Prosthetic Design and Development

Myles Cooper, Nitya Dhanushkodi, Raagini Rameshwar, Meghan Tighe

May 2015

1 Introduction

During the 2014-15 academic year, we have run an Olin Self Study focused on Prosthetic Design and Development. Our goal has been to explore prosthetic hands and realize our learning in a robotic hand that is 3D-printable and controlled by electrical signals from the human body. This goal is inspired by and brings together components of two current prostheses: entirely mechanical models from e-NABLE [1], and more expensive electrically controlled prostheses from iLimb [2].

This document outlines our results and progress in three sections. The first is the electrical system, where we outline two different circuit trials and the results of each. The second is mechanical, where we discuss the many iterations of finger and hand design. The third is integration, where we describe how the mechanical and electrical systems connect together to make a robotic hand.

2 Goals

2.0.1 Intentionality of "Play"

Since this OSS falls somewhere between academic research and a learning experience, we acknowledged our fortunate position of having no pre-determined goals. In fact, each member of the team came into this project from different places and with different learning goals. Our common goals were to immerse ourselves in the world of prosthetic hands through the DIY mentality and to create something new. With these broad goals, our projected final product changed several times throughout the year.

Allowing our final deliverable to change organically helped us achieve our learning goals for a few reasons. Without required functions or performance metrics, we were able to be more creative, allowing us to move away from safer, established designs and toward novel ideas. Also, without the pressure of a specific product, we allowed ourselves to spend more time on certain areas, which let us really immerse ourselves in the areas we wanted to know more about, thereby tuning the knowledge and insights we gained to better fit our learning goals.

2.0.2 Low Cost

Some popular trends surfaced during our research of prosthetic hands. As a group we were inspired by the work being done by e-NABLE and similar organizations: creating 3D printed prostheses and making them available for free. 3D printing is a new technology, and as such has not been pushed to its limits yet. We wanted to explore how we could contribute to this body of research and development.

We have also noticed a widespread concern about the cost of myoelectric prostheses. One might wonder why myoelectric prostheses are worthwhile if they are so expensive. The ability to read intent via muscles allows myoelectrics to be run by battery, eliminating the increasing metabolic demand of passive prostheses. Also, by reading electrical muscle signals in the residual limb, they allow a much greater degree of control.

For these reasons, one central goal of this OSS is to find a cross between the readily available 3D-printed prostheses and the expensive, high-functioning myoelectric prostheses. Microcontrollers, 3D printing, and low-cost "hobby" parts are being used for this [3, 6, 7], but no group has found a breakthrough solution, so as a team we took a shot at our own design.

3 Process: Electrical

This section describes the electrical methods utilized to obtain, filter and amplify biological signals.

3.1 Electronystagmogram (ENG)

An electronystagmogram is used to detect involuntary eye movement by detecting impulses in the muscles surrounding the eye. Doctors use ENGs to detect eye movement in patients with vertigo and other balance-related issues. An ENG takes the muscle movement as inputs from electrodes placed around the eye, then amplifies and filters the signal.

3.1.1 Circuit Diagram

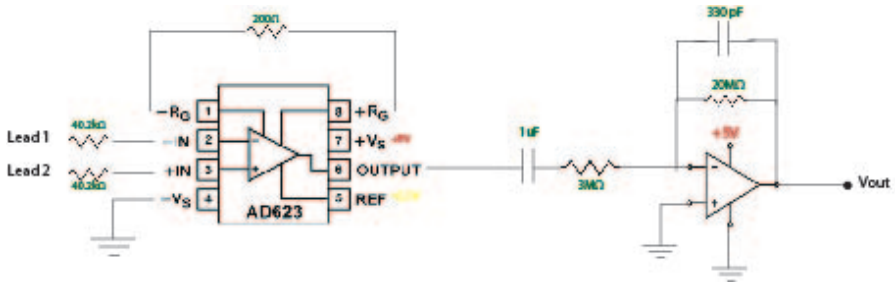

Figure 1: This ENG circuit takes in biological data, includes a simple filter and (theoretically) outputs readable data showing eye movements.

3.1.2 Circuit Components

The Instrumentation Amplifier consists of three op-amps and some precision resistors, to connect to the input electrodes. The amplifier gain is given by: $\frac{100k\Omega}{R_3}$, which is 200.

The Band-Pass Filter is a second-order filter that reduces noise by cutting out all signal frequencies except for those within its cutoff frequencies. The range of frequencies in this filter is $f_c = 0.0532Hz - 26.5Hz$. The specific component used is an AD623 from Texas Instruments.

3.1.3 Why we couldn't use it

Unfortunately, after extensive testing of this circuit, we were unable to get any clear or consistent muscle signals at all. The figure below shows one of our data samples. The test subject's muscle movements did not affect the signal, but another subject jumping around the room caused huge spikes.

3.2 Electromyograph (EMG)

3.2.1 Circuit Diagram

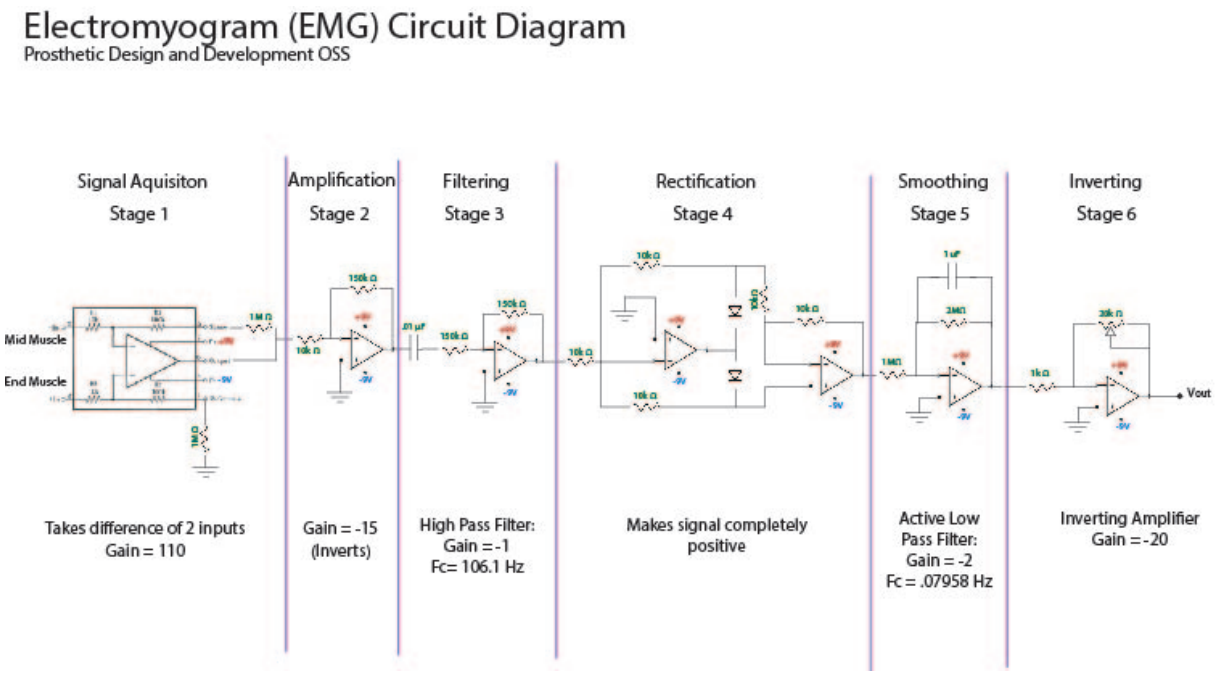

Figure 2: This is the EMG circuit [4], used to measure muscle movement using electric potential. The circuit takes in two inputs from electrodes on a muscle on the body and measures the difference between them. We recommend building the circuit in parts and testing for data at each stage. We provide graphs of the data for the stages for comparison. When building, compare the stages to the data provided and your intuition. That way, for example, you know when your output is really small, which stage your amplification is broken at.

3.2.2 Circuit Components

The voltage source is created from two 9V batteries by connecting them in series to create an 18V power source from -9V to +9V. The low end of battery 1 is at -9V, and ground is the high end of battery 1 and the low end of battery 2. The high end of battery 2 is now at 9V relative to ground!

The signal acquisition phase consists of the INA106, which measures the voltage difference between the two electrodes (labeled Mid Muscle and End Muscle). This difference amplifier chip then amplifies that voltage difference by a set gain of 10. Check that the output of this stage shows the difference between an input and an output with the output being an amplified version of the input with a gain of 10.

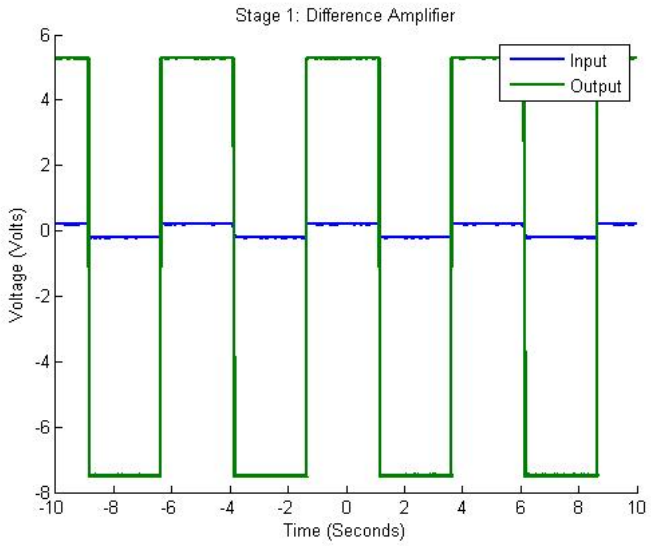

Figure 3: This graph shows the input and output of stage one of the EMG, the difference amplifier. The input is a small square wave with a voltage of 100mV and ground. The INA106 takes the difference of these two which in this case is just the square wave. The output is simply an amplified square wave.

The amplifying stage includes an inverting amplifier with a gain of -15. For this part we used the TL072 chip with a 150kΩ resistor and a 10kΩ. The gain is -15 because

$$G = -\frac{R_2}{R_1} = \frac{-150k\Omega}{10k\Omega} = -15. \tag{1}$$

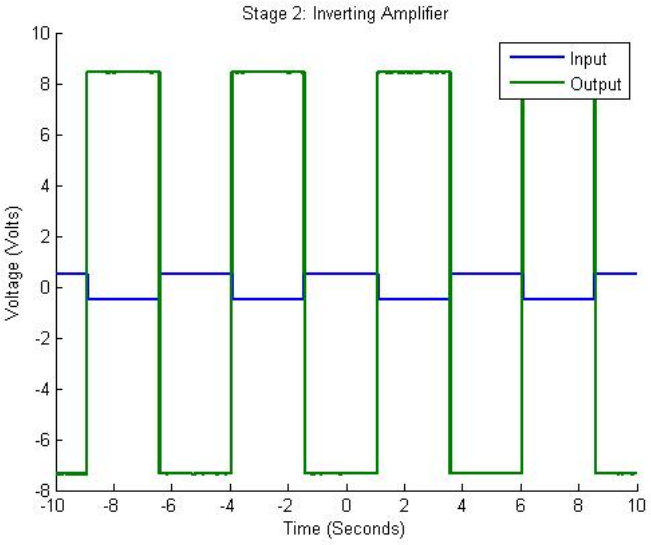

Figure 4: This graph shows the input and output of just stage two alone of the EMG. The input is a square wave with a voltage of 500mV. The output is an inverted and amplified square wave.

The filtering phase uses a high pass filter with a cutoff frequency of 106.1 Hz and a gain of -1.

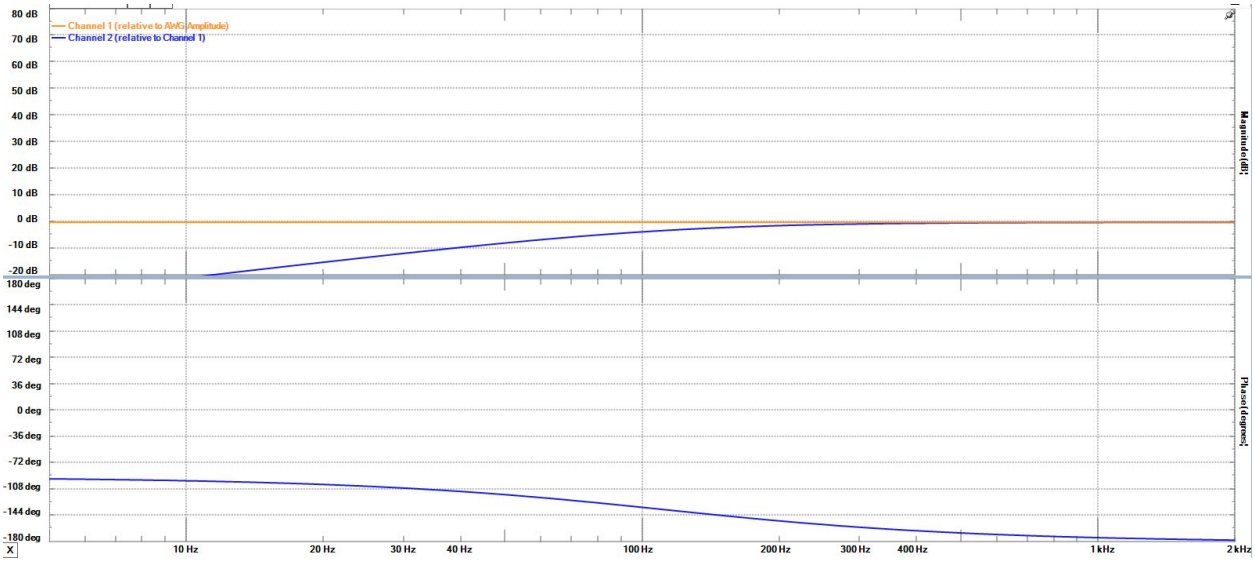

Figure 5: This graph shows bode plot of this high pass filter, confirming that the cutoff is 106.1 Hz.

The full wave rectifier flips all of the negative parts of the signal, turning them positive.

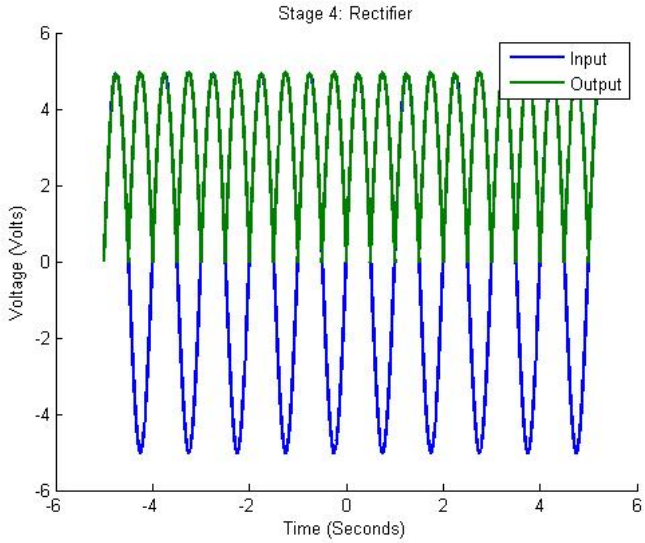

Figure 6: This graph shows the input and output of stage four of the EMG. The input was a 5 Volt sine wave. The output is a rectified version of the same wave: the negative portions are positive instead.

The smoothing phase includes a low pass filter which turns the AC signal into a DC voltage. Its main purpose is to smooth the signal and prevent low frequency noise. The filter has a cut off frequency of 0.08 Hz and a gain of -1. We couldn't show the Bode plot for this phase because the cutoff frequency is too low to show up on the Analog Discovery Bode Plot. We redesigned this from the original filter, which had a cutoff of 1.975 Hz. The first 80.6 kΩresistor was replaced with a 1MΩresistor, and the resistor in parallel with the capacitor, also originally 80.6kΩ, was replaced with a 2MΩresistor. This lowered the frequency cutoff so we could see a very clear spike in voltage when the person connected flexes, rather than high frequency voltage when this happens.

The inverting phase consists of an inverting amplifier with a gain of -20. With the use of a potentiometer, the resistance can be tuned depending on the strength and size of the signals being used.

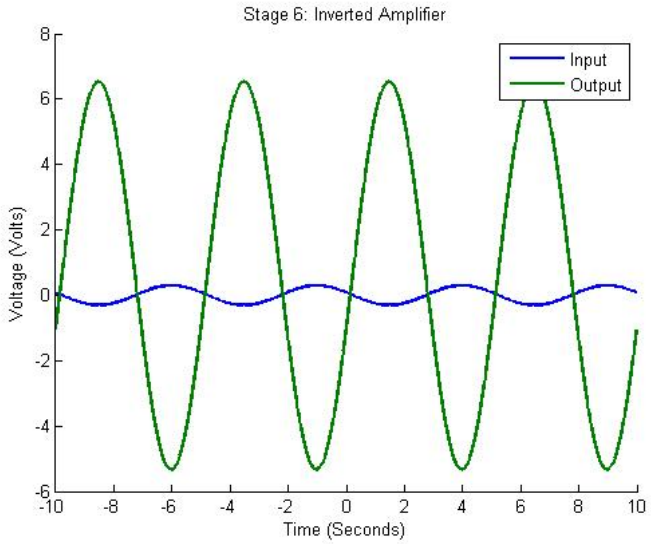

Figure 7: This graph shows the results of stage 6, an inverting amplifier with a gain of 20. Given a sin wave of 200mV, the amplifier outputs a much larger sine wave of opposite sign.

3.2.3 Results

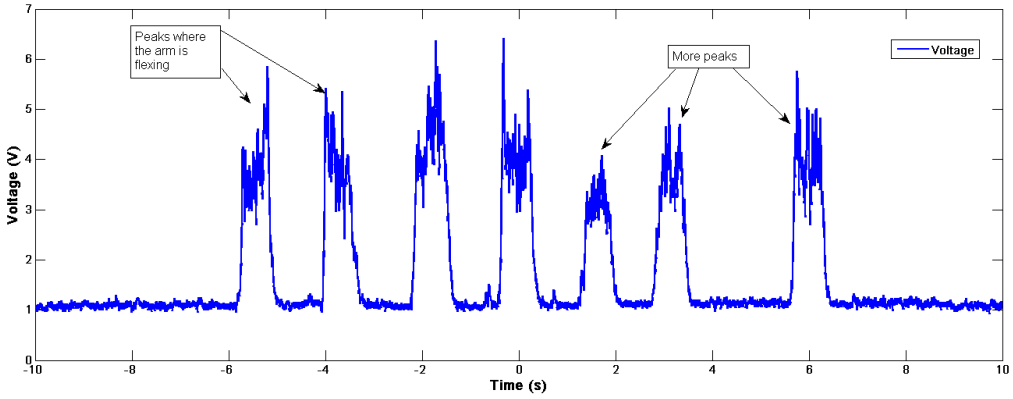

Figure 8: This graph shows results of the EMG circuit when measuring signals from the forearm.

3.3 Future Exploration

Firstly, the signal we acquire from the muscles could be smoother. Ideally, we would like to see very clean and parabolic peaks rather than the fuzzy ones we are reading now. Though the peaks work with our current system, cleaner signals mean a more accurate correlation between the mechanical hand and the muscles controlling it.

Secondly, the entire system can be a bit neater. Right now, there are a lot of wires and connections to make every time we want to turn the hand on. Ideally, the circuit would be a PCB with soldered wires that can be clipped to electrodes. We might also use an Arduino Mini rather than the Arduino Uno, since it would be easier to house a Mini in the hand itself.

Thirdly, we would like to give the user feedback from the hand, so they know when they might be gripping something too hard. Right now, we only keep track of the motor position and if we were able to turn the motor more than 180 degrees, the fingers could break without feedback. We would like to avoid this by putting force sensors in the hand and using them to stop the fingers when necessary.

4 Process: Mechanical

This section describes how to make the current hand prototype, in the process detailing the important considerations we stumbled upon.

4.1 Biomimetic Finger

The mechanical side of this project focused on creating a biomimetic finger with potential use in prosthetics. In order to understand some of this design and its terminology, one must understand the basics of human finger anatomy.

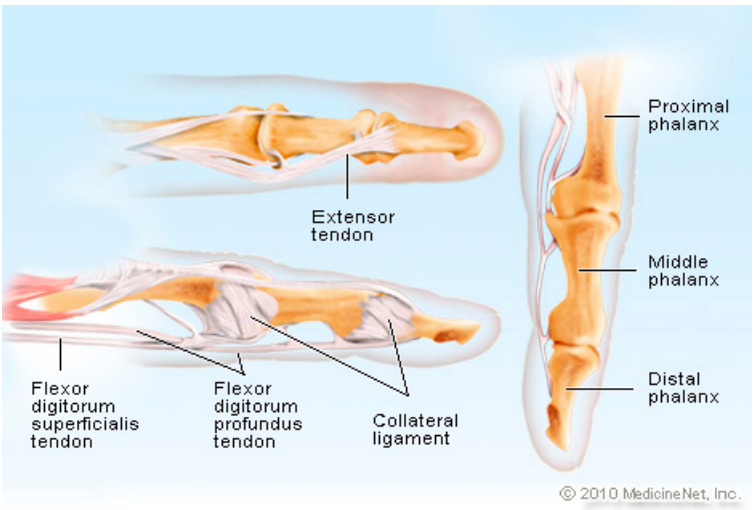

Figure 9: Three views of the finger, showing the major bones (phalanxes) and tendons (which actuate the finger).

The muscles that pull the tendons that contract and extend each finger are located in the forearm, which is why our robot hand receives myoelectric signal from the forearm. Some directional terminology to know:

proximal - closer to the body
distal - further from the body
dorsal - referring to the back of the hand
palmar - referring to the front of the hand

4.1.1 Why Biomimetic?

Human fingers are a wonderful example of under-actuation – the ability to control multiple degrees of freedom with a single actuator. We wanted to emulate the human body's natural forms of under-actuation, but in a simplified case. This meant using one "tendon" running through the palmar side of each finger, representing the *flexor digitorum* tendons of that finger.

Of course, our finger design diverges from biomimicry in several more ways. For one, the finger we made uses hinge joints with polyurethane torsional bushings acting as the pin. This acts a torsional spring to return the finger to its un-flexed state, thereby removing the need for an extensor tendon. The design is, however, roughly accurate in dimensions to a human index finger and includes both a rigid "bone" structure along with a flexible "skin" to enhance grip.

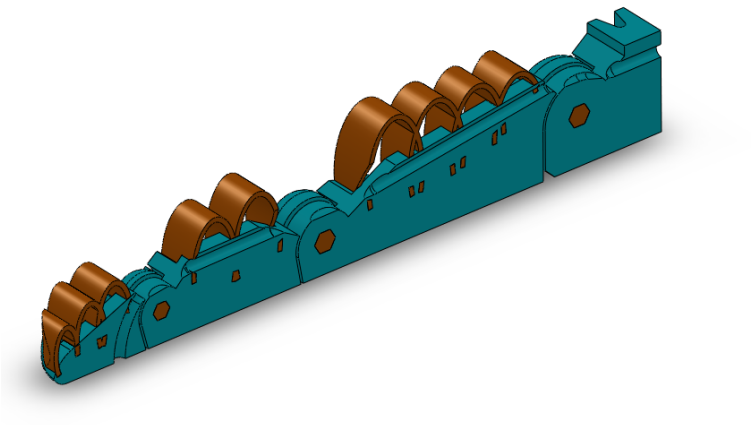

Figure 10: A cross-section iso render of the finger design, showing Ninjaflex (polyurethane) skin and torsional bushings in orange and ABS phalanges in teal.

4.1.2 Why 3D printed?

We wanted to use 3D printing as our sole mechanical fabrication method primarily because of its combination of accessibility, geometric freedom, and low cost (for CNC). We used a Makerbot Replicator 2 and a Replicator 2X for all of the 3D printing. Though these printers (especially duel extrusion printers like the 2X) are not currently common, trends in 3D printing imply that they soon will be, and we think that the added functionality that duel extrusion offers warranted the reliance on the Replicator 2X.

4.2 Printing With Ninjaflex

Printing with Ninjaflex and ABS is a conflicting endeavor because ABS usually needs a build plate heated to ∼110°C to avoid warping, while Ninjaflex only needs 50°C to stick and any hotter makes it stick too hard to the build plate. Throughout most of the OSS, we dealt with the excessive adhesion and used a Kapton tape covered build plate at 110°C. Near the end, we transitioned from Kapton tape to blue painter's tape. Though this would theoretically let us reduce build plate temperature, we kept it at 110°C to ensure that the prints would not warp. The harm of this is that taking parts off the build plate often mars the surface of the tape, but we deemed that better than ripped Kapton tape or warped parts.

One quirk of this build plate temperature dilemma is that you should be aware of whether ABS or Ninjaflex is touching the build plate (Ninjaflex will stick much harder, and will fuse to Kapton tape). To solve this, we added an ABS raft to the print settings, but often rafts are less than ideal. If only Ninjaflex is touching the build plate, we recommend lowering build plate temperature to 50°C. If both Ninjaflex and ABS are touching the build plate, we recommend keeping the build plate temperature at 110°C.

4.3 Current Mechanical Design

4.3.1 Pulley System

The pulley system, adapted from a 2008 paper [5], allows the fingers to be driven by fewer motors while keeping a compliant grip (the fingers are allowed to close different amounts around an object). Though it could theoretically be driven by one servo, we used two to allow more force to be transmitted to each finger.

4.3.2 Easy Construction

To make construction simple, the only hardware in the hand is the wire to transmit force and the mounting screws for the servo (that are included with the servo). The rest of the pieces slide or clip into place.

4.3.3 How To Put The Hand Together

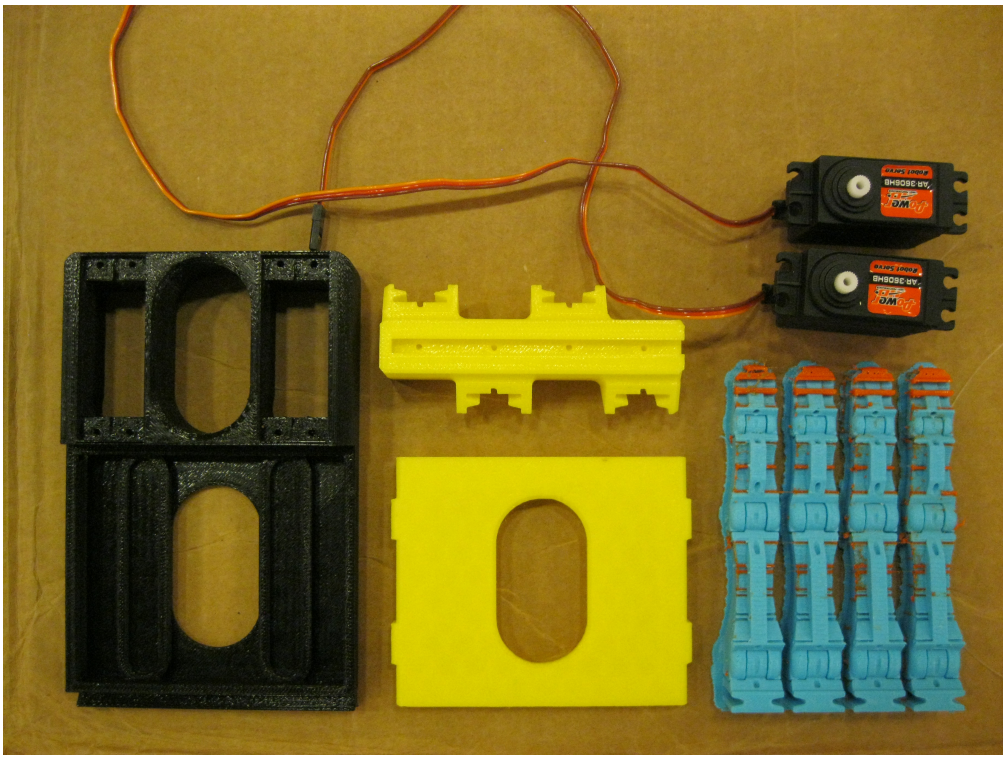

Figure 11: An overview of the major components needed to assemble the hand.

Assembling the hand is a fairly simple process, involving only three print jobs. One print job has the palm (pictured in black in figure 11).

Cleaning the Fingers

Perhaps the most tedious step in assembling the hand is cutting the raft off the four teal & orange fingers. This step is only necessary because of the difficulty printing Ninjaflex and dealing with ABS's tendency to warp.

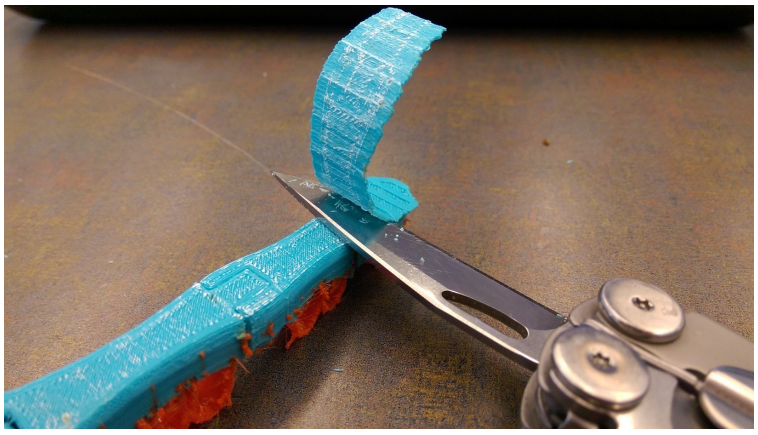

Figure 12: Removal of the ABS raft from the dorsal side of the fingers.

It is easiest to remove this raft by inserting a blade between the raft and the middle of one phalange,

then sliding the blade down the length of finger. The joints can be difficult to separate, but the rest of the raft should slice off easily.

The joints will most likely be partially fused together at this point. Break each joint loose by holding the phalanges close to the joint and twisting the hinge. This will make a sharp click as each joint breaks loose. To loosen the knuckle joint, twist the hinge backward slightly, then forward to fully break it loose.

Sliding the Pieces Together

The Fingers, knuckles, and palm all slide together, making the bulk of the construction very easy. Simply slide the dovetails at the proximal end of the finger into the corresponding slots in the knuckle and slide the dovetail on the knuckle into the dovetail on the end of the palm opposite the servo mounts.

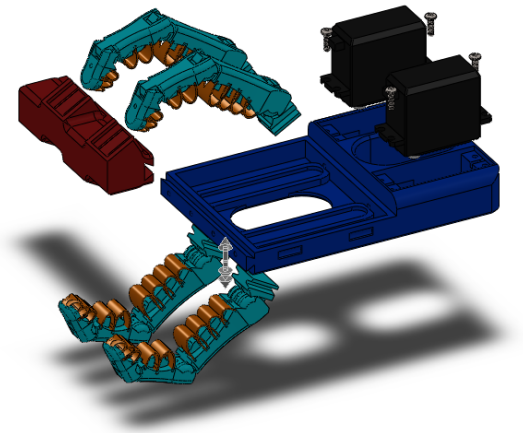

Figure 13: The orientation of the knuckle (red), palm (blue), and fingers (teal & orange) for sliding assembly.

Mounting the Servos

Mounting the servos should be as simple as sliding the servos into their cutouts with servo horn facing down and the wires extending out toward the fingers. Screw the hardware included with the servo into the mounting holes in the palm, fastening the servo down.

Stringing in the Pulleys

The pulleys sit nicely in the slots and need only be connected by string to each finger and each servo. Thread the string through the channels on the palmar side of one end finger. This will bring it through the holes in the palm to where the pulleys sit. Loop the string around the appropriate pulley and thread it back down through the adjacent finger. At the distal end of each finger (where the string comes out of the dorsal side), tie the string off to form a bulge that prevents the string from sliding through the channels during contraction. In this step, be careful to make the string taught when both ends are tied.

At this point, the other side of the pulley needs to be strung up and attached to the servo. On the pulley end, tie a half hitch (or other adjustable knot) and tighten it around the unused side of the pulley. On the servo end, use the circular servo horn and loop the string through the provided holes, tying it off and making it taught as done before.

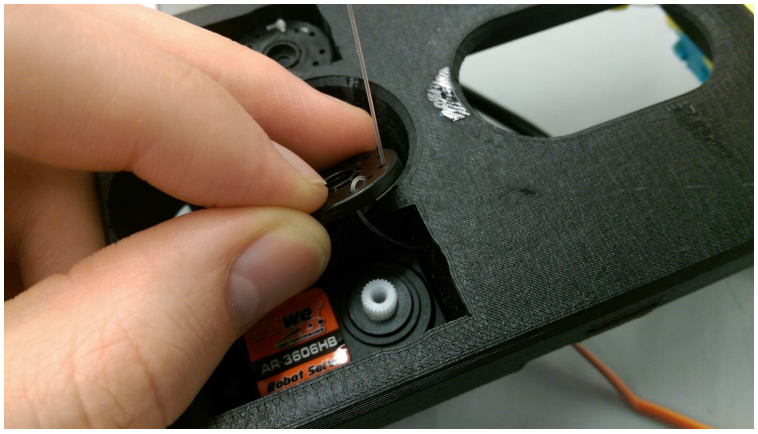

Figure 14: Tying the string to the servo horn, effectively making a pulley.

With everything tied up, simply clip on the palm cover and plug in the circuitry!

4.4 Future Exploration

First and foremost on the mechanical side, the printer settings for Ninjaflex and ABS need to be refined. Though the settings we figured out brought us through the semester, being able to print the fingers more cleanly, reliable, and without a raft would be invaluable for future development.

Since most of the mechanical design of this OSS went into the finger, future iterations of this project would benefit from revisiting the hardware-less construction of the hand. 3D printing clips and sliding fits is difficult, but worthwhile to achieve accessibility.

One major improvement on the front of simplifying the mechanical design would be to find a way to completely eliminate hardware from the servo mounts. Screws are small and easily lost, so perhaps having a housing that the servos could simply slide into would be an easier-to-manage option in the end.

To reduce weight and complexity, this design should be driven by one motor. The current design is limited by torque requirements, so moving to one motor could be tackled from two directions: reducing friction to reduce torque requirements and to find a more power-dense motor, allowing for more torque output without increasing weight or size significantly.

For ease of design, most of the current hand is a blocky design. This is not optimal for a wearable device, so future work on this hand should try to adapt the shape to better fit around a forearm. The designers should also revisit the layout of the fingers to optimize it for grasping different shapes.

5 Integration

An Arduino is the interface between the electrical and mechanical components of this system. The output of the circuit goes into an analog input, and the Arduino reads, slightly filters, and interprets this input. The servo is run by an output pin using the servo library in Arduino. The Arduino checks for values from the muscle signal above a certain threshold, and turns the servo to that position of the signal. So holding a clenched muscle holds the hand closed, and releasing your muscle opens the hand. The code is provided here:

```
#include <Servo.h>

//Create two servo objects, one for each servo controlling the hand
Servo myservo;
Servo myservoinv;

int pos = 0;    // variable to store the servo position
```

```
int sensorPin = 0;
int lastRead = 0;
int sensorValue = 0;

void setup()
{
  Serial.begin(9600);

  //One servo is on pin 8, the other is on pin 9
  myservo.attach(8);  // attaches the servo on pin 9 to the servo object
  myservoinv.attach(9);

  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);

  //We read from analog input A1
  pinMode(A1, INPUT);
}


void loop()
{

  //Read from the analog input every 100 ms. This gives us the value
  //output by the circuit based on the user's muscles
  if (millis()-lastRead > 100){
    sensorValue = analogRead(A1);
    Serial.println(sensorValue);
    lastRead = millis();
    }

   //We map this value to a 0-160 range so we know how many degrees we want.
    double y = map(sensorValue, 0, 1023, 0, 160);


   //if this value is above a certain threshold, we write it to the servos.
   //One of the servos requires an inverse signal, so it's 180-y. The +20
   //strengthens the signal a bit.
    if (y > 10){
      y = y+20;
      myservo.write(y);
      myservoinv.write(180-y+20);
    }
}
```

5.1 Future Exploration

The biggest area of opportunity with integration is in the motor used. Our system is currently driven by two position controlled servos. Using speed controlled servos would allow us more freedom and a greater degree of movement. We chose not to use speed controlled servos in this iteration due to complications with Arduino code, which is why this step is a further step in the area of integration, not the mechanical system.

References

[1] e-NABLE Organization, *http://enablingthefuture.org/*

[2] Touch Bionics, *http://www.touchbionics.com/products/i-limb-whole-hand-solutions*

[3] Bionico Hand, *http://bionico.org/*

[4] Instructables, "Muscle EMG Sensor for a Microcontroller" *http://www.instructables.com/id/Muscle-EMG-Sensor-for-a-Microcontroller/*

[5] C. Gosselin, F. Pelletier, and T. Laliberte. "An anthropomorphic underactuated robotic hand with 15 dofs and a single actuator," *IEEE International Conference on Robotics and Automation, 2008, ICRO 2008* 2008, pp. 749-754

[6] J. Simon, "Jose Delgado, Jr. Compares His $ 50 3D-Printed Hand to His $ 42,000 Myoelectric Prosthesis," *3D Universe,* 19-Apr-2014.

[7] Mizchief100, "DIY Prosthetic Hand & Forearm (Voice Controlled)," *Instructables,* http://www.instructables.com/id/Voice-Controlled-Prosthetic-Hand-Forearm/