2014

Using Arduino to Design a Myoelectric Prosthetic

Kathleen Talbot
College of Saint Benedict/Saint John's University

Recommended Citation

Talbot, Kathleen, "Using Arduino to Design a Myoelectric Prosthetic" (2014). *Honors Theses, 1963-2015*. 55.
https://digitalcommons.csbsju.edu/honors_theses/55

Using Arduino to Design a Myoelectric Prosthetic

AN HONORS THESIS
College of St. Benedict/St. John's University

In Partial Fulfillment
of the Requirements for Distinction
in the Department of Physics

by
Kathleen Talbot

May 2014

Using Arduino to Design a Myoelectric Prosthetic

Jim Crumley

Dr. Jim Crumley

Associate Professor of Physics

Clark Cotton

Dr. Clark Cotton

Assistant Professor of Biology

Jennifer Schaefer

Dr. Jennifer Schaefer

Assistant Professor of Biology

Tony Cunningham

Dr. Anthony Cunningham

Director, Honors Thesis Program

Table of Contents

Abstract

The loss of a limb can be a very traumatic experience for a person. Prosthetics are devices that can help restore some of the functionality to the user. However, without insurance, prosthetics can be very expensive, creating demand for more efficient and cheaper prosthetics. My research uses an Arduino microcontroller to design a myoelectric prosthetic – a prosthetic that reads electrical signals from the residual limb and powers motors for movement.

Introduction

Prosthetics are devices that substitute a missing or defective part of the body. Prosthetic devices have the ability to restore some, if not all, mobility and functionality to the wearer. These can range from oral prostheses to limb prostheses. Limbs are often lost through trauma, disease, or a congenital disorder. Once the limb is removed, it can be very hard to resume to a normal life. Prosthetic limbs help give the amputee some sort of mobility that they would not have had without the device.

The cost for a prosthetic leg alone ranges from $5,000 to $50,000. Even though some may cost more, they can only withstand about 5 years of wear and tear on average before needing a replacement. Thus, having a prosthetic limb is not a one-time cost. Glen Garrison, director of prosthetics and orthotics at the Hospital for Special Surgery in New York compared owning a prosthetic limb to owning a car. “It can be a pricey thing to work with.” [2]

There are various requirements for an efficient prosthetic limb. In an article published by Marco Troncossi and Vincenzo Parenti-Castelli about design of prosthetic limbs, they created a list of twelve elements in order to satisfy the needs of a patient’s prosthetic:

1. The highest possible dexterity
2. Good performance (in terms of velocity and forces/torques)
3. Appropriate robustness
4. Efficient control

5. A humanlike appearance
6. A light weight
7. Proper size and proportions
8. Good comfort for the wearer
9. Easy control for the amputee
10. Extremely reliable components of the artificial system
11. A low noise level
12. Sufficient autonomy of the energy source to allow the prosthesis to work all day.[3]

Each prosthetic limb is different, and each amputee can consider each of these twelve factors when being fitted for their new limb, however it can be hard to balance all of these factors into one device without increasing the complexity and cost.

The focus of my research was to design a low-cost upper limb prosthetic capable of elbow movement in response to electrical signals from the residual limb. I used an Arduino Uno microcontroller in my design, as well as aluminum rods for the basis of the arm. Ideally, anyone could learn to code the Arduino programming language, and be able to make the arm functional. My Arduino-arm design ended up being very low cost, and the concept can be built upon by anybody with an interest in the project.

I. History

Prosthetic devices date back to the ancient Egyptians. Often times, they used these devices for cosmetic appearance, as well as a “psycho-spiritual sense of wholeness”.[1] They believed that amputation not only affected your current life, but also it affected you in the afterlife. Even after an amputee passed away, they buried their

Figure 1: The big toe of a Mummy located in the Cairo Museum [1]

amputated limb with them to be whole in their eternal life. In the Cairo Museum, there is a mummy on display that is clearly missing the big toe on its right foot, which has been replaced with a prosthetic (Figure 1). Although ancient Egyptians were the earliest known prosthetic wearers and users, the first written account of a prosthetic was between 3500 and 1800 BC in the Rig Veda – an ancient collection of sacred Indian hymns. It told a story of a Warrior-Queen who lost her leg and was fitted with a prosthetic so she could return to battle.[1]

Greek and Romans were found to be the next users of prosthetic devices. In 484 BC, a Persian soldier cut off his leg to escape imprisonment. He then replaced his leg with a wooden prosthesis. In 218 BC, the Roman General Marcus Sergius lost his hand. In order for him to return to battle, he fashioned himself an iron prosthesis which allowed him to hold his shield. He was able to return to battle and fought four times, even when the horse beneath him was stabbed. Later in life, Sergius tried and failed to become a priest because one needed to have two normal hands in order to become a priest.[1]

The Dark Ages and Renaissance period continued the use of prosthetics when needed. During the Dark Ages, prosthetics were made from various materials, such as wood, metal, and leather. The peg leg and hook hand were also introduced in the Dark Ages. These devices provided more motility to the amputee. Unfortunately, during the Renaissance period, barbaric amputations without anesthesia were still taking place. These amputations often led to hemorrhage, and infections were likely. The only people who could afford prosthetic devices were the rich.[1]

Ambriose Paré, born in 1510, revolutionized the treatment and management of wounds. He was a barber-surgeon in the French Army. He aided in various wars, including the campaign of Piédmont from 1537 to 1538, as well as the campaign of Hesdin in 1553. He often said "I dressed him, and God healed him," when explaining his methods. During the siege of Turin, he made his first real discovery. Instead of using cauterization and boiling oil to treat the wounds, he used dressings and soothing ointments. He noticed a significant difference in the conditions of these patients. He included this discovery in his *Method of treating wounds* in 1545 (Figure 2).[1]

THE
Method of Curing VVounds
made by Gun-ſhot.
Alſo by Arrowes and Darts, with their Accidents.
Written by AMBROSE PARIE *of* Laual, *Counſellor and chiefe Chirurgean to the French King.*
Faithfully done into Engliſh out of the French Copie, by
Walter Hamond Chirurgean.

London printed by Iſaac Iaggard, and are to be ſold in Barbican. 1617.

Figure 2: The Method of Curing Wounds by Ambriose Paré – 1545

(http://parthenissa.wordpress.com/2013/08/18/bedside-manners-surgeons-and-the-wounded-romance-hero/)

New amputation techniques formed during the mid-nineteenth century, when doctors began using anesthetics such as chloroform or other ethers.[4] Along with this, various refinements, medicine, and prosthetic science greatly improved surgery and the functional prosthetic device. The Veterans Administration managed amputation centers, including the construction of the artificial limb lab at the Walter Reed Hospital, located in Washington D.C., after World War I in 1918.[4] During World War II, research on body powered prosthetics became popular. Such devices included a system in which a shoulder harness delivers movement through a cable mechanism upon movement of the back. After World War II, most of the research went towards creating myoelectric prosthetics – devices which use the muscle impulses from residual limbs to power their movement. Myoelectric prosthetics are still being researched today, and new ways are being found to read signals from the residual limbs, as well as increase the clarity of the myoelectric signal.[4]

II. Types of Prosthetics

Every amputation is different, and each upper limb prosthetic can be categorized by degree of amputation. For instance, in the upper limb a transhumeral amputation refers to a severing of the arm above the elbow, distal to the shoulder, while a transradial amputation refers to a severing below the elbow. Similarly, the transfemoral and transtibial amputation sever the leg above and below the knee, respectively. Figure 3 illustrates this concept.

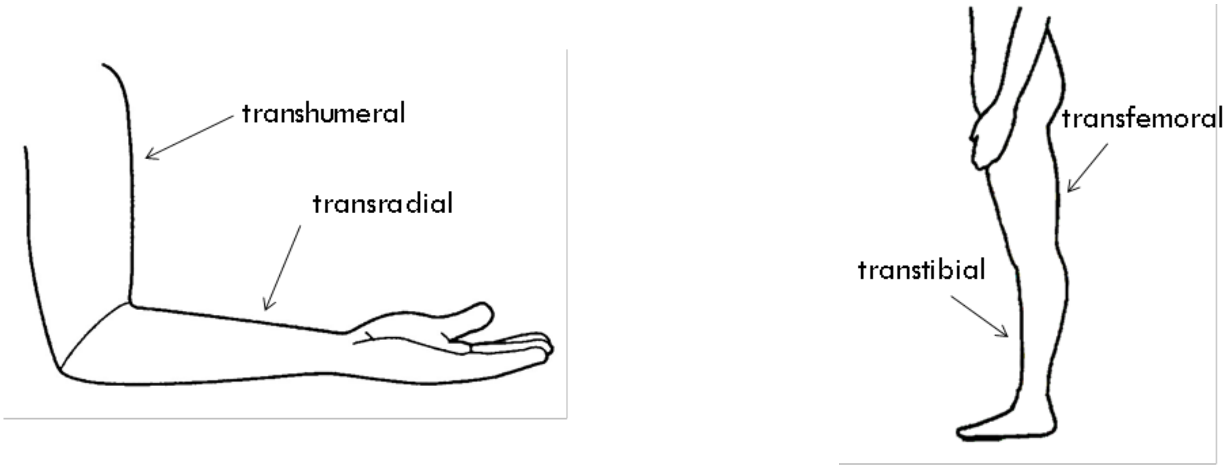

Figure 3: Levels of Amputation. The image on the left shows the levels of arm amputation, while the right shows the levels of leg amputation.

(Arm: http://www.labour.gc.ca/eng/health_safety/pubs_hs/images/forearm-palm-turned-up.jpg)
(Leg: http://www.myaamiacenter.org/MCResources/modules/cse/bodyparts2012/leg/leg_images/empty.png)

The previous categories of amputations set the stages for the types of systems that physicians can use when fitting a prosthetic device. One type of system is an osseointegrated implant for the prosthesis. This means that the prosthesis is in direct contact with the bone. This technique has been in clinical use since 1965. Osseointegration assures a stable, long-term attachment for external prostheses.[5] These types of prostheses help to improve the control of the limb, as well as eliminating several socket-caused skin disorders. Most of these implants are made from titanium because of its durability and the lack of infections that occur with them. Osseointegration, however, is not the only system for prosthetics.

One of the most common systems for prosthetics is the suspension system. These often use a harness, belts, straps, or sleeves in order to suspend the prosthetic limb on the individuals' residual limb. More modern suspension systems have what is known as a suction socket. The Open Prosthetics Project describes the suction socket as follows:

> A suction socket attaches by creating a vacuum between the stump and the prosthesis. As the patient dons the prosthesis, air is expelled from the socket through a one-way valve. The negative pressure around the stump holds the prosthesis in place until the user releases it by opening the valve. [6]

Each of these three systems can be used when trying to design a functional prosthetic device.

There are three types of upper limb prosthetics. The first is a cosmetic prosthetic. This often has little to no functional use. It is often used for cosmetic use, hence its name. It takes on the appearance of whatever part of the arm or hand is missing.[7] Body powered prosthetics are operated by multiple cables and are connected to the residual portion of the body. The advantage of these is that they do not require any electrical power supply. The main disadvantage to these type of prosthetics is that they require an unnatural movement in order to operate them to perform a task. [7] The third type is myoelectric prosthetics. This will be the main focus of my research.

Myoelectric prosthetics are controlled by muscle signals that are given off by the residual limb. Often times, these signals are recorded by using electromyography, or EMG. Surface electrodes are placed on the skin of the residual limb to pick up the signals that are transmitted down the nerves to this limb. This type of prosthetic can often allow the user multiple degrees of freedom, DOF, meaning multiple movements can often be made simultaneously and within a smooth succession.[7] As myoelectric prosthetics have become more and more advanced, they have provided some of the most life-like movements of any prosthetics out there.

III. Prosthetic Movement

Prosthetics have often been used for cosmetic purposes and have had minimal functionality in the past. However, as we continued to modernize, prosthetics began becoming more and more functional. These modernizations have allowed prosthetics to provide several degrees of freedom.

The overall goal of an effective prosthetic is to provide little fatigue, as well as supply the amputee the ability to perform more tasks than it would have previously been able to, with or without an old prosthetic. Some tasks can be more demanding than others. For example, an elbow flexing is pretty simple. It only has two degrees of movement. However, the wrist has at least four degrees of freedom, flexion, extension, adduction, and abduction.

Another way to determine an effective prosthetic is how often the prosthetic is able to repeat a given task. For instance, the waving of a hand is not exactly the same each time, even if it is coming from the same person. There will be slight deviations in how the wrist moves with each wrist movement. An efficient prosthetic wrist will imitate this movement, with only slight deviations from "natural" movement.

IV. How Muscles Contract

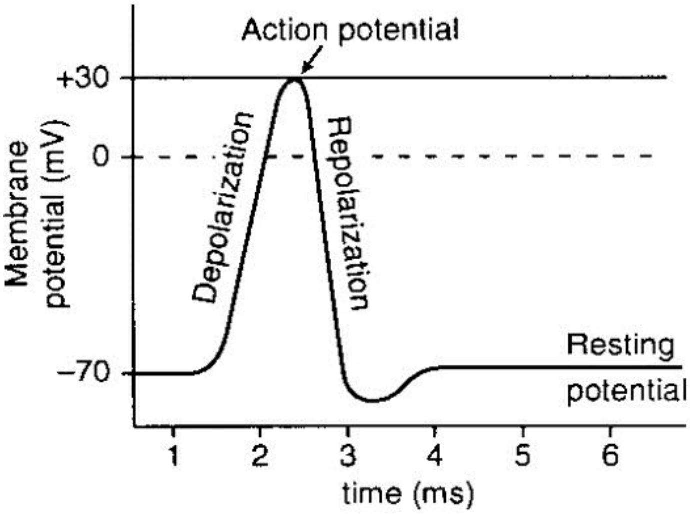

Figure 4: The stages of an action potential

(http://psychlopedia.wikispaces.com/file/view/0199210896.action-potential.1.jpg/324833618/0199210896.action-potential.1.jpg)

A. Action Potentials

The central nervous system in the brain generates what is known as an action potential. This action potential is what stimulates a particular function within the body. In our case, we are picking up action potentials that signal the muscles to move. These action potentials travel down axons. The terminal buttons of these axons release a neurotransmitter that initiates a response in the target cell. [8]

These action potentials consist of a variety of steps within the cell membrane of the axon. A summary of the steps are diagramed in Figure 4.

1. Resting membrane potential occurs when all voltage-gated ion channels are closed.
2. When a stimulus is received, the sodium activation gate opens and the permeability to sodium in the cell increases. Sodium floods into the cell. This causes an explosive depolarization of the cell. This is the rising phase of an action potential.
3. At the peak of an action potential, the sodium inactivation gate closes, and the cell's permeability to sodium falls. This causes movement of sodium into the cell to decease. At the same time, the potassium activation gate opens, and the cell's permeability to potassium increases. Potassium floods out of the cell.
4. As potassium leaves the cell, the cell begins to repolarize towards resting potential. This is the falling phase of an action potential.
5. When the cell returns to resting membrane potential, the sodium activation gate closes, while the inactivation gate opens. This allows the cell to be ready for another depolarizing event.
6. During this time, there is still outward movement of potassium, causing a hyperpolarization of the cell.
7. The potassium activation gate then closes and the membrane is able to fully return to resting membrane potential. [8]

A muscle contraction is performed by using the above mechanism. The EMG signal is then based on reading the depolarization and repolarization events of these action potentials of a muscle fiber.

B. Neuromuscular Junction

Skeletal muscle is composed of many components. A muscle is composed of a collection of muscle fibers, which contain myofibrils. Within these myofibrils are sarcomeres that contain the thick and thin filaments, myosin and actin respectively. Surrounding these muscle fibers is the sarcoplasmic reticulum, which contains an abundance of calcium. Each of these components work in synchrony when a signal is sent to the muscle cell. [8]

An axon of the nervous system is synapsed with the muscle cells. The synapse is known as the neuromuscular junction. When an action potential arrives at the terminal button of the neuromuscular junction, acetylcholine release is stimulated. The binding of acetylcholine on the motor end plate of the muscle cell triggers an action potential in the muscle fiber. This action potential moves across the surface of the membrane to structures known as T-tubules. These T-

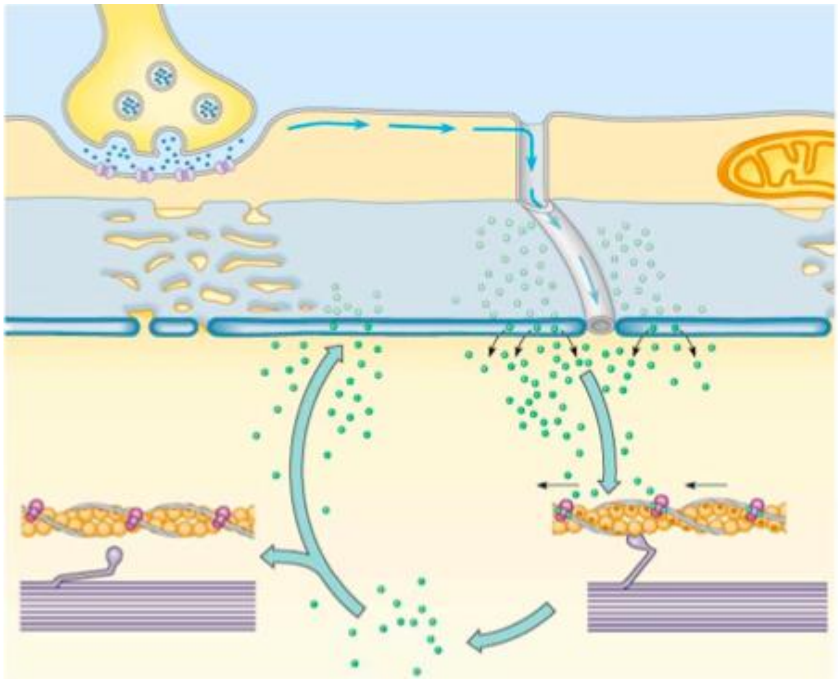

Figure 5: The neuromuscular junction. The green molecules represent Calcium ions. These are released from the Sarcoplasmic reticulum (blue) when the action potential travels down the T-tubule. Calcium interacts with the myofibrils and promote muscle contraction.

(http://classconnection.s3.amazonaws.com/548/flashcards/1531548/jpg/eccicm1336603148448.jpg)

tubules run inside the muscle fiber. They carry the action potential down into the muscle fiber and stimulate release of calcium from the sarcoplasmic reticulum. [8]

When the released calcium binds to troponin on the thin filaments, the troponin changes shape to expose the myosin binding site on the actin of the thin filaments. Myosin cross bridges are then able to attach to these myosin binding sites. Binding of the myosin heads triggers the cross bridges to bend, pulling the thick filament towards the center of the sarcomere. The energy used to power this movement is provided by ATP (Adenosine Triphosphate). After the movement of the myosin, the cross bridge detaches from the actin binding sites. If the calcium still remains present due to action potential, this process is repeated. Figure 6 shows the process of the calcium being released from the sarcoplasmic reticulum and binding to the troponin to change the shape, and myosin heads performing their ratcheting effect.[8]

Once action potentials stop, calcium is taken back up by the sarcoplasmic reticulum. Since there is no longer any calcium on the troponin, it reforms its original shape and covers up the myosin binding sites on the actin. This blocks any myosin from binding. Contraction stops and the thin filaments slide back to their original relaxed positions.[8]

V. Electromyography

Electromyography is a technique used to pick up signals produced by the nerves in target skeletal muscles. These signals are captured by electrodes and sensors and then converted into a digital signal by an encoder. This signal is then processed and displayed by a computer program (Figure 6).

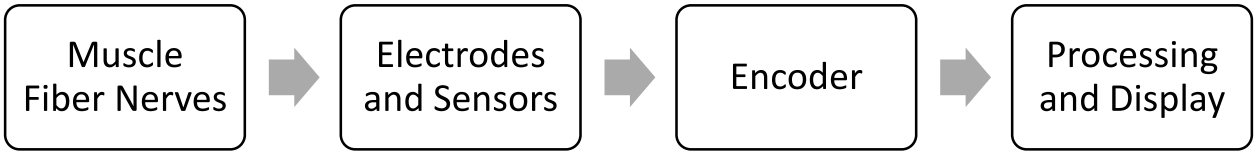

Figure 6: The process of electromyography

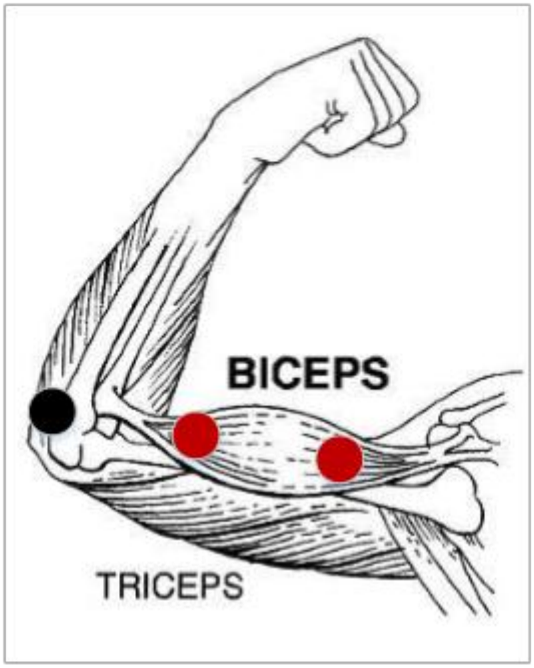

Figure 7: EMG electrode placement

(http://www.orthopaedicsone.com/download/attachments/4030731/biceps.jpg)

The electrodes and sensors are what make electrical contact with the skin. These electrodes are often times directly connected to the sensor. Many modern electrodes and sensors contain a “snap-on” mechanism that allows an easy connection between the two. Two of these electrodes are often placed on the target muscle, while the third is meant to ground the signal and is often attached on or near a bone. For example, if we are to pick up signals from the biceps brachii muscle group, two electrodes would be placed on the upper and lower portions of the muscle, while one electrode is placed near the elbow to ground the electrodes. When action potentials occur in the muscle fibers, the electrodes receive this spiking activity. The EMG is an electrical recording, not mechanical. The strength of muscle contraction corresponds to the size of the signal strength, which in this case corresponds to the amount of voltage output.

The processing software often times converts the raw EMG information into a smooth curve. To do this, the software often uses what is called the Root Mean Square, or RMS.[9] The RMS coincides with the mean power of the signal. This now smooth graph of the EMG signal can now be analyzed. An example of the RMS method is shown in Figure 8.

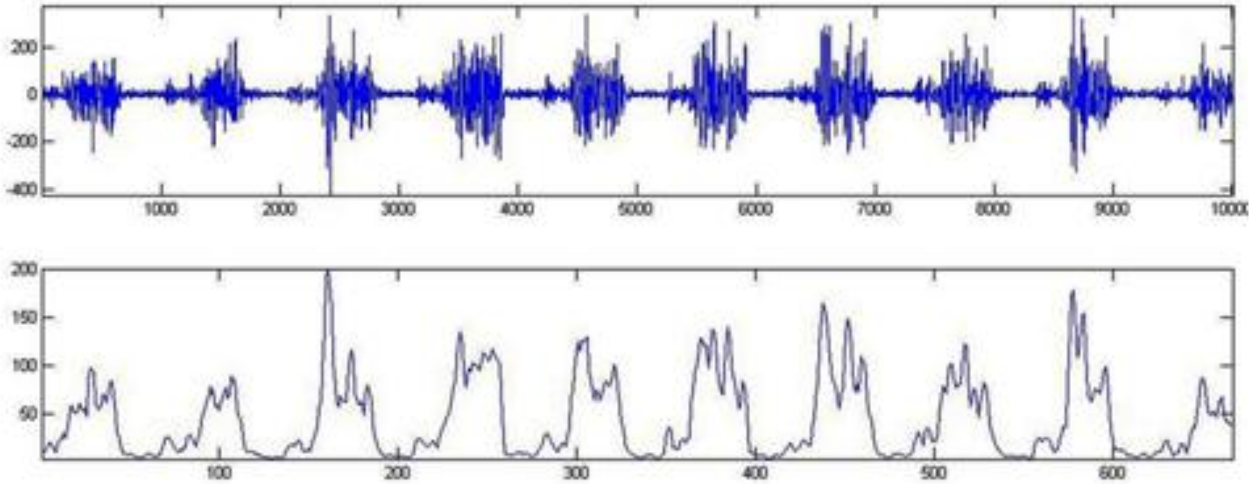

Figure 8: Example of raw EMG conversion to its root mean square (RMS) equivalent

(http://1.bp.blogspot.com/-podnGYNuTaY/T8WR-XK7ASI/AAAAAAAAAD8/76dr0fxpDGw/s1600/ideal1.jpg)

VI. Myoelectric Prosthetics

A. Non-Pattern Control

As stated before, myoelectric prosthetics use EMGs in order to turn nerve signals into a desired movement. Non-pattern control myoelectric prostheses use only the nerve signals to generate movement. These prostheses use electrical motors, often powered by a rechargeable battery, in order to power the desired movements.[10] Myoelectric prosthetics have several advantages, including the relative ease of use, comfort, and an incredible promotion of building muscle tone.[11] The residual limb tissue affecting the amount of noise on the EMG signal is among the disadvantages of myoelectric prosthetics.[11] Regardless of disadvantages, myoelectric prosthetics are some of the most researched prosthetic devices in the world today.

Myoelectric prosthetics have various types of functions. One type uses EMG signals to estimate the amount of force an intact limb would have produced, and in turn, allows the prosthetic to move with a similar force. Even though movement like this is ideal, it also has some limitations. An electrode placed on a particular area might not target one specific muscle. [12]

However, there may be other muscles conducting signals that can be picked up by that electrode and create an alteration from the ideal response from the muscle. Along with this, in order to measure an average power signal of a muscle, several electrodes must be placed to measure an average.[12] However, having several electrodes runs counter to the purpose of creating a simplistic model for force output.

B. Pattern Control

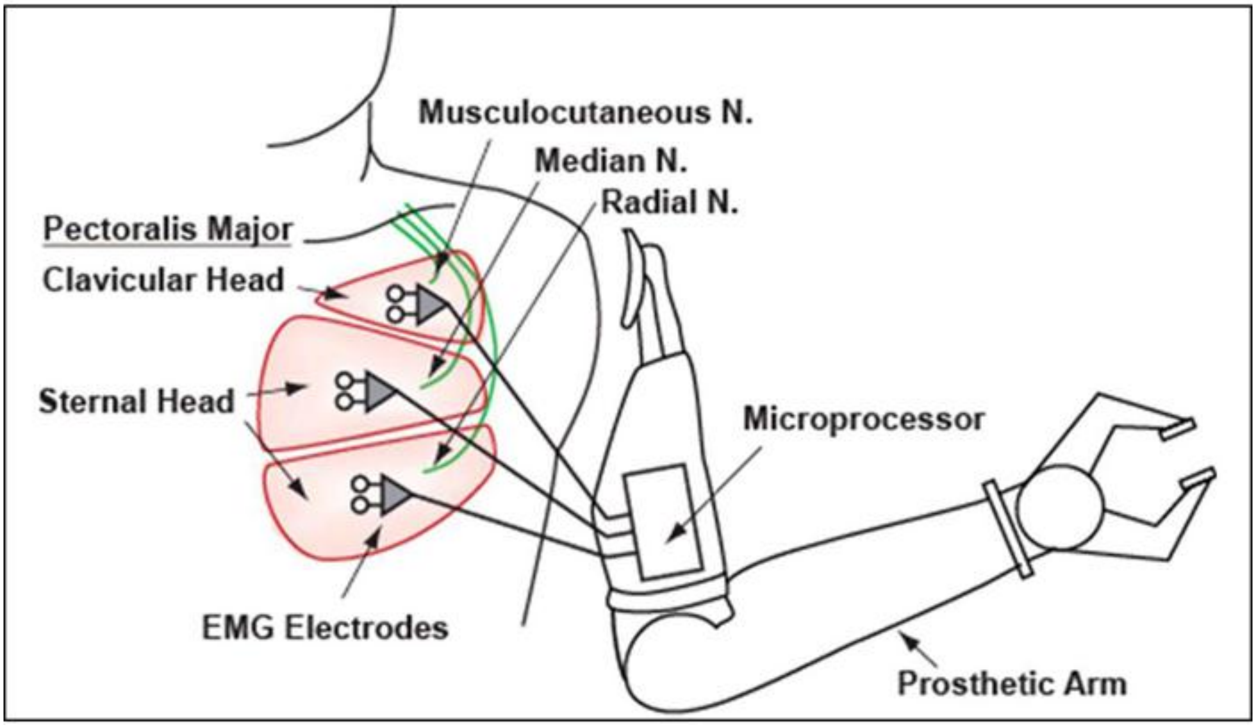

Figure 9: Target Muscle Reinnervation [14]

Another form of myoelectric prosthesis is created by using pattern recognition. This often requires rigorous training and a multitude of electrodes in order to get the desired motions, however, pattern recognition prosthetics often provide the user the ability to perform multiple motions in one smooth succession.[13]

In order to use one type of pattern recognition to control the prosthetic device, a procedure called target muscle reinneravtion (TMR) is often performed. This procedure often transfers selective nerves of the brachial plexus to new muscle sites (Figure 9). Relocation of these nerves depends on the type of amputation, however, it has been found that this type of procedure and prosthetic device works best with patients with a shoulder disarticulation or transhumeral amputation. With a shoulder disarticulation, the median, radial, musculotaneous, and ulnar nerves are transferred to the pectoral muscles.[14] In a transhumeral amputation, the median nerve is transferred to the medial biceps, and the distal radial nerve is moved to either the brachialis or lateral triceps.[14] From here, pattern recognition is used to move the prosthetic. The patient must go through a vigorous training regimen in order to train themselves to flex particular muscles, of which they are not used to flexing, in order to move their new prosthetic. A newly developed program to help the training process is called The Acquisition and Control Environment (ACE). This program is a flexible MATLAB-based environment that among many features, allows for configuration of the prosthetic device.[14] With some training and practice on this system, the patient would ideally be able to move their new prosthetic in a fluid like movement, almost as if they had not lost the limb to begin with.[14]

Materials and Methods

I. Materials

An Arduino Uno microprocessor was the basis of the materials for this design. The Uno is relatively small and quite functional. Attached to the Arduino is an Adafruit Motor Shield (v2). The motor shield is able to drive DC and stepper motors much more efficiently than using Arduino alone.

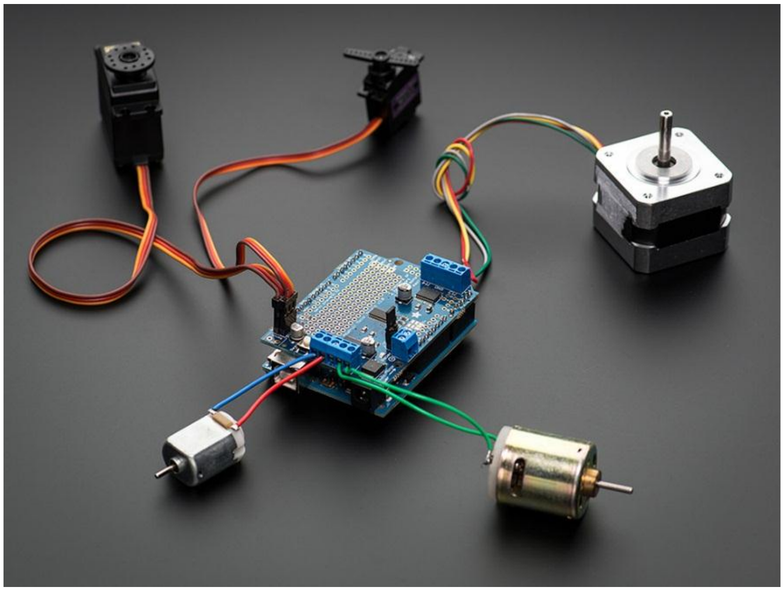

Figure 10: Arduino Uno Motor Party Pack (v2)

(http://www.adafruit.com/product/1438)

The shield has the ability to drive up to 4 DC motors or 2 stepper motors at one time, along with an ability to stack if multiple motors are required. If you were looking for a more functional prosthetic with more motors, the Adafruit Motor Shield would be able to provide this.

In addition to the motors supplied in the Adafruit Motor Party Pack (6 volt DC hobby motor, small stepper motor, standard size servo motor, and micro-size servo motor), a 68oz and 125oz stepper motors were purchased from SparkFun, an online retail store that specializes in “making electronic projects possible.” These motors were purchased under the impression that the one supplied in the party pack would not have been able to support the arm. Stepper motors are nice because with each “step,” they move a fixed angle. These motors produce a lot of torque and are able to move heavier objects than a servo motor. One downside with the stepper motor is the fact that it does not detect if the object does not move; it just keeps going until the Arduino program tells it to stop moving. However, this type of motor seemed reasonable for purposes of the project.

The metals used in the construction of the arm are a 1-inch and ½-inch diameter aluminum rod. I went with aluminum since it is extremely malleable, flexible, and lightweight. For prosthetics, being lightweight is a key component, or at least life-like. The elbow was chosen because it was the simplest of movements, flexion and extension, compared to the wrist or fingers, which have a vast amount of movements. The elbow joint created the ½-inch aluminum rod and specially designed hinges created out of a thin sheet of aluminum.

In order to read the electric potential from the muscles of the arm (EMG), a Muscle Sensor Kit was purchased from SparkFun. This kit includes a special microcontroller, with electrode hook-ups. The electrodes are place in the same location as described previously. Ideally, the microcontroller would pick up the electrical signals from the arm and relay the signals to the motors for movement. The Muscle Sensor, however, does not send the raw signals from the muscles to the motors. The sensor converts the signal into a rectified and smoothed wave as described previously with RMS, such that it can work well with the Arduino's Analog-to-Digital Converter (ADC).[15] This is conversion is illustrated in Figure 11. The ADC converts the analog voltage into a digital signal to be processed.

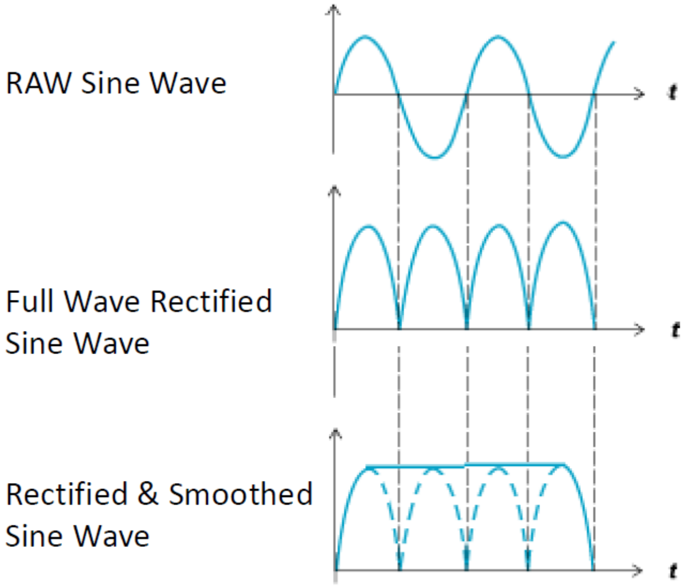

Figure 11: Conversion of Raw Sine Waves to the Rectified and Smoothed Sine Wave by the Muscle Sensor [15]

II. Design

The design of the arm is very basic. It consisted of a nine-inch section of the 1-inch aluminum rod, while the forearm was a twelve-inch section of the ½-inch aluminum rod. The lengths of the rods were estimated by measuring my own upper arm and forearm lengths. The elbow was specially designed since I had a hard time finding the kind of elbow joints that I was looking for. Using the thin sheet of aluminum, I cut out a special design shown in Figure 12. This design was placed on both the upper arm portion, and the lower arm portion in the orientation shown

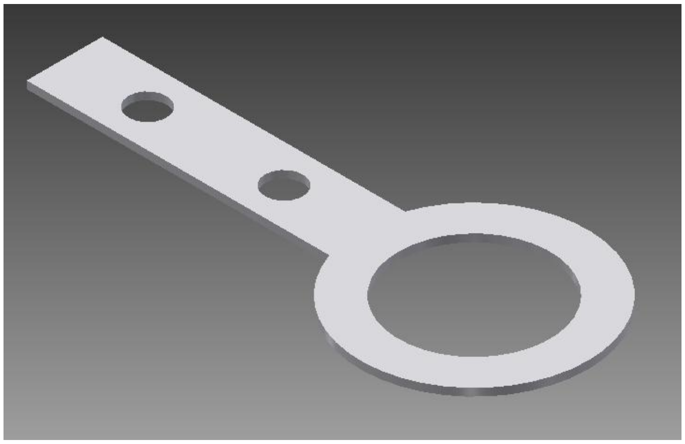

Figure 12: Elbow hinge

in Figure 13. When the ½-inch aluminum bar is placed in the holes created by this piece, a hinge joint is created, much like that of the elbow in the arm.

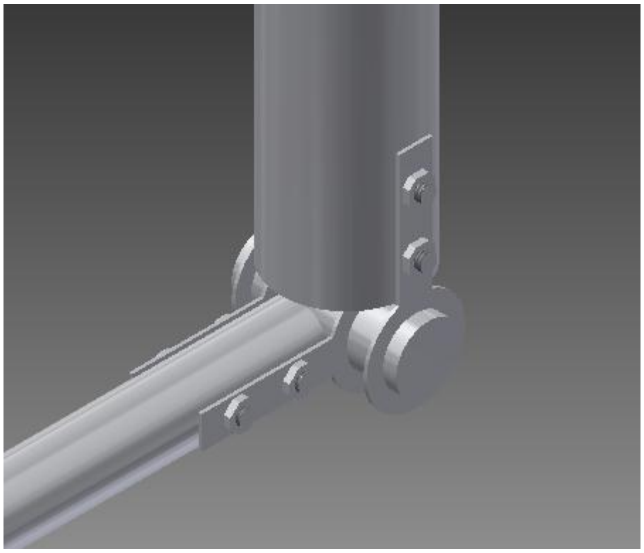

Figure 13: Final elbow hinge

A wire would be connected to the stepper motor and then connected to a position on the forearm. Once a signal for contraction was received, the stepper motor would begin rotating, thus pulling the forearm upwards, creating a smaller angle and simulating flexion. When relaxation begins, the motor would move in reverse, thus returning the arm to its original position. The initial algorithm can be found in Appendix 5.

III. Why Use Arduino?

According to the Arduino website, Arduino is "an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software."[16] Arduino can be used to make interactive objects or environments. Often times, the Arduino microprocessor is connected to a breadboard which houses the various components of the circuit, such as resistors and capacitors. The Arduino can read input from a variety of sensors, such as touch, and can control various outputs, such as sound, light, and motors.

The Arduino programming language is loosely based on the C programming language. There are four basic elements to the language: expressions, statements, statement blocks, and function blocks. Expressions combine operands and operators. This can be used to set variables, add to functions, etc. Statements complete an instruction desired by the programmer. Statement blocks consist of one or more statements grouped together so they are viewed by the compiler as though they are a single statement. Often times, these are also known as loops (if, else if, for, while, etc.). The function block, also known as a method, is a block of code designed to accomplish a single task.[17] What sets Arduino apart from various other microcontroller boards is its ease of use, rather cheap cost, and extreme functionality. This is why I went with using Arduino. Plus, I had a small background in the programming language and its multiple uses.

Results and Discussion

Upon starting this project, I was very ambitious. My initial idea before proposing the project was to work with creating a hand. However, with as many special movements the hand and wrist does, the elbow was more simplistic. Even then, the elbow proved to have some difficulty when it came to the coding aspect.

I first began by wiring up the Motor Shield in order to get the stepper motors moving. I ran into some problems with this because the stepper motors just seemed to vibrate instead of actually rotate. This problem was easy to fix, though. Stepper motors have two coils inside them which help them “step”. Each step rotates a particular set degree in one direction, depending on the stepper motor itself. In the motor I was using, there are 6 wires, 2 for each of the coils, and the other two could be ignored at the time. The wiring I had was incorrect in that I had the one wire from each of the coils wired together, instead of each individual coil wired together. Once this problem was fixed, the motors were able to move without any problems.

The Muscle Sensor proved to have some difficulty. Initially, the wiring was the problem. Appendix 1 shows how the Muscle Sensor microcontroller should be wired to the arm, as well as to the Arduino Uno microcontroller. Upon hooking up the batteries to the microcontroller, as shown in the image in Appendix 1, the microcontroller got very hot, to the point of not being able to touch it anymore, and a burning odor came from it. This was only one of the many complications with the microcontroller.

In the processing Arduino code, there is an import for the “processing.serial” library. This allows for the Arduino to “talk” with the other microcontroller, and display or save the EMG data. However, when trying to compile the program, there was a processing.serial error, specifically, it was not allowing me to import the processing.serial library into the Arduino program. Without this library, I would have no way to visualize the results of the muscle contractions. After extensive troubleshooting, I was not able to resolve this probem.

Since the Muscle Sensor was not working very well, I focused more on the motor movement itself. Ideally, I wanted to have the motor “flex” the arm a certain degree, and then move back to its original position after a certain amount of time. There are various example codes in the Arduino library that assisted me in working with this undertaking. By using the Adafruit Stepper Library (<AFMotor.h>), I was able to set the number of steps in order to achieve the desired distance that the arm would move, as well as be able to release it back to its original position after a given time.

Had time permitted, I would have run an experiment using a number of subjects to measure the EMG activity from their biceps brachii muscle. Each subject would be hooked up to a LabPro EMG sensor. This is completely non-invasive. Surface skin electrodes would be placed on the biceps brachii muscle group of each of the subjects. These electrodes are then connected to the LabPro interface, which is connected to the computer. The subjects would then have been asked to relax their arm with a 5 to 15 pound weight in their hand. When prompted, the subjects would then count to three as they curl the weight up towards their shoulder and hold it for a few seconds, before curling down for three seconds. These steps would be repeated about ten times per subject.

Once all the data was collected, the EMG would be converted to relative force output. I would have found an average amplitude for contraction force and checked for statistical significance. This information would have been imputed into the Arduino Processing code provided by the Muscle Sensor microcontroller. This information would then have been read by the Arduino, and once the muscle contraction exceeded that average amplitude, the motor would move until contraction stopped. Upon relaxation of the muscle, the motor would have moved in reverse and the arm would have moved back to its original position. However, since the Muscle Sensor microcontroller did not work out as planned in the time I had for the project, this part of the project became unnecessary, but the data could be collected for future research with the Arduino microcontroller and Muscle Sensor.

In the end, upon completion of my project, I was able to come up with a design, and beginnings of an Arduino code that could be easily adapted once the Muscle Sensor is in working order. I also was able to cut out various pieces of the arm, but time permitting, was not able to put it all together.

Conclusion

Overall, the project shows how versatile Arduino can be. The concept behind the project holds, in that using Arduino provides a cheap and effective way to create or tweak a myoelectric prosthetic. The current design I created for the elbow right now is basic and can be upgraded with a little imagination and hard work.

The background research that I did with Arduino proved that the concept and idea was possible. Several people have used the Muscle Sensor Kit in order to use their muscles to control a device. For example, a young gentleman was able to code the Muscle Sensor into reading the contraction of his bicep muscle in order to move the character Mario in Super Mario Bros. AdvancerTechnologies, the developers of the Muscle Sensor, posted a video on how to create a replicate Iron Man's armor by using their sensor attached to an UnderArmour arm sleeve. The sensor picked up the contractions of the muscles and displayed lights and played sound with each contraction, making it seem like you were Iron Man himself.[15]

The metals that were used were relatively cheap, roughly \$25 and \$10 for 4 feet each of the 1-inch and ½-inch diameter aluminum rods, respectively. From the metal purchased, about 4 arms could have been made if I so chose to. The Arduino itself, along with the motors, shield, and sensors, total roughly \$200. As long as a computer is available, and battery power for the Arduino and the Muscle Sensor, a myoelectric arm can be built and created for under \$300. This is an amazing reduction in cost compared to the \$5,000 to \$50,000 for a prosthetic that lasts five years, at most. The Arduino arm may not be as sturdy or versatile as these more advanced prosthetics, but if you are on a budget, or health insurance does not cover enough of an advanced prosthetic, and you are looking for something more life-like than having the residual limb, then Arduino's versatility can aide in your search for a design of a new limb. The Arduino software is very open, and anyone with a creative mind can put to great use. There can be a huge field of work out there where people can tweak how their arm works by a simple tweak in the Arduino code.

Future Work

There are several modifications to improve the design that I created for the myoelectric prosthetic arm. First, a more secure elbow joint could be created. There are various metal pieces can be found in hardware stores, or even designed, to create a more functional elbow. Recall that my elbow only consisted of cutting a special design out of a thin sheet of aluminum and attaching it to the sides of the aluminum rods (see Figures 12 and 13).

Second, a further understanding of the Muscle Sensor should be had in order to create the myoelectric aspect of the prosthetic. The idea is out there, and it is possible to do so, however, I was not able to get this to work. For this to work, a threshold should be set on contraction so when that threshold is reached, the myoelectric arm will move.

Along with an understanding of the Muscle Sensor, there are different ways that one could read the input signals. For example, you could translate strength of contraction into how far the arm moves, or is able to hold its position with resistance pressing against it. Another example is designing pattern recognition software for Arduino, to make the arm more state of the art and functioning better.

Finally, besides reducing the cost of the arm, even though it is low cost as is, more research can be done on increasing the amount of functionality in the arm. For instance, creating a hand or a hook that works in synchronous with the arm. Since Arduino is so versatile, many projects can be deduced from this one idea. This research project is very open ended, and with the right imagination, can be taken anywhere.

Acknowledgements

I would like to thank Professor Jim Crumley for introducing me to Arduino my junior year in our Research Seminar for MapCores. Without this, I would have probably been doing a different thesis all together, and not seen the incredible openness of Arduino, and the different projects that can be created with a little imagination. Along with this, Professor Crumley helped give me feedback on my research, and helped debug some of the Arduino problems when they arose since not many knew about Arduino. Thank you to Professor Clark Cotton and Professor Jennifer Schaeffer for being on my defense committee. Finally, I would like to thank all of the MapCores professors, Professor Jim Crumley, Professor Kris Nairn, and Professor Lynn Ziegler for being encouraging and supportive these past four years.

Bibliography

1. Thurston, A. J., Paré and Prosthetics: The Early History of Artificial Limbs. *ANZ Journal of Surgery* 2007, *77* (12), 1114-1119.

2. Mohney, G. Healthcare Costs for Boston Marathon Amputees Adds Up Over Time 2013. http://abcnews.go.com/Health/health-care-costs-boston-marathon-amputees-add-time/story?id=19035114.

3. Troncossi, M.; Parenti-Castelli, V., Synthesis of Prosthesis Architectures and Design of Prosthetic Devices for Upper Limb Amputees. In *Rehabilitation Robotics*, Kommu, S. S., Ed. Itech Education and Publishing: Vienna, Austria, 2007; p pp. 648.

4. Ott, K.; Serlin, D. H.; Mihm, S., *Artificial Parts, Practical Lives: Modern Histories of Prosthetics*. NYU Press: 2002; p 359.

5. Tillander, J.; Hagberg, K.; Hagberg, L.; Mranemark, R., Osseointegrated Titanium Implants for Limb Prostheses Attachments: Infectious Complications. *Clinical Orthopaedics and Related Research* 2010, *468* (10), 2781-2788.

6. The Open Prosthetic Project. openprosthetics.org.

7. Harker, H.; Wolf, T., "Reinnervating" The Prosthetic: Using Targeted Muscle Reinnervation to Improve Upper-limb Myoelectric Prosthetics. University of Pittsburgh: Swanson School of Engineering, 2012; Vol. Session B3, p 8.

8. Sherwood, L., *Human Physiology: From Cells to Systems*. 8th Edition ed.; Cengage Learning: 2012.

9. Ltd., T. T., Basics of Surface Electromyography Applied to Physical Rehabilitation and Biomechanics. *Thought Technology Ltd* 2009, (MAR908-03).

10. CL., P.; JM., L.; RF., K., Electromyogram-based neural network control of transhumeral prostheses. Journal of Rehabilitation Research and Development, 2011; Vol. 48, pp 793-54.

11. Corbett, E. A.; Perreault, E. J.; Kuiken, T. A., Comparison of electromyography and force as interfaces for prosthetic control. *Journal of Rehabilitation Research & Development* 2011, *48* (6), 629 - 642.

12. Doerschuk, P. C.; Gustafson, D. E.; Willsky, A. S., Upper Extremity Limb Function Discrimination Using EMG Analysis. *IEEE Transactions on Biomedical Engineering* 1983, *BME-30* (1), 18-29.

13. Hargrove, L. J. A Comparison of Surface and Intramuscular Myoelectric Signal Classification. University of New Brunswick, 2003.

14. Scheme, E.; Englehart, K., Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use. *Journal of Rehabilitation Research and Development* 2011, *48* (6), 643-660.

15. AdvancerTechnologies, Three-lead Differential Muscle/Electromyography Sensor for Microcontroller Applications. 2013.

16. Arduino - Homepage. http://arduino.cc/.

17. Jack Purdum, P. D., *Beginning C for Arduino*.

Appendix I: Muscle Sensor Setup

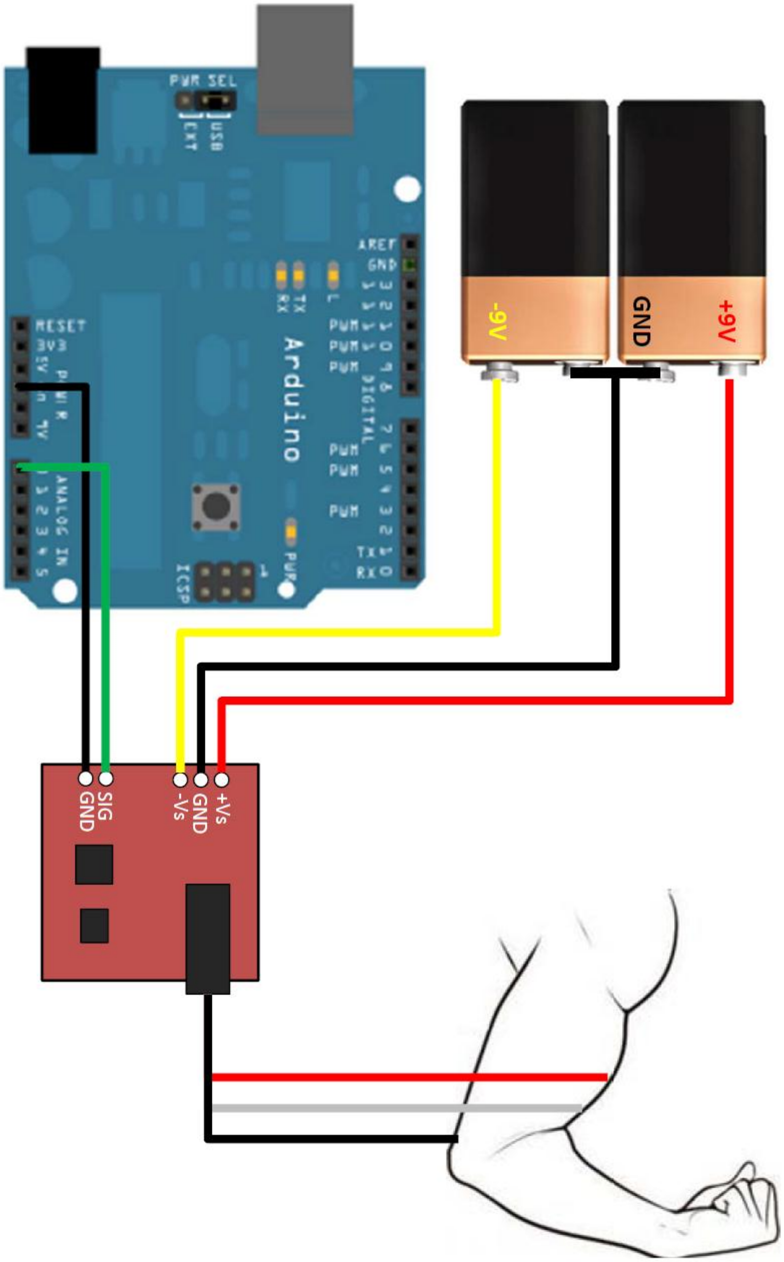

Figure A1: Muscle Sensor hookup diagram for Arduino Uno. Two 9-volt batteries are connected together and grounded on the sensor. The positive and negative ends of the batteries are wired to +Vs and –Vs, respectively. The sensor is also grounded on the GND pin on the Arduino Uno, and the SIG hookup gets connected to ANALOG IN 1. There is a hookup on the Muscle Sensor microcontroller that allows the electrodes connected to the bicep to be connected to the sensor itself.[14]

Appendix 2: Muscle Sensor Schematic

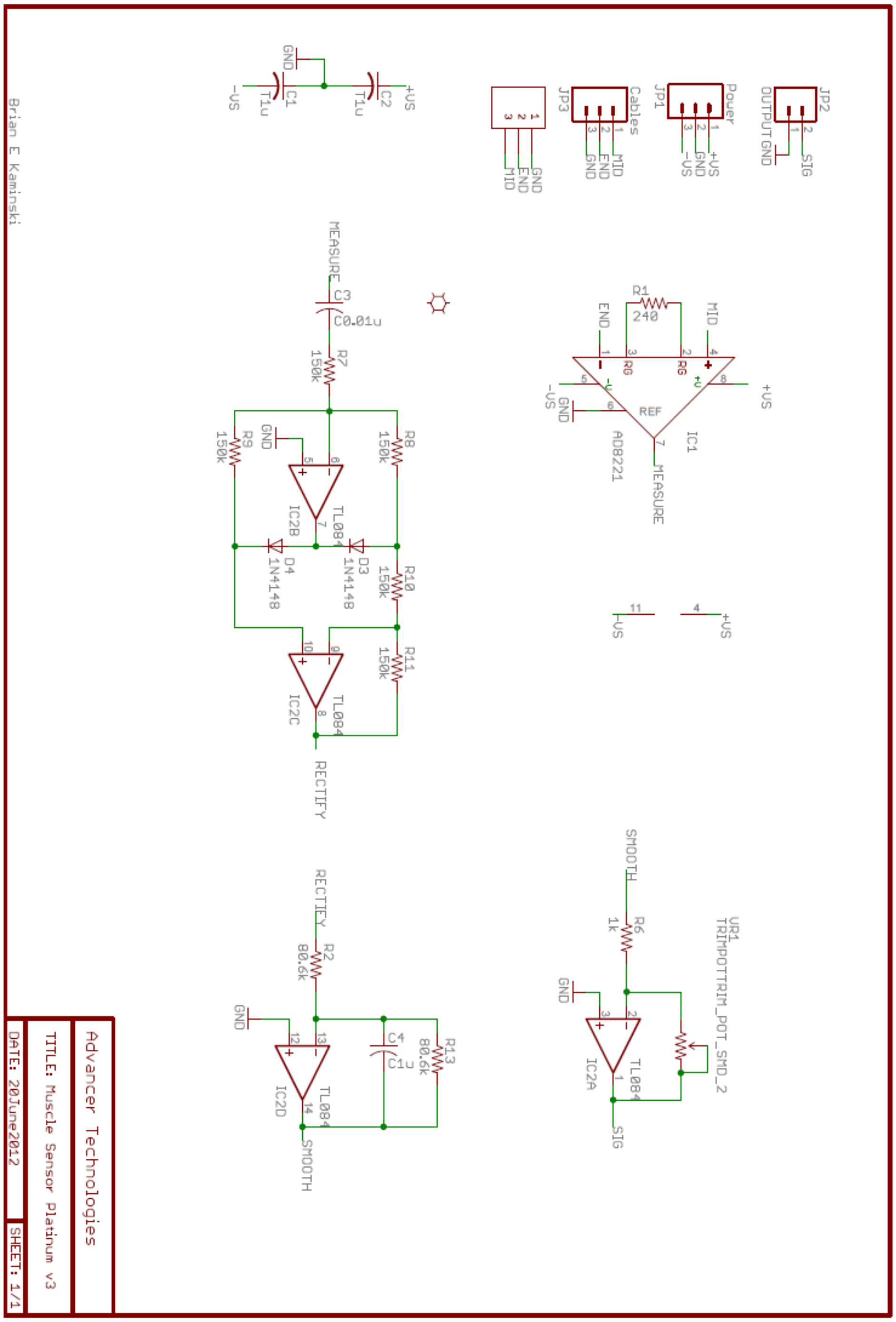

Figure A2: Muscle Sensor Schematic from AdvancerTechnologies. This schematic provides a working understanding of how the muscle sensor works. It uses various sets of comparator in order read electrical signals from the muscle and relay the information to the Arduino Uno.[14]

Appendix 3: Arm Schematic

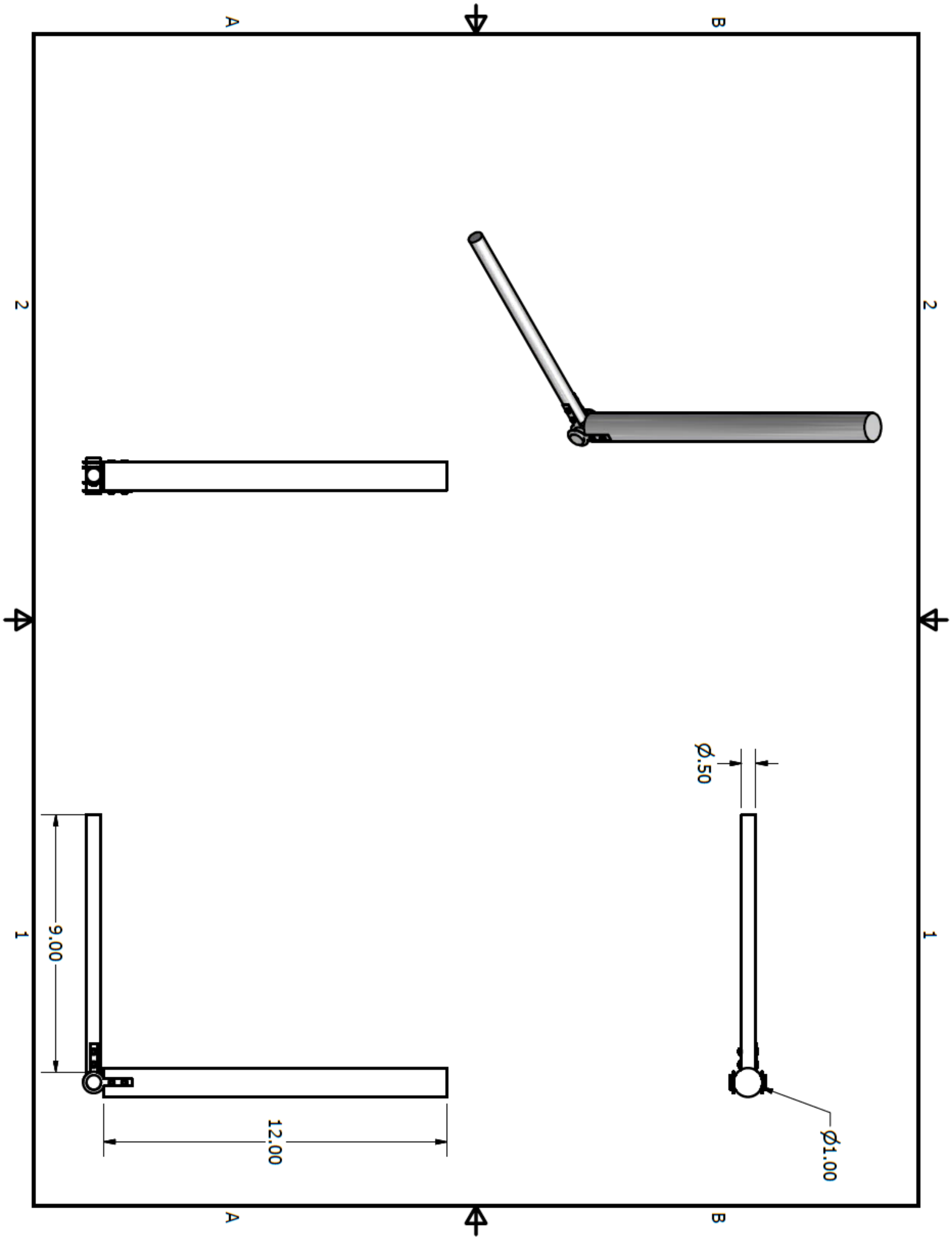

Figure A3: A final schematic for the myoelectric arm with moving elbow. This was produced using Autodesk Inventor software. Each piece was created in Inventor, and then built in an assembly file to create the final design.

Appendix 4: Elbow Hinge Schematic

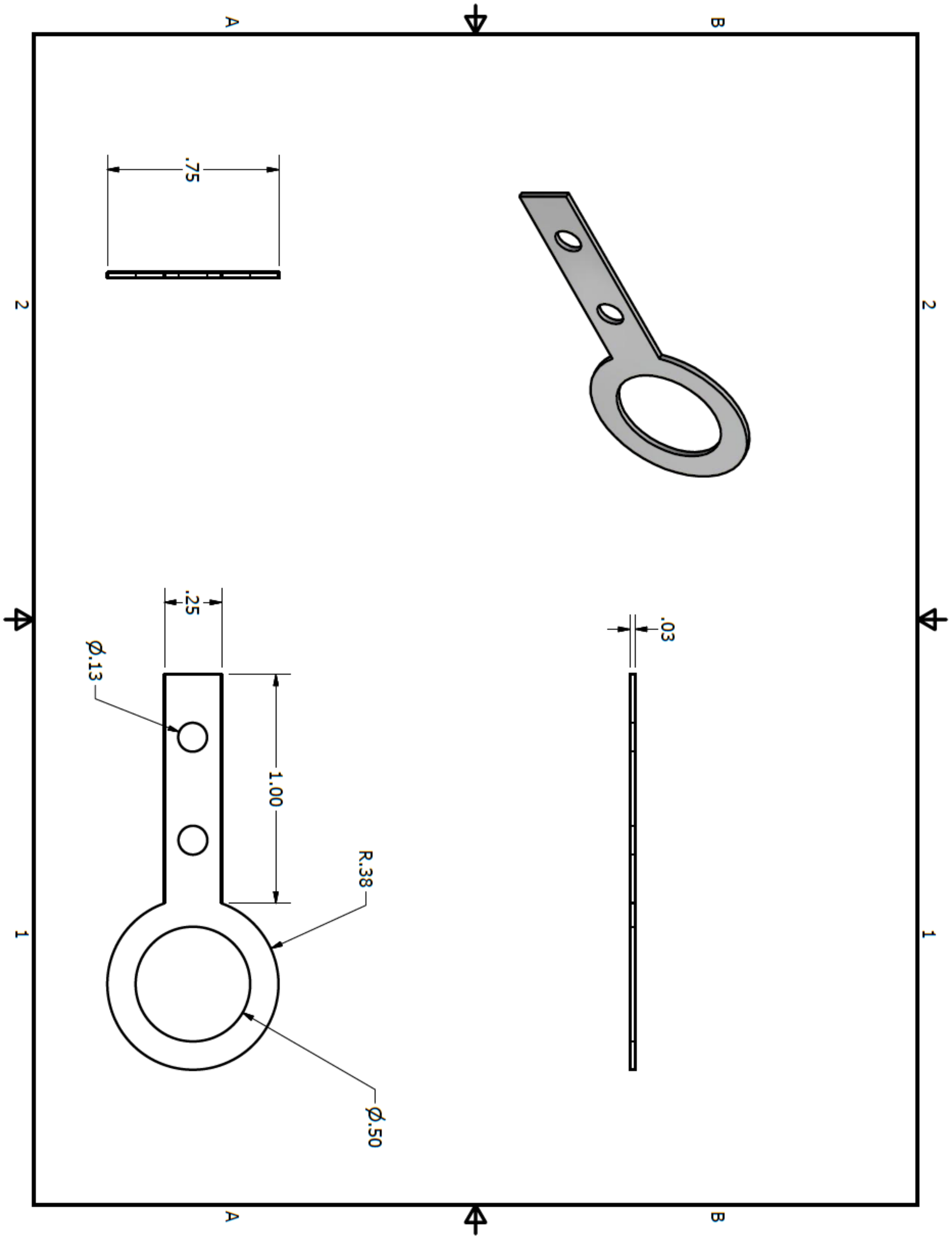

Figure A4: A final schematic for the elbow hinge design. This was produced using Autodesk Inventor software. It is cut out of a thin sheet of aluminum and attached to both sides of each aluminum rod.

Appendix 5: Initial Algorithm

```
stepper(400, 1)                              // sets up the stepper motor
minForce = X                                 // minimum force for contraction
maxForce = Y                                 // maximum force for relaxation

setup()
    stepper.setSpeed(500)                    // sets speed to 500 rpm
    set-up Muscle sensor
end setup

loop()
    if (sensor.force > minForce)             //when sensor force exceeds minForce
        stepper.step(100,FORWARD)            // steps forward 100 steps
        delay(1000)
        if (sensor.force < maxForce)         // if sensor force goes below maxForce
            stepper.step(100,BACKWARD)       // step backward 100 steps
            delay(1000)
    release()                                // release coils
end loop
```

Appendix 6: Arduino Code

```
// Arduino Arm Sketch
//
//  Sets up the Arduino Stepper Motor and calls
//  it to rotate forward 400 steps, hold, and then
//  move backwards 400 steps.
//
//  By Kate Talbot

#include <AccelStepper.h>               // includes the AccelStepper library
#include <AFMotor.h>                   // includes the Adafruit Motor library

AF_Stepper stepper(400, 1);

void setup()
{
  Serial.begin(9600);                  // set up Serial library at 9600 bps
  Serial.println("Stepper test!");     // prints to make sure stepper is ready

  stepper.setSpeed(50);                // sets the speed to 50 revolutions per minute
}                                      // end of setup

void loop()
{
 stepper.step(400, FORWARD, DOUBLE);          //moves the motor forward 400 steps
 delay(3000);

 stepper.step(400,BACKWARD, DOUBLE);          //moves the motor backwards 400 steps
 delay(3000);
 stepper.release();                           // releases motors coils
}
```