Design of an Electromyography Controlled Transradial Prosthetic

Michael Botros and Matthew Choi

***Abstract* - In order to improve the quality of life of amputees, an electromyography-controlled transradial prosthetic was designed. The system was designed by first creating the main electronics circuit. This circuit consisted of three surface electrodes, a MyoWare electromyography sensor, an Elegoo UNO microcontroller, a servo motor, and electrical wires. Open Bionics, an open-source mechanical design, was utilized in order to reduce design time and maximize functionality. The mechanical design was fabricated out of 3D printed plastic, fasteners, fishing line, and silicone sheet. An exponential smoothing algorithm and speed-proportional control system were implemented into the microcontroller code. Calibration required the serial input of the patient's maximum voluntary isometric contraction into the code. Due to the faulty sensor, the system worked only half of the time. The sensor would give faulty data, which was unable to actuate the mechanical system. However, this project demonstrated the feasibility of speed-proportional control in a real-time prosthetic system.**

Keywords - electromyography, prosthetic hand, speed proportional control

I. INTRODUCTION

The need for prosthetic limb replacements has been growing; data from the United States in 2005 show that 41000 people were living with major upper limb absence which equates to 1 in 10000 people [1]. In this project, a speed-proportional electromyography (EMG) controlled transradial prosthetic was developed. The prosthetic was meant to target amputees, with a low to moderate activity level.

The design of a transradial prosthetic actuated by an EMG system is highly desirable. It allows for a rapid real-time response from the patient, and high controllability through speed or force proportionality. Also, the use of the microcontroller provides options in customizability and adaptation since the code can be reconfigured over time.

II. BACKGROUND

A. Physiology of EMG

EMG is the discipline that deals with the detection, analysis and use of electrical signal that comes from contracting muscles. One muscle has many motor units; Figure 1 shows the structure of one of these motor units. [2]

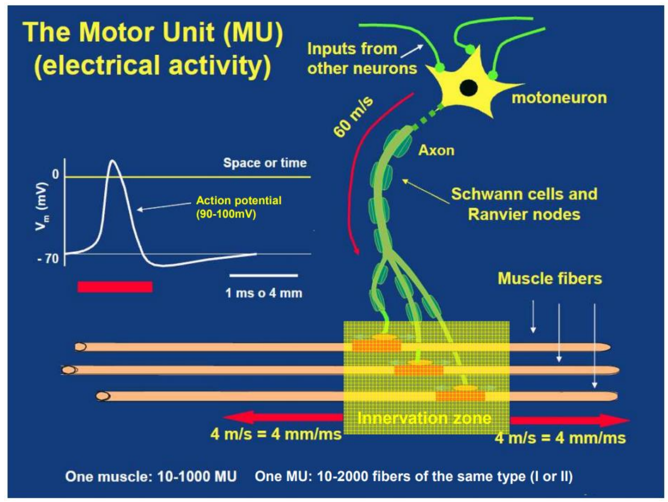

Figure 1. Structure and Electrical Activity of a Motor Unit

Each motor unit consists of the motor neuron and muscle fibers [2]. An action potential is a sudden change in electrical polarity due to an above-threshold stimulus (a stimulus that brings the potential in the neuron to above -55 mV (up to +40 mV) from a resting potential of -70 mV). [3]

Upon the command for motion, action potentials travel down the axons of the motor neurons from the motor cortex in the brain to the muscle fibers of the muscle responsible for this motion. At the innervation zone, muscle fibers move from one side to another through the sliding filament theory which in basic terms, is the swinging of myosin along actin. This movement of muscle fibers causes the muscle to contract. [2]

EMG is the summation of all action potentials happening in a muscle at one time; however, these action potentials do not fire synchronously. In weaker contractions, fewer muscle fibers are recruited and therefore, the EMG signal is relatively small. However, for larger contractions, spatial summation is used to recruit more muscle fibers to produce a larger EMG signal. [4]

B. Three Common Types of Prosthetic Hands

In a body-powered prosthetic hand, prosthetic actuation is synergistic with the actuation of an intact body part [5]. An example of this kind of prosthetic is a device that operates by extending the living arm to pull on cables and open a claw hand [6].

In a flex-sensor based prosthetic hand, the flexion of living fingers are sensed to queue the synergistic actuation of the prosthetic fingers at the same time [5].

Furthermore, an EMG sensor-based prosthetic hand has actuation that is signaled by muscle contraction [5]. This type of prosthetic hand is the application of this project.

III. METHODS

A. Design Steps

These steps were taken in the design process of the EMG-controlled transradial prosthetic (in comparison to the final presentation, steps 5 and 6 were switched for a better representation of the design process):

1. Identify the interfacing between biological, mechanical and electrical systems.
2. Identify functional and non-functional requirements.
3. Choose mechanical and electrical components to satisfy the requirements.
4. Prototype the mechanical and electrical systems.
5. Calibrate and verify the electrical system.
6. Test and verify mechanical system functionality.
7. Implement the full algorithm in the Arduino code and verify both systems combined.

B. Interfacing of Biological, Mechanical and Electrical Systems

The device is semi-automated with the following steps for interfacing as illustrated in Figure 2:

1. A command is sent from the user to close the myoelectric prosthetic hand.
2. An EMG sensor detects the myoelectric activations of the forearm muscles.
3. Signals from the muscle sensor go to a microcontroller.
4. The microcontroller commands the servo motor to actuate the hand, and outputs the EMG signal on a computer screen.

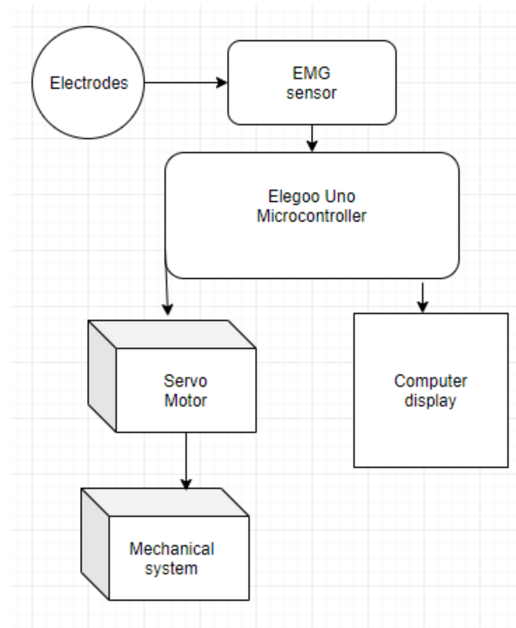

Figure 2. High-level Block Diagram Showing the Interfacing between Systems in the Device

C. Functional and Non-functional Requirements

To ensure the design of the prosthetic system maintained specificity, functional and non-functional requirements were determined. The functional requirements that were achieved included: low time delay, speed proportionality, lightweight and mechanically strong, filtered signal, and one degree of freedom. Force proportionality and multiple wrist degrees of freedom were not implemented, due to time constraints.

As for non-functional requirements, only a few were completed in order to prioritize the functional requirements. Optimized output signal, minimal wiring and device size, and lack of interference with the residual limb were completed. The protective casing, impact resistance, comfort, and patient parameterization were not implemented again due to time constraints.

D. Technical Specifications of Device Components

In regards to the electrical systems, Table 1 lists the electronics that were chosen and provides technical specifications to justify why they were selected.

Table 1. Technical Specifications of Electronics

Component	**Technical Specifications**
Myoware Muscle Sensor (AT-04-001)	● Gain adjustable from 0.00201 to 20100 ● Common-mode rejection ratio (CMRR) of 110 dB
3M Red Dot Surface Electrodes	● Sticky gel ● Sensor made of Ag/Ag-Cl coated plastic
Fitec FS5106B Servo Motor	● 6V DC ● 160 - 190 mA ● Maximum torque = 0.5894 N-m ● Rotation speed of up to 1 rotation per second
Elegoo UNO R3 Microcontroller	● 100% Arduino-compatible

For surface EMG signals, a gain of 1000 and a CMRR of at least 80 dB are appropriate [7, 8]. The Myoware Muscle Sensor has an adjustable gain potentiometer; this gain (G) is obtained with eq. 1,

$$G = \frac{201\, R_{gain}}{1\ k\Omega} \qquad \text{(eq. 1)}$$

where R_{gain} is the potentiometer resistance in kΩ [8, 9]. For the project, the default resistance of 50 kΩ was used to provide a gain of 10 500 which is a more than adequate amplification of the approximately 40 mV EMG signal [9]. For the differential amplifier imbedded in the sensor, the CMRR is the ratio of the differential gain over the common mode gain and this ratio quantifies that ability of the sensor to reject signals that appear simultaneously and in-phase on both inputs of the amplifier [10]. It is important to reject these signals because common mode components will cause an error in the measurement of signals [10].

As for the electrodes, the 3M Red Dot Surface Electrodes were selected because they are non-invasive and they have sticky gel which reduces the impedance between the skin and electrode surface [11].

The Fitec FS5106B servo motor is considered to be a high-torque, lightweight servo motor, so it was selected as the actuator to provide adequate grasping [12].

Using the Elegoo Uno microcontroller, any Arduino code can be executed, making it easy to reconfigure the algorithm of operation. The algorithm of operation has two parts: exponential filtering (explained in section F) and speed proportional control (explained in section H).

E. Prototyping

As illustrated in Figure 3, the primary output for the selected EMG sensor is an amplified, rectified and integrated signal (in other words, the EMG's envelope).

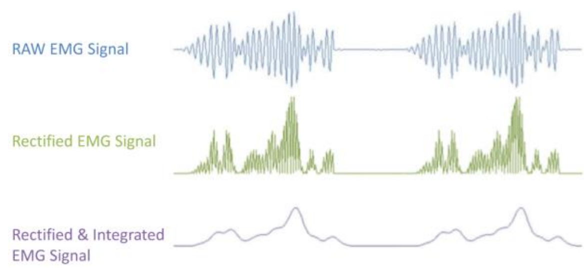

Figure 3. EMG Envelope (Purple Curve) Produced by MyoWare Muscle Sensor

To obtain the primary output, wires were soldered onto the power supply, ground and output signal ports of the muscle sensor as shown in Figure 4.

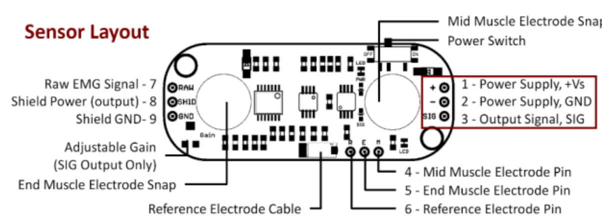

Figure 4. MyoWare Muscle Sensor Layout

The electrical circuit consisted of the microcontroller, EMG sensor and servo motor; the wiring for this circuit is displayed in Figure 5.

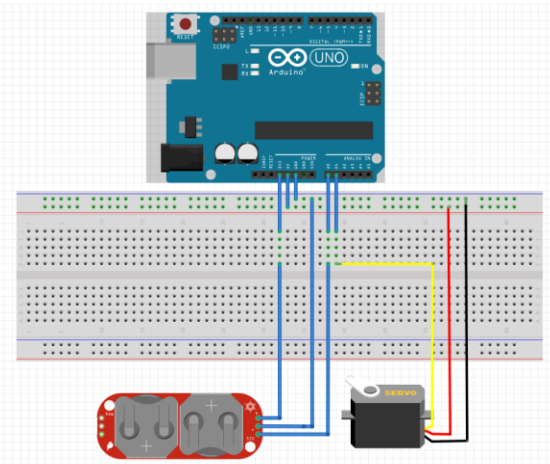

Figure 5. Wiring of the Electrical Circuit

As for the mechanical system, an open-source design coming from Open Bionics was replicated. Most of the structural parts for this design were 3d-printed with a 0.4 mm-nozzle Ultimaker 2+ with polylactic acid (PLA) plastic and an open filament system [13]. The following settings on the 3d-printer were selected:

- 0.8 mm wall thickness
- 0.2 mm layer height
- 0.8 mm bottom and top thicknesses
- 40% infill
- 60 mm/s printing speed

The fingers of this design are bioinspired; nylon monofilament fishing line with a tensile strength of 10 lb was used as tendons and were routed through the 3d-printed phalanges of the fingers. Furthermore, this fishing was used to sew the phalanges to rectangular strips of silicone rubber sheet with a hardness of 50 A (slightly less bendable than a rubber tire). These silicone rubber strips were placed in between the phalanges in order to provide restitution after bending the fingers.

The final prototype combining the mechanical and electrical systems is depicted in Figure 6 with the red MyoWare Muscle Sensor and surface electrodes shown on the left, the Elegoo Uno microcontroller shown on the bottom and the built Open Bionics hand shown in the middle. This hand is actuated by a servo motor at the base of Open Bionics hand (not shown in Figure 6). More details about how this open-source design works will be discussed in the mechanical testing section of the report (section G).

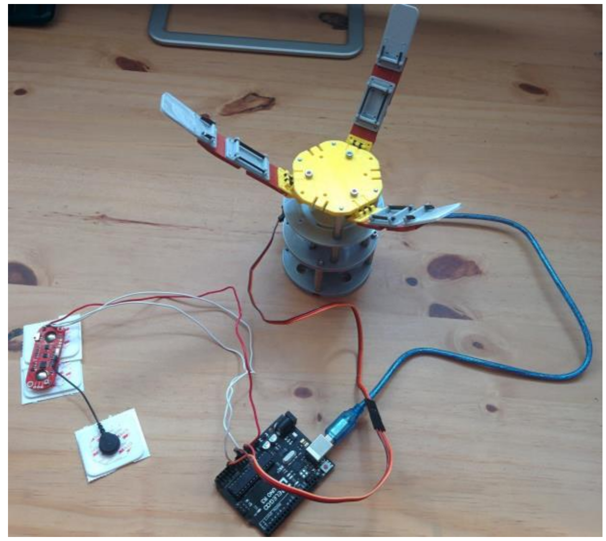

Figure 6. Final Prototype of the EMG-Controlled Transradial Prosthetic

The Open Bionics hand has the following dimensions:

- Base height of about 150 mm
- Base radius of about 80 mm
- Finger length of about 120 mm

F. Calibration and Verification of Electrical System

Similar to the setup shown in Figure 7, when operating the device, two surface electrodes are placed on the belly of the flexor digitorum superficialis, the main muscle responsible for clenching the fist and for gripping, and one surface electrode was placed near the elbow as a ground electrode.

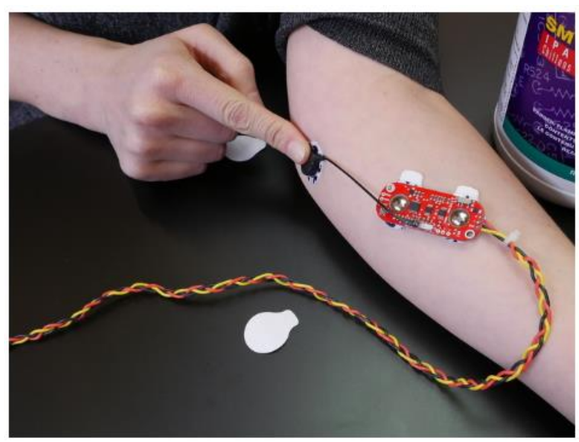

Figure 7. Electrode Placement (Two on the EMG Sensor and One Ground Electrode) [14]

Before calibration, noise needed to be filtered out of the EMG signal. As explained in the prototyping section of the report (section E), the MyoWare Muscle Sensor provided the processed EMG envelope. Further filtering of this EMG envelope in real-time was carried out with an exponential smoothing algorithm described by eq. 2,

$$y_n = wx_n + (1 - w)y_n \qquad \text{(eq. 2)}$$

where y_n is the smoothened value, w is the weight constant, and x_n is the new data point [15].

This exponential smoothing algorithm is a recursive type algorithm, which was called each time the programs main loop was iterated. The weight constant w was taken to be 0.5, since this value of w minimized the mean squared error [15].

Calibration of the EMG sensor is important because muscle morphology varies from person to person [16]. For the project, normalization was used as the method of calibrating the EMG sensor. Two advantages of normalization are that this technique helps reduce variability caused by EMG factors such as electrode placement, orientation and skin impedance, and it provides values that can be better compared across people and across days [17].

For our project, the maximal voluntary isometric contraction (MVIC) method was used since it is the most common method of normalizing EMG signals. An isometric contraction is one during which the length of the muscle stays constant; in this case, the MVIC of interest is that of the flexor digitorum superficialis. After the user was hooked up to the EMG system (electrodes and a computer to display the EMG-signal), the user squeezed a hand gripper as hard as possible for two seconds. There were a total of three repetitions of this squeezing with two minutes in between each repetition (to avoid the effects of muscle fatigue). To ensure repeatability, three Arduino serial output values from the three repetitions were obtained for the same subject under the same conditions. The MVIC value was chosen to be the maximum value out of the three EMG waveforms displayed on the computer. Next, the normalized EMG signal was found by dividing the real-time smoothened serial output by the MVIC value (which is kept constant in the Arduino code). [18]

G. Test and Verification of Mechanical System

Figure 8 indicates the major components in mechanical design of the Open Bionics hand. The three tendons of the bioinspired fingers are tied to the top of the differential disk and the pulley string is tied to the bottom of the disk. When the servo motor rotates clockwise, the pulley attached to the servo motor also rotates clockwise to pull downward on the differential disk. When the disk moves downward, the tendons of the fingers are pulled to close the hand. When the servo motor returns to its initial position, the silicone rubber between the phalanges elastically restores the fingers to open the hand. [5]

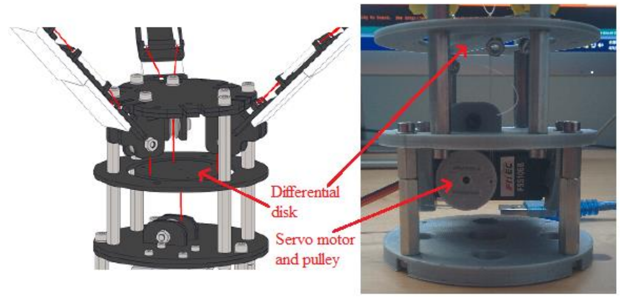

Figure 8. Open Bionics Mechanism Consisting of a Differential Disk, Servo Motor and Pulley [5]

A servo motor angle of 180 degrees was set for an open hand. Next, Arduino code was written to increment the servo motor angle slowly until the hand was fully closed. The closed-hand servo motor angle was determined to be 135 degrees.

H. Implementation of an Algorithm with Speed Proportional Control

In the algorithm to operate the device, the EMG sensor senses the electrical activity in the muscle of interest and outputs serial values corresponding to the voltage of the muscle contraction. If the EMG sensor outputs a real-time value above the threshold value (set to 50), the microcontroller proceeds with normalization (followed by actuation). This threshold value was determined qualitatively; it corresponds to the serial output of a comfortable contraction. Next, the servo driver in the microcontroller determines the normalized value by dividing the real-time sensor serial output by the MVIC serial output (previously determined during calibration). If the normalized value was greater than 1, then it was reassigned to 1. Next, the mapped value was determined by mapping the normalized value ($0 < x < 1$) to the speed boundaries ($1 < y < 200$). This mapping was done linearly in order to obtain a value between 1 and 200 for the servo motor rotation speed that was directly proportional to the normalized values. So long as the servo motor's angular position is in the range of 135 degrees to 180 degrees, the servo motor rotates from 180 degrees (open hand) to 135 degrees (closed hand). When the hand is closed, the servo motor stops moving. Upon the release of the contraction, the servo motor is commanded to go back to 180 degrees for the hand to open.

Please refer to Appendix A for the Arduino code used to implement the previously described algorithm. The next section of the report describes the results of the verification of the electrical and mechanical systems combined.

IV. RESULTS

A. Encountered Difficulties with the EMG Sensor

After the implementation of the Arduino code, the prototype was tested. However, after three weeks of sensor problems, testing was largely inaccurate. In order to determine the faulty component of our system, the code and electrical system were tested. As a result of isolation testing, the sensor was determined to be very temperamental. Due to the nature of the sensor, testing and verifying the system was difficult.

The sensor would often give strange non-zero values, oscillating within 3 serial values. However, sometimes it would suddenly begin to work, and the prosthetic would actuate in accordance with what one would expect from the algorithm.

B. Results of Calibration with Normalization

Aside from these difficulties, a few successful tests of the code were completed. As previously described, the calibration was done using the MVIC normalization technique. On the computer displaying the EMG signal, the serial plots for the three independent maximum-strength grips were recorded and analyzed. Figures 9, 10 and 11 show these three serial plots (plots of serial output vs. time). Figure 11 was observed to have the highest serial value of 151, meaning that the system could be calibrated around this number.

Figure 9. Serial Plot for MVIC Normalization - Repetition 1

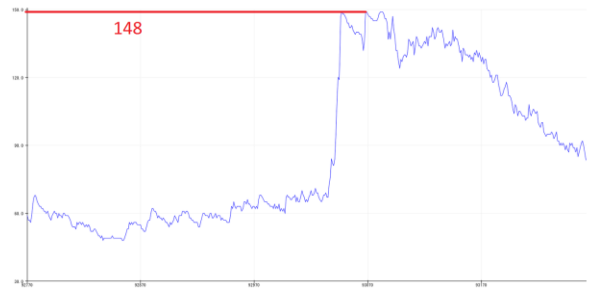

Figure 10. Serial Plot for MVIC Normalization - Repetition 2

Figure 11. Serial Plot for MVIC Normalization - Repetition 3

C. Verification and Validation of Prototype (Electrical and Mechanical Systems)

The MVIC value was taken from Figure 11, and used to calculate the normalized values in real-time. Finally, the code was tested to see if the mechanical hand would actuate after stimulus from the forearm muscle of the subject. The test was successful as seen in Figure 12 where the hand is fully closed the EMG waveform is present upon the squeezing of the hand gripper.

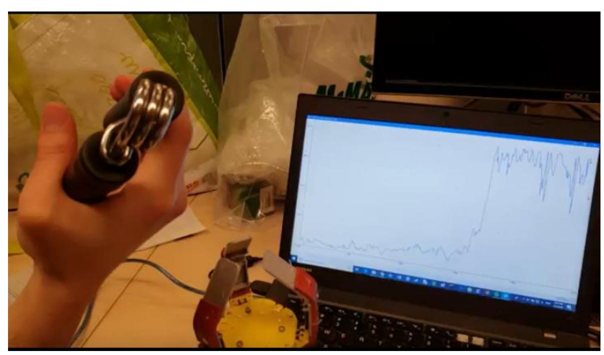

Figure 12. Functioning Prototype (Closed Hand in Response to an EMG Signal (Shown on the Screen))

The functionality of the fully completed system was established. Under ideal conditions, the sensor retrieved the correct EMG signal, which was between 50 and 160 serial units on the serial plotter. The Arduino then actuated the servo, which was able to turn from its maximum angle of 180° (open) to its minimum angle of 135° (closed).

It was also observed that there was slight time delay between the actuation of the servo and the movement of the prosthetic fingers. This behaviour was due to the nature of the differential disk and the fishing line. The differential disk needed time for the fishing line of the finger tendons to become taut in order to move the fingers. This was only a minor time delay.

V. DISCUSSION

A. Comments for Calibration

After several attempts in calibration, mainly due to the nature of the sensor, a few graphs were produced (Figures 9, 10 and 11). As previously stated, the final calibration graph was able to give us the MVIC value, since it had the largest peak serial value. However, all three graphs had very close values. This was substantial for the system because it pointed to a high repeatability since muscle fatigue was controlled.

For this project, calibration was done with the MVIC normalization method which looked solely at isometric contraction of the flexor digitorum superficialis in gripping.

However, there are some gripping tasks employ multiple muscles in the forearm as demonstrated by a study done by Ngo and Wells. In this study, various gripping tasks were carried out and the normalized EMG amplitudes for those tasks were determined by taking the exertion of a maximum gripping force as the reference task (this reference task is the same as that used in the calibration process of the EMG sensor for this project). Using this reference task, this study showed that there are gripping tasks that produced normalized EMG values larger than 1. For example, resisted moment gripping tasks (e.g. squeezing an object while flexing the wrist to resist induced wrist extension) produced normalized EMG amplitudes of up to 2.8 ($p < 0.05$). This study concluded that researchers wishing to normalize forearm activity during power gripping tasks should use resisted flexor and extensor moment tasks (as opposed to a simple squeezing task) to obtain better estimates of the forearm muscles' maximum electrical activation magnitudes. Since these resisted moment tasks use multiple muscles in the forearm, multiple EMG sensors instead of just one EMG sensor would need to be used to obtain results that are more accurate than the results obtained for this project. These sensors would be placed on the muscles responsible for these resisted moment tasks. [17]

B. Comments on the Implementation of the Electrical and Mechanical Systems

Overall, the system was extremely difficult to fully implement. Since the system was wholly based on one sensor for all input (EMG), every other component was affected by the faulty sensor. The EMG sensor that was purchased required manual solder. The solder had to be redone several times, since the wires would bend and break. Eventually, the positive terminal connection had broken. Sodor was applied to secondary connection to the positive terminal, so power was restored to the sensor.

Eventually, the sensor functioned properly. As previously described, the sensor would oscillate at a constant serial value (+/- 3) and this value was not affected by stimulus from the electrodes via muscle flexion.

In the future, a more robust sensor could be used with less difficulty. Another option would be a redundancy system, where if one sensor failed another could take its place.

On the coding side, the exponential smoothing algorithm did not function as well as planned. It performed slightly better than just simply averaging the real time values. The EMG signal contained a lot of noise, evident by Figures 9, 10 and 11. Although these values were smoothed by the algorithm, because they had such a large frequency and varying amplitudes, the algorithm could not smoothen them perfectly. It was also difficult to bandpass the serial values, since further restrictions of the already tight band filter would affect the shape of the waveforms. Data loss is extremely undesirable when running this type of real-time system.

Finally, the servo driver also had some difficulty, again due to the noisy signal. In order to implement speed proportionality, the speed of the servo would vary a lot, since the data was polled from the sensor every millisecond. Sharp spikes with large positive or negative amplitudes with respect to the average would cause the servo motor to twitch. In order to fix this issue, the code would have to attempt to project the course of the motion of the hand, which was difficult to implement and did not fit our time constraint. Instead, it was written such that the hand would continue to close regardless, until the hand was closed, in which case servo would attempt to open should the command to do so, be received.

VI. CONCLUSION

The implementation of the EMG prosthetic system was difficult. First, there was not enough time required to design the mechanical system from scratch. Instead, the open-source bionic hand was used in order to substantially reduce the amount of time required to build the mechanical system and maximize functionality.

Also, the signal quality from the EMG sensor was low. It did not work a lot of the time, but when it did, it had a lot of noise. Testing the system was difficult since this was the only sensor.

Aside from these mishaps, core functionality of the system were achieved. Speed proportionality was implemented, which allowed more control over the system by the patient. The signal was also conditioned in order to make the prosthetic function properly. Finally, the functional mechanical system was achieved. This mechanical system functioned well with little to no time delay from the servo.

In the future, there are a few things that would be able to further improve this system. Since the mechanical system needed to be built in a short amount of time, it was made out of 3D-printed plasted, which was not strong enough for everyday patient use. Fabricating the prosthetic out of aluminum or a stronger polymer would improve the durability and shock resistance of the prosthetic for patient use.

A strap and casing for the physical system would also be required in the future. The casing would protect the electrical system, as well as make the system aesthetically pleasing. The strap would be attached the system to the patient's arm, to allow the patient to wield the prosthetic as intended.

Another improvement would be the use of more advanced mathematical models in the signal processing. A simple exponential smoothing algorithm was used, but in the future the smoothing could be expanded upon. Perhaps using a double exponential smoothing algorithm would further improve the signal.

Added degrees of freedom in the wrist would also benefit the patient. Wrist flexion/extension and pronation/supination would allow the patient more control over the prosthetic, imitating a natural hand.

Finally, a more reliable medical-grade EMG sensor could be used. Signal quality would improve, making it easier to design code to control the prosthetic. Higher reliability would also mean that the number of follow-ups would be reduced since the sensors lifetime would be longer.

The full design of this system was extremely useful in increasing appreciation for the design of these medical mechatronics. Due to the long lifetime and high durability requirements of this system, a lot of design work must be put in.

VII. REFERENCES

[1] A. Chadwell, L. Kenney, S. Thies, A. Galpin, and J. Head, "Corrigendum: The Reality of Myoelectric Prostheses: Understanding What Makes These Devices Difficult for Some Users to Control," Frontiers in Neurorobotics, vol. 12, 2018.

[2] L. Mademli, "The physiological background of EMG."

[3] M. Haiderbhai, "Pathophysiology and Medical Devices," in Bioinstrumentation Tutorial 1, 22-Jan-2019.

[4] S. Basiratzadeh, "MCG 4150 - Bioinstrumentation Laboratory 2 EMG," in Bioinstrumentation Laboratory 2, 04-Mar-2019.

[5] OpenBionics: Open-Source Robotic and Bionic Hands, "Robotic and Bionic Hands," OPENBIONICS. [Online]. Available: https://www.openbionics.org/. [Accessed: 05-Mar-2019].

[6] S. McHugh, YouTube, 16-Apr-2007. [Online].

Available: https://www.youtube.com/watch?v=Ii7ByNe5SiM. [Accessed: 12-Mar-2019].

[7] What factors affect EMG Signal Quality? [Online]. Available: https://www.delsys.com/emgworks/signal-quality-monitor/factors/. [Accessed: 22-Jan-2019].

[8] B. Kaminski, "MyoWare(R) Muscle Sensor (Product ID AT-04-001)," MyoWare(R) Muscle Sensor (Product ID AT-04-001), 13-Feb-2019. Obtained more information about the technical specifications of the MyoWare Muscle Sensor in this email

[9] Digi-Key, [Online]. Available: https://www.digikey.ca/products/en?keywords=SEN-13723. [Accessed: 03-Feb-2019].

[10] N. Vun, Differential Mode and Common Mode Gain of Differential Amplifier. [Online]. Available: http://www.ntu.edu.sg/home/aschvun/FAQ/DiffAmp.html. [Accessed: 13-Feb-2019].

[11] A. Abbas, “MCG 4150 Biomedical Instrumentations,” in Bioinstrumentation Lecture 2, 22-Jan-2019.

[12] Digi-Key, [Online]. Available: https://www.digikey.ca/product-detail/en/sparkfun-electronics/ROB-11965/1568-1319-ND/5766906. [Accessed: 03-Feb-2019].

[13] “Ultimaker 2 : Engineered to perform | Ultimaker,” Ultimaker.com. [Online]. Available: https://ultimaker.com/en/products/ultimaker-2-plus. [Accessed: 06-Apr-2019].

[14] K. Hartman, “Getting Started with MyoWare Muscle Sensor,” Placing Electrodes | Getting Started with MyoWare Muscle Sensor | Adafruit Learning System. [Online]. Available: https://learn.adafruit.com/getting-started-with-myoware-muscle-sensor/placing-electrodes. [Accessed: 06-Apr-2019].

[15] S. Schafroth, Time Series Analysis - Basic Techniques. [Online]. Available: http://www.biosym.uzh.ch/modules/models/Modeling_Basics/Time_Series/timeseries.xhtml. [Accessed: 20-Mar-2019].

[16] Y. Wu, M. N. M. Martnez, and P. O. Balaguer, “Overview of the Application of EMG Recording in the Diagnosis and Approach of Neurological Disorders,” Electrodiagnosis in New Frontiers of Clinical Research, 2013.

[17] J. Reschke, “HTTP Authentication-Info and Proxy-Authentication-Info Response Header Fields,” Elsevier, 2015.

[18] M. Halaki and K. Gi, “Normalization of EMG Signals: To Normalize or Not to Normalize and What to Normalize to?,” Computational Intelligence in Electromyography Analysis - A Perspective on Current Applications and Future Challenges, 2012.

VIII. APPENDIX A

Parts 1 and 2 of the Arduino code used to implement the algorithm described in sections III, H and III, F are displayed in Figures A1 and A2 respectively.

```
#include <VarSpeedServo.h>
VarSpeedServo servo1;
int algCounter = 0; //time the algorithm
int servoTimer = 0; //time the servo macromovement

//Exponential smoothing
int x_i = 0; //raw data point from sensor
int x_i_prev = 0; //previous raw data value value
int s_i = 0; //smoothened data point for output to serial
int s_i_prev = 0; //previous smoothened value
int a = 0.5; //smoothing factor {0 < a < 1}, gives lowest MSE (mean square error)

//Servo driver
int MAX_VALUE = 134; //Max of 3 full strength EMG measurements
int MAX_ANGLE = 135; //Max angle the servo will turn (fully closed hand)
int MIN_ANGLE = 180;
int normValue = 0; //Normalized value
int mapped_value = 0; //Mapped value norm->speed
bool status = true; //hand is open initially

void setup() {
  Serial.begin(9600);
  servo1.attach(A1); //analog pin A1
  servo1.slowmove(MIN_ANGLE, 150); //servo to position 0
}
void loop() {
  int sensorValue = analogRead(A0); //sensor to analog pin A0
 if(algCounter == 0){ //initial i = 0
   s_i = sensorValue; //smoothened value is raw data value for first point i = 0
   Serial.println(s_i); //output the i = 0 value
   s_i_prev = s_i; //save s_i in s_(i-1)
 }
 else{ //i > 0
   s_i = (a * sensorValue) + ((1 - a)*s_i_prev); //smoothened value eqn for i > 0
   Serial.println(s_i); //output the smoothened value
   s_i_prev = s_i; //save s_i for the i + 1 iteration
 }
  algCounter++;
```

Figure A1. Arduino Code - Part 1

```
normValue = sensorValue / MAX_VALUE; //find normalized value from MVIC
if(normValue > 1){
  normValue = 1; //restrict normalized value 0 < x < 1
}
mapped_value = map(normValue, 0, 1, 1, 200); //Map the normalized value between [0,1] to [1,255] for speed control
if(status){ //if hand status is open
  servo1.slowmove(MAX_ANGLE, mapped_value);
  if(servoTimer >= 500){ //Arbitrary time constant, makes it so that servo doesn't attempt to close before fully open
    status = !status; //Hand status is now closed
    servoTimer = 0; //Reset timer
  }

}
else{
  servo1.slowmove(MIN_ANGLE, 200); //Open the hand at constant rate
  status = !status; //Hand status is not open
}
delay(1); // delay in between reads for stability
```

Figure A2. Arduino Code - Part 2