com/thing:1294517> [Accessed 25 February 2021].
15. Marr, B., 2018. 7 Amazing Real-World Examples Of 3D Printing. [online] BernardMarr.com. Available at: <https://www.bernardmarr.com/default.asp?contentID=1554> [Accessed 25 February 2021].
16. Mayfieldclinic.com, 2018. Electromyography (EMG) & Nerve Conduction Studies. [online] Mayfieldclinic.com. Available at: <https://mayfieldclinic.com/pe-emg.htm> [Accessed 25 February 2021].
17. 2017. How to Make Arduino DIY Foam Robot Hand. [video] Available at: <https://www.youtube.com/watch?v=QOyghUxLdqE&list=WL&index=59> [Accessed 1 March 2021].
18. E.L. Secco, C. Moutschen, et al, Development of a sustainable and ergonomic interface for the EMG control of prosthetic hands, 6th EAI International Conference on Wireless Mobile Communication and Healthcare, November 14–16, 2016, Milano, Italy
19. E.L. Secco, C. Moutschen, et al, Development of a sustainable and ergonomic interface for the EMG control of prosthetic hands, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 192, 321-327, 2017, Springer
20. E.L. Secco, C. Moutschen, et al, A sustainable & biologically inspired prosthetic hand for Healthcare, EAI Transactions on Pervasive Health and Technology, 4, 14, 2018
21. E.L. Secco, P. Caddet, A.K. Nagar, Development of an Algorithm for the EMG Control of Prosthetic Hand, Soft Computing for Problem Solving, Advances in Intelligent Systems and Computing, 1139, chapter 15, Springer, 2019, DOI: 10.1007/978-981-15-3287-0_15
22. E.L. Secco, J. Scilio, Development of a symbiotic GUI for Robotic and Prosthetic Hand, Intelligent Systems Conference (IntelliSys) 2020, Amsterdam, The Netherlands