Sensing, Classifying Movements, and Actuation of Biomimetic Robots

by

Scott M. Kennedy

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
August 3, 2019

Keywords: electromyography, motion classification, shape memory alloy, actuator

Approved by

Michael Zabala, Chair, Assistant Professor of Mechanical Engineering
Edmon Perkins, Assistant Professor of Mechanical Engineering
Mark Schall, Assistant Professor of Industrial and Systems Engineering

Abstract

Human skeletal muscle motion is a complex electrical, chemical, and mechanical process. The electrical, or electromyography, signal originates from the brain and travels to the muscle initiating the contraction of the muscle. This signal can be detected in the muscle roughly 500 ms before movement occurs and can be used to classify the motion of the subject to inform the motion of an assistive robot. The electromyography signal can be monitored by surface electromyography sensors adhered to the subject's skin. These sensors are highly dependent on position and orientation of the sensor with respect to the belly of the muscle being monitored and requires both knowledge of musculature and extensive setup time to use. In order to implement a better sensor solution capable of being applied without any knowledge of anatomy, a textile suit was outfitted with electromyography sensors. The suit was shown to accurately classify the motion of the user using a K-nearest neighbor algorithm compared to the traditional method of sensor adhesion. This suit works to inform the movement of an exoskeleton. Along with detecting intent of motion from the user, exoskeletons and assistive robots need to move like their users. Shape memory alloy actuators have previously been used as micro-actuators. With a high strength to weight ratio, small form factor, and low cost, shape memory alloys are an attractive actuator option that contracts and lengthens similarly to human skeletal muscles. However, shape memory alloy actuators also have high power consumption, low strain (4-8%), and low operating frequency (< 3 Hz). In order to overcome these drawbacks, a shape memory alloy actuator was designed and tested in a bimorph configuration. This shape memory alloy actuator design addressed the drawbacks of traditional shape memory alloy actuators and could potentially expand the usage of this technology in biomimetic robots.

Acknowledgments

I would like to express my gratitude to my supervisor Dr. Michael Zabala for the constant support, mentorship, and engagement through this master's thesis. I would also like to thank my Dr. Edmon Perkins and Dr. Mark Schall for their support and assistance with the various projects that I worked on during my master's program. I would like to thank Morgan Price and the other members of the Auburn University Biomechanical Engineering Lab for their support.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Biomimetic robots and exoskeletons that move in parallel with a human operator present more complex problems compared to traditional robotics. For biomimetic robots, mimicking the motions of biological creatures involves complex joints and actuators. For exoskeletons, sensing the intent of motion from the user in order to inform the motion of the exoskeletons is needed for smooth operation. Biological creatures use skeletal muscles in order to actuate their limbs and interact with the physical world. Skeletal muscles are linear actuators with a complex microstructure and an activation process with electrical, mechanical, and chemical components. This process along with the ligaments that stabilize the joints allow biological creatures to create complex motions.

Exoskeletons are being used in both the medical and defense industry in order to augment the strength and motion of the user. Patients that have sustained a serious injury in which they have lost the use of one or more extremities, such as their legs and can no longer walk, can use exoskeletons to either regain or supplement this motion. Exoskeletons can also be used as strength augmentation devices in order to allow the user to carry more weight or walk/run for a longer period of time. In order for these various types of exoskeletons to be the most effective, the exoskeleton must move in sync with the user as to not inhibit the motion of one another. Traditionally, force/pressure sensors in the exoskeleton have been used to detect the intended motion of the user. This produces a suit that lags behind the user rather than moving in time with them. Electromyography, which is the electric potential generated in your muscle prior to muscle contraction, can be used to detect and classify motion prior to the motion occurring. Recent efforts have focused on using this signal in order to inform the motion of exoskeletons.

Advancement of biomimetic robots depends on the advancement of the actuators that move them and the batteries that power them. Research in artificial muscle technology such as shape memory alloys could potentially improve the motion of the robots and allow robots to move more similarly to their biological counterparts. Shape memory alloys are a type of soft actuator that can "remember" its shape at various temperatures. Shape memory alloy actuators are typically trained to contract at high temperatures and extend at low temperatures similarly to how human skeletal muscles contract and extend in length. Currently, shape memory alloy actuators have been limited by high power consumption and low displacement and operational frequency requirements.

In this thesis, the use of textile-embedded surface electromyography sensors for motion classification during dynamic actions was investigated. Surface electromyography sensors were adhered to various lower limb and midsection muscles in order to detect the action potential generated in the muscle. The sensor data was then fed into a K-nearest neighbor algorithm in order to classify the motion being performed by the subject. Both the signal quality and motion classification accuracy was compared using the method of traditional sensor application with a textile embedded sensor suit using the same sensors sewn into a commercial off-the-shelf garment.

For the advancement of biomimetic actuators, a bimorph shape memory alloy actuator was designed and tested. This bimorph actuator was designed using the smallest diameter of nitinol wire while minimizing length. These two qualities allow the shape memory alloy to change temperature quickly while maintaining a low power consumption. The bimorph actuator also allows for higher operational frequencies and increased displacement at the actuator's natural frequency. Bimorph actuators of various lengths were made and tested in order to determine their natural frequencies and other dynamic characteristics.

Chapter 2

Background and Literature Review

2.1 Wearable Robotics

The development of exoskeletons, active prostheses, and wearable robotics has increased in recent years [1]. Exoskeletons can be divided into two classes: passive and active. Passive exoskeletons, or those that do not consume power, attempt to reduce the metabolic cost of certain activities performed by the wearer. Active exoskeletons, or those that do consume power, are typically actuated by electric motors, hydraulic actuators, pneumatic actuators, or a combination of these [2–7]. Many different control strategies have been proposed for active devices [8]. Controlling the movement of an exoskeleton based on movement that has already occurred (e.g., using data collected from inertial measurement units [IMUs]) requires reactive control rather than predictive control. This is not ideal for exoskeleton operation because this delay could hinder, rather than assist, the motion of the operator. Biological signals generated by the operator, such as the electrical activity associated with muscle recruitment collected using surface electromyography (EMG), are advantageous and can be sensed before motion ensues. If processed fast enough, this information would allow an exoskeleton to move in harmony with the operator, rather than lag behind due to latency. Previous studies have used EMG to classify motions for active prostheses, such as pattern recognition in arm prostheses [9–12], locomotion classification for lower limb active prostheses [13–17], and motion classification for upper limb prostheses [18, 19].

2.2 Skeletal Muscles and Electromyography

Humans are made of three different types of muscles: smooth, cardiac, and skeletal. Smooth muscles are responsible for the contraction of hollow organs, such as blood vessels. Cardiac muscles compose the heart and produce rhythmic beating due to their intertwined structure. Smooth and cardiac muscles produce involuntary motion. Skeletal muscles are responsible for voluntary motions produced by humans, such as flexing and extending the elbow joint. Skeletal muscles can only produce force while pulling and working in antagonistic pairs to actuate a joint. An antagonistic pair is a pair of muscles where one contracts and actuate the joint in the direction of which it pulls, while the other muscle relaxes/extends and produces a force that resists the motion. An example of an antagonistic pair of muscles in the human body is the biceps brachii and the triceps that produce flexion and extension, respectively, about the elbow.

Skeletal muscles are organized in a unique way in order to optimize efficiency and power production. Skeletal muscles are made of a combination of passive and active materials. The muscle as a whole (such as the bicep brachii) is made up of fascicles. Fascicles are bundles of 10-100 muscle fibers (muscle cells). The muscle fibers are made of bundles of myofibrils, which consist of a series combination of sarcomeres. The main filaments of contraction, actin and myosin, are housed in the sarcomere. Along with these structures that house the active components of skeletal muscles, there are many elastic structures that hold the active structures together. The epimysium surrounds the whole muscle. The perimysium surrounds the fascicles. The endomysium separates individual muscle cells from one another in the fascicle. While these elastic structures don't contribute to the active component of force generation of the muscle, they do provide structural support and passive force generation when the muscle is extended beyond its resting length.

The process of muscle contraction begins when a nerve impulse reaches an axon terminal, acetylcholine (ACh) is released. ACh diffuses to the receptors on the muscle fiber membrane (sarcolemma) and Na+ channels open and allow Na+ to spread through the cell. This results

in an action potential to spread over the sarcolemma and through channels in the muscle cell called the transverse tubules. The action potential in the muscle cell is the signal that is recorded during electromyography testing. The action potential causes a release of Ca+2 which binds to troponin resulting in the revealing of binding sites in the myofibril allowing for contraction to happen between the thick and thin filaments.

The thick filament (primarily composed of myosin filament) and thin filament (primarily composed of actin filament) are the primary contributors to the active force generation of the skeletal muscles. The think filament is a bundle of individual myosin filaments. The bundle consists of pairs of myosin filaments twisted together with the head extending off the structure. The predominate theory on muscle contraction is called the sliding filament theory. Once the binding sites are exposed on the actin filament, adenosine triphosphate (ATP) causes the think filament to extend, bind to the thin filament and pull resulting in a sliding motion of the two filaments over one another producing an overall contraction of muscle length.

2.2.1 Traditional Surface Electromyography

The electromyography signal can be captured with both invasive and noninvasive techniques. Invasive EMG, often times called needle EMG, involves sticking a needle like probe into the belly of the muscle for data acquisition. This type of sensing technique is more accurate than noninvasive EMG. However, the insertion of a needle into a muscle during dynamic activities can cause discomfort and change the motion, and therefore the EMG signal, that it is intended for capture. Noninvasive EMG, or surface EMG, sensors monitor the signal on the surface of the skin. Surface EMG signals can be easily collected from the surface of the skin [20, 21], but signal quality is highly dependent on the location of the sensor, the tissue composition of the subject, and the skin preparation technique [22]. In order to use surface EMG sensors, electrodes are adhered to the skin on the belly of the muscle being monitored. While the sensor does add additional mass to the segment, there is considerably

less discomfort during dynamic tasks compared to invasive EMG sensors. Since the signal is measured on the surface, the signal is not as "clean" as invasive EMG. The action potential that is generated in the muscle prior to contraction is propagated through the various layers of muscle, skin, and fat until it reaches the electrodes of the sensor. These various anatomical layers add resistance and, therefore, can change the signal. This propagation of the signal through the skin also causes surface EMG to read signals from neighboring muscles. For example, if a surface EMG sensor is placed to monitor the bicep brachii, the sensor could pick up the EMG signal from the brachialis since both muscles run through the upper arm in relatively similar locations. Invasive EMG does not have this problem as the electrode is placed directly in the muscle of interest. Surface EMG sensors are also susceptible to motion artifacts during dynamic activities. During dynamic activities, the skin can move relative to the muscle. This causes a shift in the sensing area of the sensor resulting in a change in signal characteristics. In order to remove motion artifacts from the actual signal, surface EMG signals can be filtered in the post-processing stage. An example of raw EMG is shown in Figure 2.1.

Traditionally, EMG sensors are adhered to the skin superficial to the muscle belly via double-sided tape after the skin has been shaved, cleaned and abraded with an alcohol wipe. The sensor is then wrapped with pre-wrap to ensure the sensor maintains contact with the skin and to reduce the movement of the sensor as much as possible. This method of application results in the most accurate surface EMG recordings of a single muscle, but the application process adds substantial setup time, especially if a large number of muscles are being monitored. This process can be a hindrance with EMG controlled prostheses and wearable robotics as the operator may not have the time or technical knowledge to properly apply all the EMG sensors that the device requires.

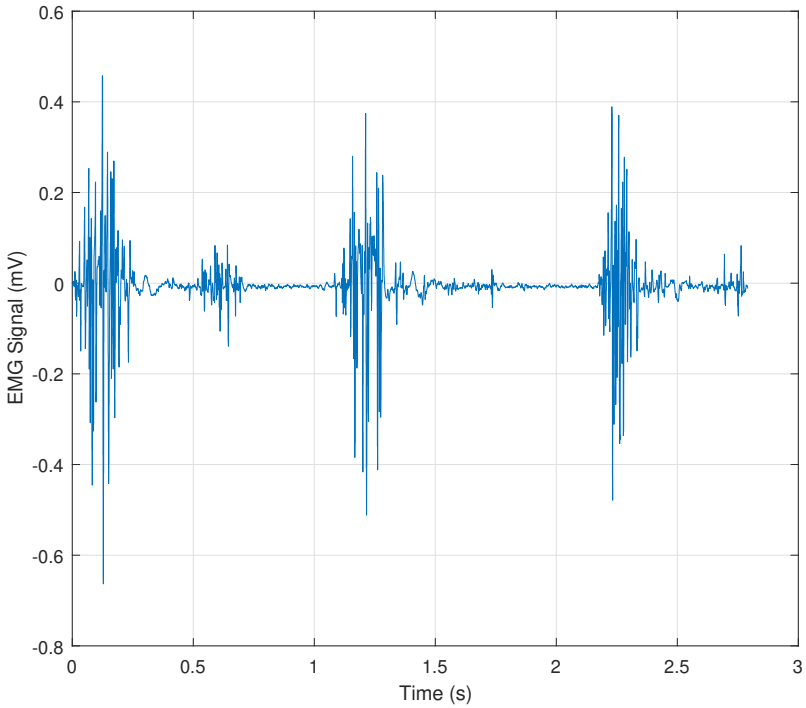


Figure 2.1: Example of raw EMG data

2.2.2 Textile Electromyography

Alternatives to traditional EMG measurement techniques include array-based EMG [23] and textile electrodes [19, 24]. EMG-embedded textile garments aim to circumvent the drawbacks of traditional EMG. Many of these textiles are made by sewing conductive fibers into the fabric to act as the electrodes. These fibers are typically used to collect the EMG data from a group of muscles, whereas traditional electrodes monitor a single muscle. Since these sensors are embedded in fabric rather than adhered to the skin, the sensors can move relative to the skin causing movement artifacts to appear in the signal, which in turn may change its characteristics. Previous studies have been conducted comparing the normalized average rectified value (NARV) of EMG signals using traditional electrodes and textile-based electrodes during isometric, low velocity activities [25], and highly dynamic activities [24]. The studies that considered isometric and low velocity activities observed good in-session comparability between the two electrode designs. However, since the movements studied

were isometric and low velocity, the possibility of movement artifacts due to the lack of skin-to-sensor fixation was likely reduced. Colyer and McGuigan [24] compared the two methods of EMG sensing during the dynamic activities of walking, running, cycling, and squatting. They suggest that the two methods were comparable during lower activation levels, but increased in variability as the activation levels increased.

2.2.3 Electromyography Signal Processing

In order for EMG data to be useful, it has to be processed from its raw state. Typically the process involves rectifying, filtering, averaging, and then normalizing. The raw signal (Figure 2.1) has both positive and negative values. However, only the magnitude of the signal is of importance and not the direction (positive/negative). Therefore, the simplest way of rectifying the signal is taking the absolute value. Once the signal is rectified, the signal is filtered in order to remove any parts of the signal that are not the action potential of the muscle such as movement artifacts. It is common practice to apply a 6th order Butterworth bandpass filter with cutoff frequencies around 20 and 200 Hz. The part of the signal associated with motion artifacts is the lower end of the frequency specturum which is why the frequency components from 0 to 20 Hz is filtered out of the signal. The frequency component of 200 Hz and above is filtered out to remove any high frequency noise. There is variation of the cuttoff frequencies used in the published literature, but generally speaking 20 and 200 are the frequencies used in the filter.

This rectified and filter signal is then commonly averaged over a window of time. The last step in processing is normalizing the signal. The raw signal is usually on the scale of millivolts. The raw magnitude of the signal does not indicate anything about the level of activation or give any valuable information about the action performed. As mentioned earlier when using surface EMG sensors, the signal is propagated through various anatomical layers. These layers change the magnitude of the signal depending on their thickness and various properties. Therefore, depending on the person, the magnitude of the signal can be different if

their anatomical layers are different despite the muscle producing similar forces. Therefore, the signal needs to be normalized in order to compare across subjects and extract useful information such as percent activation. There are numerous ways to normalize an EMG signal. The most common is using a maximum voluntary isometric contraction (MVC). This is done by having the subject activate the muscle as much as possible and applying a variable resistance force in order to stop the joint from moving. For example, in order to achieve a MVC for the bicep brachii, the subject will flex his arm as much as possible while a resistive force is applied to the forearm in order to hold the angle between the upper arm and forearm constant. An MVC needs to be recorded for each muscle used in the study and, therefore, requires significant time to collect and also requires additional personal to apply the resistive force. Another technique of normalizing is normalizing to a well-practiced activity such as walking. This method is beneficial because normalizing data can be collected from multiple muscles during a single trial. Since a motion like walking is well practiced, the EMG signal for the muscles involved in this motion are consistent for that day.

2.3 Traditional Actuators

Robots have traditionally been actuated by electric motors, combustion engines, and pneumatic/hydraulic actuators. These actuators have been well studied thus making these actuators widely available, easily implemented, and have well defined control theories. However, these actuators typically have miniaturization problems, limited degrees of freedom, and low strength to weight ratios. One of the biggest drawbacks of traditional actuators is their low strength to weight ratios. This causes traditionally actuated robots to be bulky and limits their usefulness. Along with a low strength to weight ratio, traditional actuators have limited degrees of freedom. For example, a standard electric motor has just one degree of freedom and it rotates about its axis. The motor is attached to a pin joint and rotates one of the segments connected at the joint. In order to have multiple degrees of freedom at a joint, additional motors would need to be used. Finally, traditional actuators can only be

scaled down to a certain size. For example, an electric motor is made up of various parts such as screws, wires and a circuit board. Therefore, a motor can only be scaled down to a certain size depending on the sizes of its parts.

2.4 Soft Actuators

Recently, soft actuators such as ionic polymer-metal composites, shape memory alloys, and dielectric elastomers have been developed as micro-actuators due to their excellent performance at small scales [26–28]. These various types of soft actuators produce linear or bending motion and can be molded to produce unique movements to emulate movements of biological organisms, (eg., a human hand [29] and caterpillar [30]). Many of these actuators perform well in aqueous environments leading to the development of biomimetic underwater soft actuated robots (eg., jellyfish [31–33], turtles [34, 35], and cuttlefish [36]).

Ionic polymer-metal composites (IPMC) are composed of an ionically conductive membrane between two electrodes. The membrane contains the electrolyte, and, upon application of a voltage, cations move from the anode to the cathode resulting in swelling of the cathode and a bending of the actuator [37,38]. Dielectric elastomers (DE) are similar to parallel plate capacitors with a compliant dielectric material between two electrodes. When a voltage is applied to the electrodes, a resulting electric field creates an attracting force between the two electrodes causing the dielectric material to decrease in thickness and increase in length and width [39–41].

2.4.1 Shape Memory Alloys

Shape memory alloys (SMA) are metals that undergo a phase transformation based on their temperature resulting in macroscopic strain [42, 43]. SMAs exhibit two working principles, the *shape memory effect* and the *pseudo-elastic effect*. These principles produce the same motion, but the motion is achieved by different means. The shape memory effect is achieved by cycling temperature, whereas the pseudo-elastic effect is achieved by cycling

stress. For actuation purposes, use of the shape memory effect is desired over the pseudo-elastic effect since the temperature of the SMA is easier to manipulate than the stress on the SMA. At low temperatures and low stress, the SMA is in its "twinned" martensite phase. With the application of stress, the material is stretched into its "detwinned" martensite phase (seemingly undergoing plastic deformation). Upon the application of heat, the material will transform into the austenite phase and the material will recover the seemingly plastic deformation and return to its short, compact structure. Substantial research and resources have been invested into the construction of phase kinetic models and constitutive models describing the behavior of SMA actuators while in use [44–50].

SMAs are an attractive actuator option compared to other types of soft actuators due to their high strength to weight ratio, inexpensive cost, commercial availability, and small form factor [51, 52]. Typically, SMA actuators are heated by resistive heating and are capable of pulling payloads that are 500 times their own weight [53]. The most common shape memory alloy is nickel-titanium (NiTi), commonly referred to as nitinol. In order to achieve useful stroke and force generation, nitinol actuators are manufactured in the form of a wire and trained to shorten in length ($\sim$4%) as its temperature increases past the austenite transformation temperature.

Composite SMA actuators aim to transform the linear motion of SMA actuators into bending/twisting motions. The most common way that this is done is by creating a unimorph or bimorph structure. A unimorph actuator has a SMA layer coupled with a passive noncompressible layer (commonly composed of a plastic such as polylactic acid). Since the SMA is offset from the centerline, a shortening in the SMA wire (when heated) results in a bending motion of the passive layer. As the SMA is cooled, the energy stored in the passive layer as it is bent returns the unimorph to its original shape. A bimorph has two active layers on both sides of a passive layer. The active layers work in antagonistic pairs to actuate the bimorph. As the top layer is heated and contracted, the bottom layer is cooled (passively) and lengthened. This causes the bimorph actuator to bend towards the top layer. Then,

the bottom layer is heated while the top layer is cooled, which causes the bimorph to bend towards the bottom layer.

Since SMA actuators rely on the temperature change of the actuator, their response time is relatively slow compared to other types of soft actuators previously mentioned, which is a major drawback of these actuators. However, recent research aimed at developing methods of increasing the frequency of SMA actuators has been conducted, including using multiple smaller wires [35] and using a wire with a diamond-shaped frame lattice cut by an ion beam [54].

Song et al. developed a high frequency SMA bimorph bending actuator using multiple small diameter SMA wires of various lengths with an acrylonitrile butadiene styrene (ABS) passive layer [35]. They found that there was an increase in deformation length and angle at certain frequencies and that the frequencies increased as the SMA actuator decreased in length. However, they reported that their actuator required 50 watts or more for operation. In order for high frequency and high displacement SMA actuators to advance the development of biomimetic untethered robots, a low power and durable SMA bimorph actuator needs to be developed.

2.4.2 Unimorph and Bimorph Actuators

Unimorph and bimorph actuator configurations have been used with multiple different types of soft actuators. The main purpose of using one of these configurations is to augment the strain and frequency of actuation. The focus of this thesis is shape memory alloys. Thus, unimorph and bimorph configurations that use shape memory alloys will be investigated, but it is worth noting that this principles can be applied to other soft actuators.

The unimorph actuator configuration is simply a passive material (such as PLA, ABS, some type of metal or rubber) adhered to the active material (the SMA in the case) as shown in Figure 2.2.

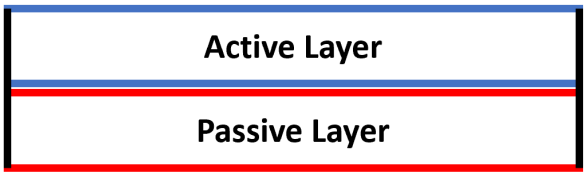


Figure 2.2: Unimorph actuator design

The SMAs used in this paper are trained to shorten in length as they transform into the austenite phase. Therefore, when the SMA layer deforms, the passive layer also deforms since the ends of the two layers are constrained together and the passive layer is incompressible (Figure 2.3). Once the SMA layer excitation signal is turned off and the SMA layer begins to cool, the passive layer acts as the bias spring and pulls the active layer back into its original state (Figure 2.2).

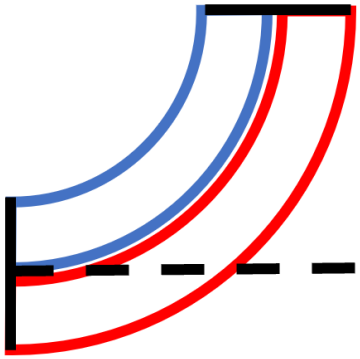


Figure 2.3: Unimorph actuator when deformed (Blue = SMA, Red = Passive layer)

The structure of the unimorph and bimorph actuator transforms the motion of soft actuators and augments their strain. Traditional SMA acutators are trained to contract in length when heated and have been reported as having 4-8% length change meaning that a 100 mm linear SMA actuator would only be capable of 4-8 mm of displacement. However, SMAs in a unimorph or bimorph configuration have been shown to produce much more displacement [32]. This is due to the constraints on the actuator. Since the SMA layer and passive layer are adhered together on both ends, their radius of curvature will remain the same when they bend.

$$\frac{R_{SMA}}{R_P} = \frac{(1-\epsilon)L_{SMA}}{L_P} = (1-\epsilon) \tag{2.1}$$

$$R_P = R_{SMA} + d \tag{2.2}$$

Where R is the radius of curvature, ϵ is the strain of the SMA, and L is the length (P denotes quantities related to the passive layer). The two equations simplify to the following relationship

$$R_P = \frac{d}{\epsilon} \tag{2.3}$$

As the equation shows, in the unimorph configuration, the displacement becomes a function of the strain from the active layer and the distance between the midline of the active layer and the midline of the passive layer.

The bimorph configuration is similar to the unimorph configuration except that it has an active layer on both sides of the passive layer (Figure 2.4).

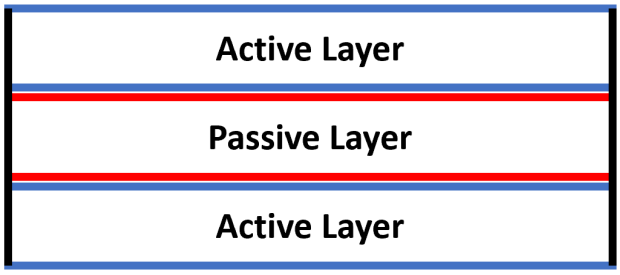


Figure 2.4: Bimorph actuator design

The actuation cycle for a bimorph actuator is as follows:

1. One of the active layers (layer #1) is excited resulting in bending motion of the whole structure in the direction of the excited active layer.

2. The excitation signal to layer #1 is stopped.

3. The excitation signal to the other layer (layer #2) is started.

4. The structure bends towards layer #2.

5. The excitation signal to layer #2 is stopped.

In order to create an oscillatory response with a bimorph actuator, the cycle is repeated. The bimorph actuator has many advantages compared to the unimorph actuator. Unimorphs are limited to one direction of movement since the unimorph configuration has only one active layer. Unimorph's operational frequency is also slower than that of the bimorph. The passive layer is responsible for returning the actuator from its actuated (bent) state to the un-actuated (initial) state. Therefore, the operational frequency is dependent on how quickly the passive layer pulls the active layer back to its initial state before the next actuation cycle can begin. Conversely, bimorph actuators have two active layers that are alternatively excited allowing one active layer to "pull" the opposing active layer out of its actuated state. The antagonistic pairs of active layers allow bimorph actuators to be oscillated at the frequency of excitation signal switching. However, since the bimorph actuator has two active layers, it consumes more power than the unimorph.

Chapter 3

Electromyography Clothing to Detect the Electrical Signal in Muscles and Classify Motion

The focus of this chapter is the comparison of surface EMG applied in a traditional manner (skin shaved and cleaned with alcohol with electrode taped to skin) and EMG compressed to unprepared skin via a textile base suit built using commercial off-the-shelf products. These EMG signals are then fed into a K-nearest neighbor classifier and used to classify the motions of the wearer in order to advance exoskeleton control.

3.1 Subjects

A total of three (3) subjects were tested with the protocol described below. Various anthropometric characteristics of the three subjects are shown in Table 3.1.

Table 3.1: Anthropometric values for subjects used in data collection.

Subject	Height (m)	Weight (lbs)	Age (yrs)
1	1.75	145	25
2	1.72	140	33
3	1.71	183	25

Prior to participating in the study, all subjects provided written informed consent. All subjects had no lower limb injury in the last six months, nor any history of adhesive allergies. All study procedures were approved by the Auburn University Institutional Review Board.

3.2 Equipment

Gold-standard kinematic data was collected using a 10-camera motion capture system (Vicon, Vantage V5 Wide Optics cameras, each with 22 high power IR LED strobe at 85

nm; Vicon Motion Systems Ltd, Oxford Industrial Park, Oxford, UK), and ground reaction force data was collected using two force plates (ATMI BP400600, 2000 lb. capacity; Advanced Mechanical Technology, Inc., Watertown, MA). Seventy-nine reflective markers were placed on participants using double-sided tape using the Point Cluster Technique for maker placement [55].

Continuous surface EMG (bandwidth 20-450 Hz, collected at 1111 Hz) and motion data was acquired simultaneously and bilaterally using fourteen (14) Delsys Trigno IM sensors that combine pre-amplified EMG electrodes with an IMU (accelerometer [148 Hz], gyroscope [148 Hz], and magnetometer [74 Hz]; Delsys Trigno; Delsys Inc., Boston, MA). Each sensor contained four silver bar contacts that allow for two differential EMG inputs with two patented stabilizing references. The focus of this chapter is the comparison of surface EMG applied in a traditional manner (skin shaved and cleaned with alcohol with electrode taped to skin) and EMG compressed to unprepared skin via a textile base suit. As such, only the EMG data from the Delsys sensors was analyzed and used to train the classification model. The motion data from the IMU sensors was considered outside of the scope of this study, and thus was not included.

For both traditional and suit-based testing, the EMG electrodes were located on the surface of the skin directly over the bellies of the muscle listed below using published guidelines [56]:

- Gluteus maximus (one half the distance between the greater trochanter and the sacral vertebrae at the level of the trochanter on an oblique angle, EMG #1 & 2).
- Biceps femoris (one fifth the distance from the gluteal fold to the bag of the leg, EMG #3 & 4).
- Rectus femoris (one third the distance between the knee and iliac spine, EMG #5 & 6).

- Gastrocnemius (just distal from the knee approximately 1-2 cm medial to the midline, EMG #7 & 8).
- Tibialis Anterior (parallel to, and just lateral to the medial shaft of the tibia, at approximately 1/3 the distance between the knee and ankle, EMG #9 & 10).
- Erector spinae (parallel to the spine approximately 2 cm lateral to the spinous process, EMG #11 & 12).
- Rectus abdominus (approximately 2 cm lateral to the umbilicus, EMG #13 & 14).

These specific sensor locations were selected as they provide muscle activation information associated with the initiation, maintenance, and inhibition phases of the activities described in the experimental protocol described below. All raw EMG signals were digitized at 1000 Hz using the Delsys EMGworks software. Finally, linear interpolation was used on all sensor data that resulted in a rate of 2160 Hz for the development of the classification algorithm to ensure the same number of data points for all signals collected. All data collection took place in the Auburn University Biomechanical Engineering (AUBE) Laboratory utilizing Nexus software (Version 2.6.1) and a Lock+ Sync Box (Vicon) to ensure synchronicity of all collected data.

3.3 Sensor Embedded Garment.

For the sensor-embedded textile EMG testing, a suit was created using an Under Armour compression shirt and pants (Under Armour, Baltimore, MD). A custom textile-sensor interfacing clip was constructed in Solidworks and 3D printed in polylactic acid (PLA). The clip (shown in Figure 3.1) was designed to allow the Delsys Trigno sensors to attach and detach from the suit to be charged between uses.

Surface EMG signals are highly sensitive to the placement of the sensor in relation to the desired muscle. In order to achieve optimal sensor location, the garments were initially

Figure 3.1: Model of 3D printed sensor-textile interface clip

put on by Subject #1 (Reference Subject) and the locations of the muscles of interest were marked on the garments. The clips were then sewn into the compression garment at the corresponding markings (Figure 3.2). This suit was tailored to Subject #1, but used for all other subjects in the study. Therefore, this initial process of locating and marking of the muscles was only done once at the beginning of the data collection process for Subject #1. All other subjects were instructed to put on the sensor-embedded garment as they would normal clothes.

(a) front (b) back

Figure 3.2: Under Armour compression suit outfitted with sensor-textile interfacing clips

3.4 Sensor Embedded Textile Testing.

This test session utilized the Delsys EMG sensor-embedded compression suit for EMG data collection. The subject was asked to put on the sensor embedded textile suit and then

a researcher inspected the location of the EMG sensors to ensure that each sensor was at the relatively correct location for the corresponding muscle. Once this initial inspection was complete, there were no other inspections to correct any movement of the sensors during the dynamic trials. It is also worth noting that EMG data collected with the suit was always completed first before any shaving of the skin or other skin preparation was done. This simulates undesirable conditions for operation in the suit since there is still hair and dirt that can affect the contact of the sensor embedded EMG electrodes and the skin.

Once the subject put on the EMG-embedded suit, motion capture markers were adhered to the suit. Motion capture data was used to determine the beginning and end of the various tasks performed by the subject. The subject was then asked to perform nine tasks with three trials per task. The order in which the tasks were performed was randomized for each subject using a permutation generator that generates the order of the actions using the built in random number generator in MATLAB. The tasks were as follows:

1. *Walking (W):* Subject walked across the laboratory space at a self-selected speed.

2. *Running (R):* Subject jogged across the laboratory space at a self-selected speed.

3. *Backwards Walking (BW):* Subject walked backwards across the laboratory space at a self-selected speed while carrying a simulated "gun" (polyvinyl chloride (PVC) pipe, 2 inch diameter, 36 inch long) held in the "ready to shoot" position.

4. *Gun Walk (GW):* Subject walked across the laboratory space at a self-selected speed while carrying a simulated "gun" held in the "ready to shoot position.

5. *Gun Run (GR):* Subject jogged across the laboratory space at a self-selected speed while carrying a simulated "gun" held in the "ready to shoot position.

6. *Pivot Left (PL):* Subject walked towards the center of the laboratory space, planted with their left foot, turned to the right and walked to the edge of the lab space, performed with "gun" held in "ready to shoot" position.

7. *Pivot Right (PR):* Subject walked towards the center of the laboratory space, planted with their right foot, turned to the left and walked to the edge of the lab space, performed with "gun" held in "ready to shoot" position.

8. *Angled Reversal and Return (ARR):* Subject walked towards the center of the laboratory space, planted with their right foot, walked backwards three steps at an angle of approximately 45° to initial progression, then walked forward through the center to the edge of the laboratory space, performed with "gun" held in the "ready to shoot" position.

9. *Walk to Kneel (K):* Subject walked towards the center of the laboratory space, dropped into a kneeling position, then rose to continue walking across the laboratory space, performed with "gun" held in the "ready to shoot" position.

These tasks were chosen as common actions that a soldier would perform in the field while in combat since the objective is to develop motion classification capabilities for a combat-ready exoskeleton.

3.5 Traditional Testing.

After the EMG-embedded suit testing, the subject took off the suit and the same Delsys EMG sensors were adhered to the subject. First, the muscle bellies of the muscles previously listed were identified and the area was shaven, cleaned and abraded with an alcohol wipe before the EMG sensor was adhered with double-sided tape. The sensor was then wrapped with pre-wrap to maintain compression and contact with the skin during the various tasks. Once the sensors were applied to the subject, motion capture markers were again placed at various anatomical positions as to determine beginning and end of the tasks. The same nine tasks previously listed were again performed three times each with the traditional approach. The same random permutation generator was used resulting in a different task order. This testing was done immediately after the sensor-embedded textile testing.

3.6 Second Sensor Embedded Textile Testing.

The same subjects were invited to the laboratory for a second round of suit testing on another date following the initial data collection. The protocol for this round of testing was identical to the sensor-embedded textile testing previously described. The order of the tasks were again randomized and the same suit was used from the first round. The second day of suit testing was used to compare the day-to-day repeatability of the suit designed in this study.

3.7 Signal Processing

For signal analysis, the EMG data was processed in MATLAB R2017a (MathWorks, Natick, MA). Each trial was first shifted to a zero mean, rectified, and filtered with a 6th order low pass Butterworth filter with a cutoff frequency of 15 Hz. The signal was then averaged over the length of the trial to yield the average rectified value. Since EMG signals across participants can be highly variable due to physiological differences, the mean average rectified value of the EMG signal during the three walking trials for each participant was used to normalize the values of the other eight actions. This produced a normalized average rectified value (NARV) per muscle, per task performed. Using the walking EMG data as the method of normalization has been shown effective in comparing inter-subject EMG data [57]. For motion classification, the algorithm was written in Python. Confusion matrices were created to illustrate percent correct and incorrect motion classifications while providing an overall accuracy.

3.8 Analysis

Descriptive statistics (mean and standard deviation) were used to describe the traditional and sensor embedded textile measurement techniques measured on the same day.

Bland-Altman plots were used to determine bias between the measurement techniques as well as limits of agreement.

In order to classify the above actions, a K-Nearest Neighbor (K-NN) classifier was developed using a custom distance function. A key piece to this classification algorithm is the definition of the distance between two sensor collections, which is then utilized by the classifier to determine the similarities between two signals. Since the classification algorithm itself is not the subject of this paper, the details of the distance used for the classification algorithm are in Appendix A. In short, the distance function is the sum over all sensors of the l_1 distance (or taxicab metric) between each collection at the sensor divided by the sum of the l_1 norm of each collection at that sensor as a normalization.

Informally, the goal of this classifier was to compare a slice of unknown movement data to each of the known sensor collections and find the "closest example" of known measurement. For example, a given 50 ms slice of EMG data for an unknown action was compared to the "walking" training EMG signal to determine the place that minimizes that distance. However, this was completed for all sensors simultaneously to find the shift that minimizes the distance of all the signals at once. Then, comparing the distance for each action, the algorithm classified the most likely match. Formally, the K-nearest neighbor classifier (with $K = 10$) is defined with training data consisting of 50 ms slices of data each labeled by their known actions. The distance alluded to above is the definition of "nearest" in this context. For all results shown below, a 3-fold cross validation was performed of the models using two trials to train and the remaining data as the test set. In order to create the training set, 50 ms worth of data at 5 ms intervals were extracted within each of the training samples. The test samples were then 100 randomly chosen 50 ms windows from each test trial. 50 ms windows of data was chosen because it was found that the acceptable lag between the EMG signal and movement is 300 ms based on empirical analysis [58].

3.9 Results

The NARV and the standard deviation for the traditional electrode, textile electrodes collected on the same day, and textile electrodes collected on a different day are shown in Figure 3.3. The NARV collected by the traditional electrode was generally higher than the textile electrode. The differences between the NARV of the measurement techniques typically becomes larger for higher activation movements such as "run" and "gun run".

Coefficients of variation were used to interpret the variability between the traditional electrodes and textile embedded electrodes collected on the same day (Figure 3.4). The sensors on the gluteus maximus, erector spinae, and rectus abdominus show high variability when measured by the sensor-embedded textile EMG compared to the traditional EMG.

Bland-Altman plots were generated to assess bias of the NARV for two cases: (i) the traditional EMG sensing technique and the sensor-embedded technique and (ii) sensor-embedded sensing on two different days (Figure 3.5). The difference between the NARV of the measurement techniques was plotted vs the mean NARV of the measurement techniques for all activities except "walking" since "walking" served as the normalization method. Due to the higher variability of the sensors located at the gluteus maximus and rectus abdominus, these sensors were excluded from this analysis so that loss of contact occurring at these sensor sites did not skew the results. This analysis showed a bias of 0.864 NARV (2SD limits of agreement = $\pm$ 9.032) towards the traditional EMG for Figure 3.5a and a bias of 0.017 NARV (2SD limits of agreement = $\pm$ 4.496) towards the sensor-embedded EMG collected on the second day for Figure 3.5b. The cone-shaped distribution of the data suggests that as the activation level increases (e.g., the mean of measurement increases) the difference between the techniques also increases.

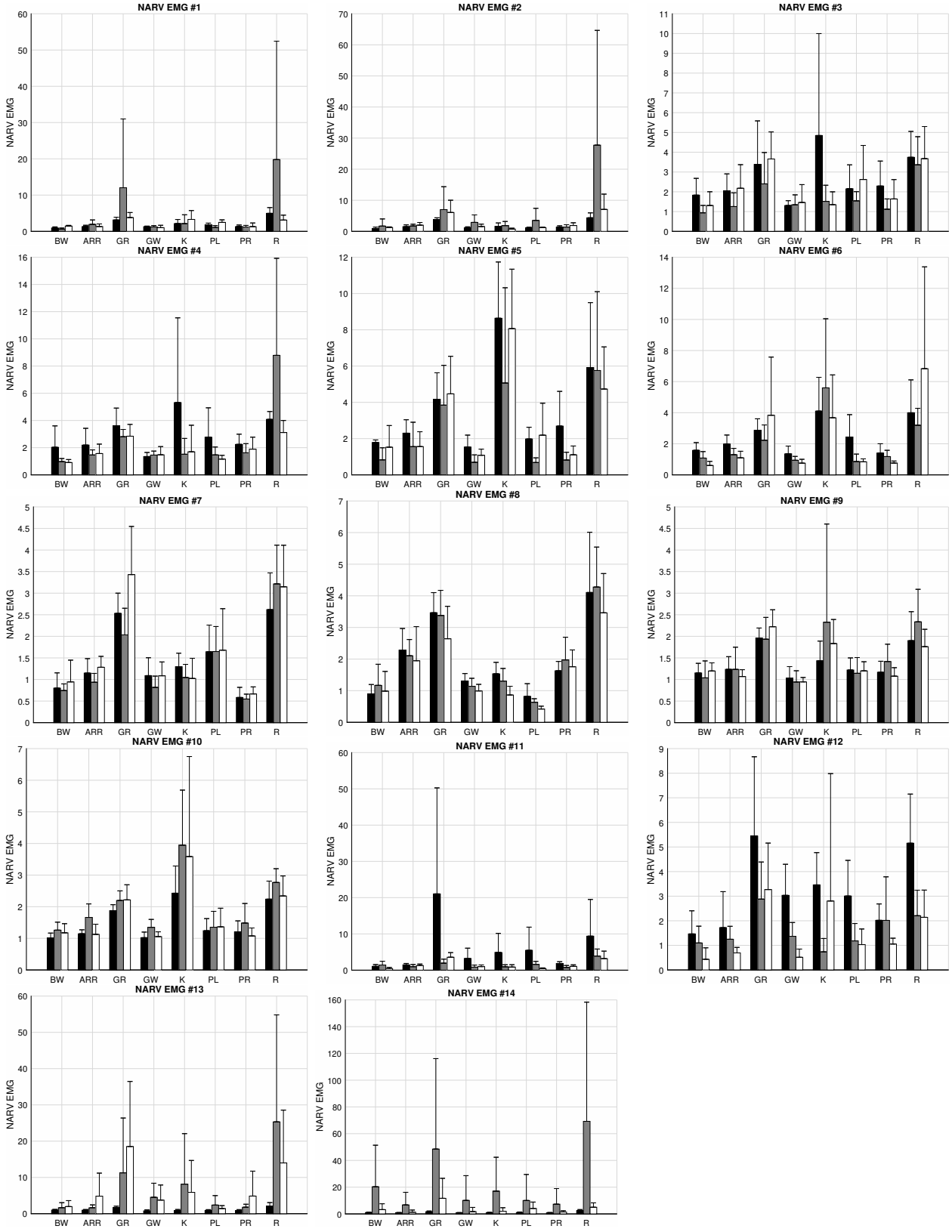


Figure 3.3: Normalized average rectified value (NARV) EMG for the nine actions performed during testing. Black - traditional electrode, Gray - suit collected on same day, White - suit collected on different day.

Even though the coefficients of variation were high with the textile embedded sensors, the motion classification was highly accurate for training and testing across all three subjects. The classification algorithm was more accurate using the traditional electrodes (accuracy of 92.7%, 87.3%, and 79.9% for Subjects #1, 2, and 3 respectively) compared to the textile electrodes (accuracy of 90.4%, 80.4%, and 79.41% for Subjects #1, 2, and 3, respectively) as shown in Figure 3.6-3.7.

The accuracy of the algorithm substantially decreased when training with traditional electrodes and testing with textile-embedded electrodes to 54.6%, 18.6%, and 19.2% for Subject #1, 2, and 3 respectively. Accuracy decreased further when training with textile electrode data and testing on traditional data to 49.6%, 13.2%, and 29.5% for Subject #1, 2 and 3 respectively. For the latter case, the algorithm predicted only one or two actions accurately for the majority of the time.

Additional training and testing procedures were conducting using data from different participants, such as training using traditional electrode data from Subject #1 and testing using traditional electrode data from Subject #2. The results of the different test configurations are summarized in Table 3.2 with the diagonal of the table's values being the results of training and testing with the same subject's data repeated from Figures 3.6-3.7. These results suggest that cross-participant usage of training and testing data results in decreased accuracy of the classification algorithm. Due to the low accuracy of the motion classification algorithm of cross measurement techniques (e.g., training with sensor-embedded textile and testing with traditional), cross participant cross measurement technique classifications were not run resulting in the blank spaces in Table 3.2.

All subjects were brought back for a second round of testing using the sensor-embedded garment. The set of data from the first collection was used to train the classification algorithm and the set from the second round was used for testing resulting in an accuracy of 36.8%, 22.7%, and 35.3% for Subjects #1, 2, and 3 respectively (Figure 3.8).

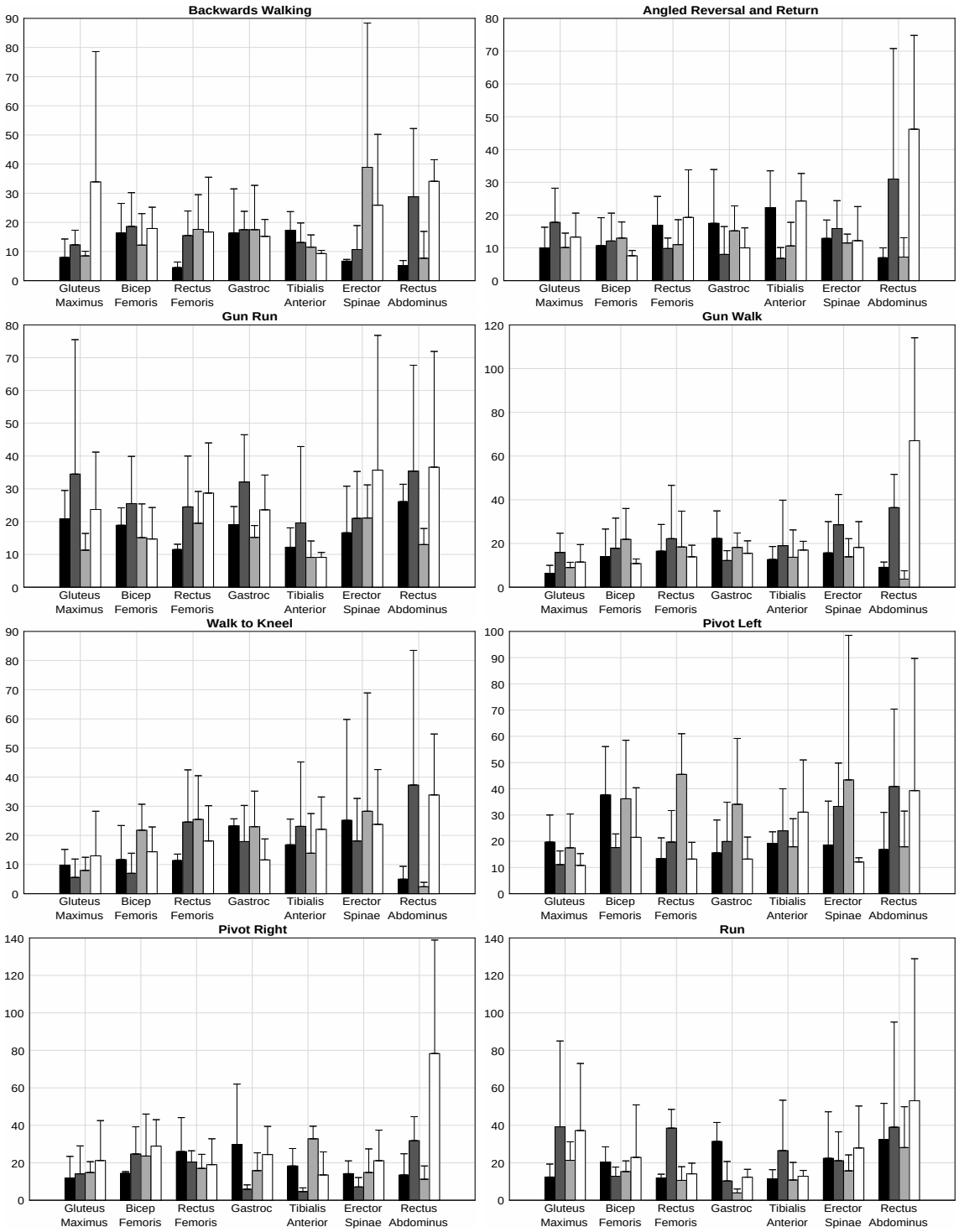
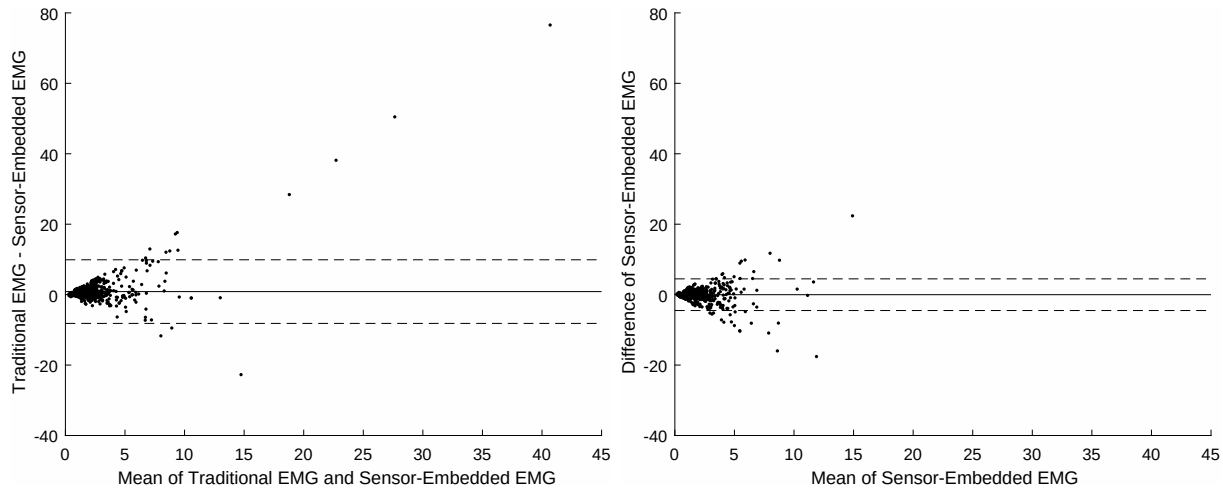


Figure 3.4: Mean percent coefficient of variation (%CV) for each muscle for each activity (SD) for traditional measurement technique and sensor embedded textile technique (Black - traditional on left side of the body, Dark Gray - textile on left side of the body, Light Gray - traditional on right side of the body, White - textile on the right side of the body.



(a) Traditional - Sensor-Embedded (b) Sensor-Embedded on Different Days

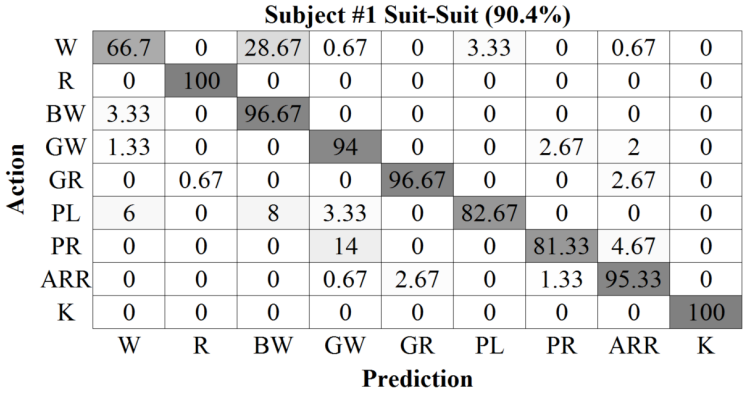
Figure 3.5: Bland-Altman plots for (a) traditional EMG compared to sensor-embedded EMG and (b) sensor-embedded EMG compared across two days.

Table 3.2: Motion Classification Accuracy for Cross Participant Tests (#_ indicates which subject used, T = Traditional electrode, S1 = Suit Electrode collected same day).

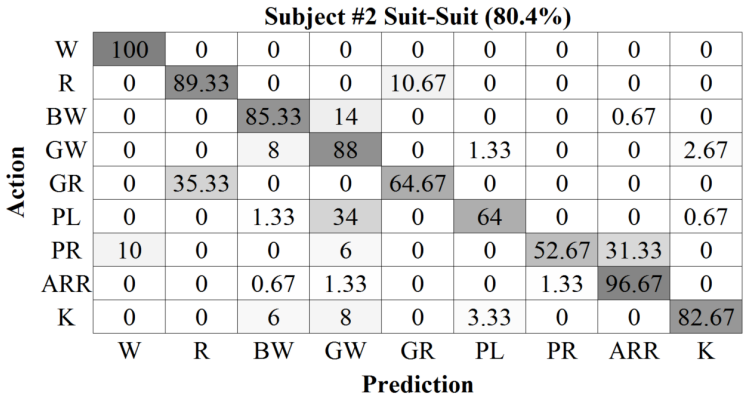
		Testing Set					
		#1 T	#1 S1	#2 T	#2 S1	#3 T	#3 S1
Training Set	#1 T	92.7	54.6	32.8		36.0	
	#1 S1	49.6	90.4		22.4		25.6
	#2 T	27.4		87.3	18.6	26.1	
	#2 S1		15.6	13.2	80.4		11.6
	#3 T	32.9		26.0		79.9	19.2
	#3 S1		28.0		26.9	29.5	79.4

3.10 Discussion

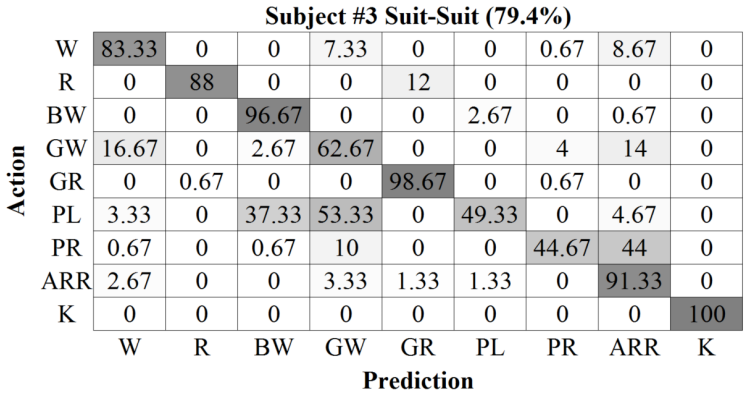
The goal of this study was to evaluate the efficacy of an EMG-embedded textile garment for the use of motion classification based on myoelectric activity using commercially available EMG sensors and garments. Previously developed textile EMG sensors have been used to capture the myoelectric activity of a muscle group, such as the hamstrings, whereas the textile presented here aims to capture the activity of a specific muscle using commercial, off-the-shelf products. This study has shown that the NARV collected by the suit was comparable to that collected by the traditional electrode set-up for certain muscles being measured (biceps femoris, rectus femoris, tibilias anterior, and gastronemius). As expected, the signals



(a) Subject #1



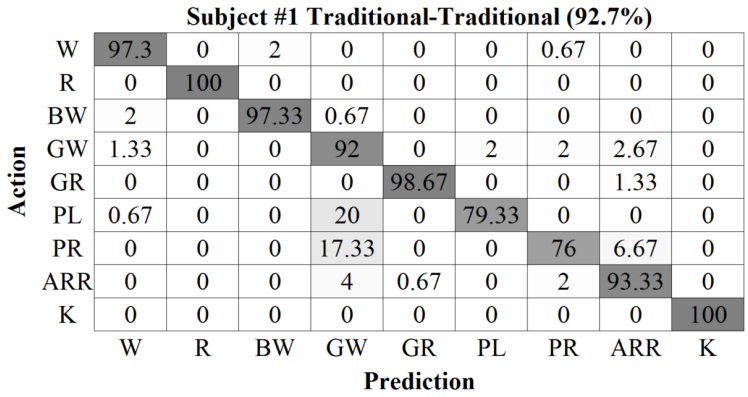
(b) Subject #2



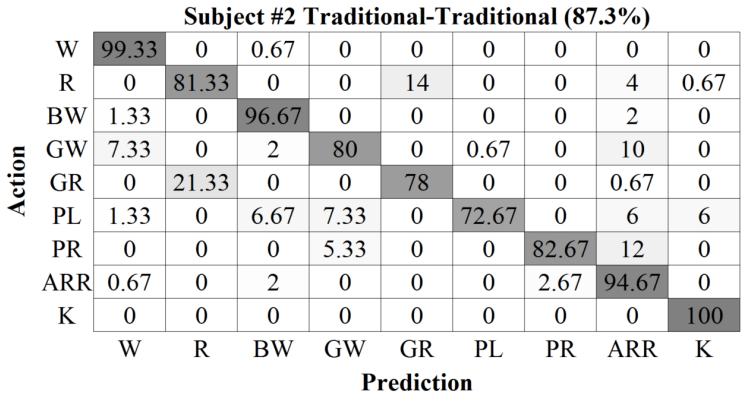
(c) Subject #3

Figure 3.6: Confusion matrix showing motion classification accuracy (%) using sensor embedded textile electrodes for training and testing.

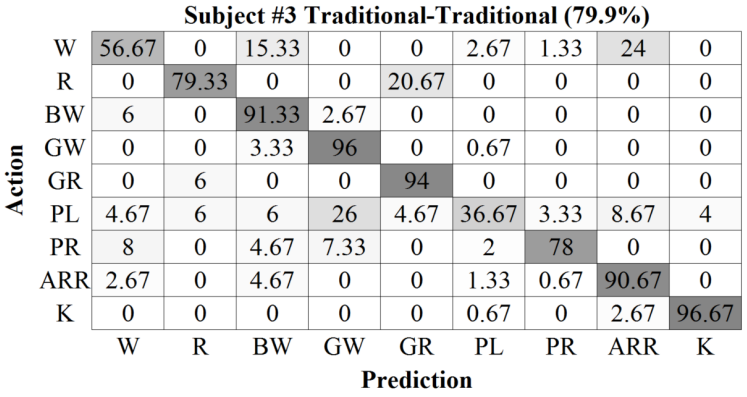
were the most comparable for activities involving lower activation levels such as walking and kneeling. For higher activation actions, such as run, the difference in the magnitudes of the NARV were larger and the percent coefficient of variation was substantially higher



(a) Subject #1



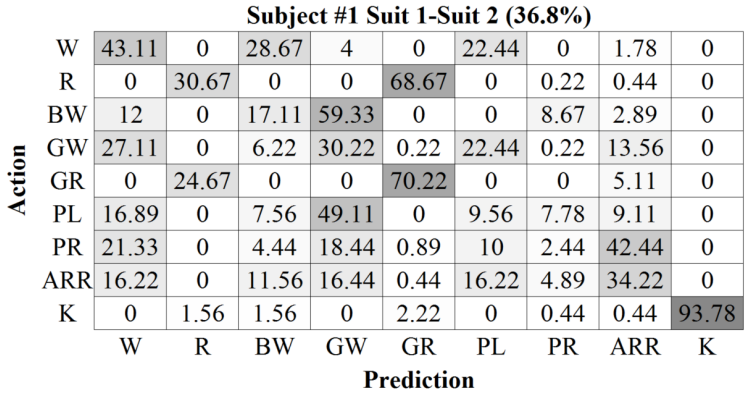
(b) Subject #2



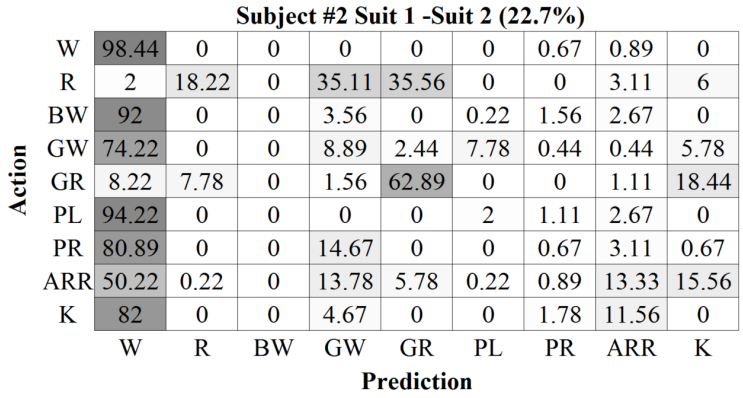
(c) Subject #3

Figure 3.7: Confusion matrix showing motion classification accuracy (%) using traditional electrodes for training and testing.

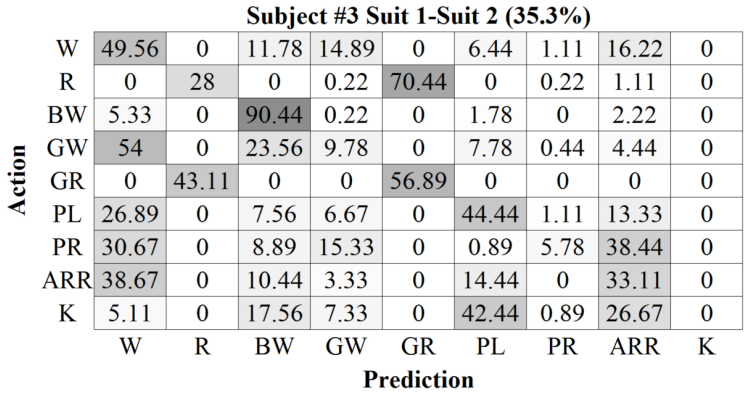
for the sensor-embedded textile electrodes. This is also illustrated in the Bland-Altman plots as the difference between the measurement techniques increases as the mean increases resulting in the fan-like shape (Figure 3.5). This may be attributed to the higher potential



(a) Subject #1



(b) Subject #2



(c) Subject #3

Figure 3.8: Confusion matrix showing motion classification accuracy (%) using sensor-embedded garment data collected on different days for training and testing.

for movement of the sensor relative to the skin causing motion artifacts to be present in the signal along with poor connection between the sensor and skin. The gluteus maximus, rectus abdominus, and erector spinae showed the most variability among the monitored muscles.

The rectus abdominus and erector spinae sensors are located in the shirt component of the textile suit. The suit was custom made for one of the subjects (Subject #1), but used by two others to test its performance on subjects of similar size. Due to different physiological aspects of the subjects, the shirt was looser in the areas of the rectus abdominus and erector spinae for the subjects for which the suit was not custom fitted. This led to decreased sensor contact, especially during higher velocity activities, likely contributing to an increase in signal variability. This indicates the need for properly fitted textiles for use of this technology.

Overall, the motion classification algorithm was highly accurate when training and testing using the same type of electrode for the same subject. There were misclassifications when using textile and traditional electrodes, but some of these misclassifications may be misleading. For example, in Figure 3.6, 35.33% of the "gun run" actions were classified as "run" predictions for Subject #2. This is a misclassification, but only a minor distinction, holding a PVC pipe while performing the running action in "gun run" trials, was present. Since the majority of the EMG sensors were placed on the lower extremities, it is likely that the EMG data during "run" is very similar to that of "gun run" leading to a misclassification. A similar misclassification occurs between the "walk" and "gun walk" trials. Again, the only difference between these two actions was carrying a PVC pipe in the "gun walk" action. This paper focuses on macro actions of the operator (motions that involve movement of multiple joints e.g., walking), but some of the actions performed by the subjects are combinations of other bulk motions. For example, the "angled reversal and return" motion is a step forward, slight pivot to the subject's left side, two steps back, and then a walk forward. Since the action at any one point in the "angled reversal and return" could demonstrate similar characteristics of one of its components over a 50 ms window used in testing the motion classification, this motion could easily be misclassified as one of its components. This is illustrated in Figure 3.7 for Subject #3 where the classification accuracy for "angled reversal and return" was 90.7%, but the motion was classified as "walking" (2.7%), "backwards walking" (4.7%) and "pivot left" (1.3%). Even though these classifications were stated as incorrect, the algorithm could

have been correctly classifying what the subject was doing for the 50 ms during the testing sample. This may indicate a need to classify smaller motions that can be combined to form an overall bulk motion, such as classifying knee flexion/extension, hip flexion/extension, hip abduction/adduction, and ankle dorsiflexion/plantar flexion combined into a walking motion. This could potentially eliminate possible misclassifications between "angled reversal and return" and walking/backwards/pivot by simply classifying the motion of certain joints.

Cross-participant motion classification training and testing tests such as training using textile electrodes from Subject #1 and testing using textile electrodes from Subject #2 were also performed. The overall accuracy of the motion classification algorithm for each of these test cases was lower compared to the test cases using the same subject's data for training and testing (Table 3.2). This highlights the importance of custom calibration data for each intended user as the algorithm as currently designed is not transferable to subjects with a lack of training data, but the physical suit developed in this study was transferable to other people of similar size to the reference subject. Inter-day testing and training produced poor motion classification accuracy (Figure 3.8). Misclassifications between similar activities such as "run" and "gun run" produce an even greater decrease in accuracy. This also highlights the need for motion classification at the joint level rather than the bulk motion level. The addition of more EMG sensors located on the upper extremities would also likely increase the overall accuracy and help distinguish between actions like "run" and "gun run".

The limitations of the study include the small sample size of participants used for testing due to the exploratory nature of the study. The classification algorithm was also limited to the nine actions listed in the Methods section that are specific to combat. Many of those actions are low velocity and are similar to that of walking. The difference between the NARV of the sensor embedded suit and traditional set-up grew as the velocity of the action increased. This could pose a potential problem if additional higher velocity actions were included in the classification algorithm. The algorithm is also limited to only working for motions that have training data.

Ongoing work involves exploring implementing a more flexible sensor-textile clip (rather than a PLA clip) that would allow the suit to be more durable and conform better to the user. In addition, more compression in the suit would decrease the amount of variability in the sensors resulting from lack of contact and movement between the sensor and skin. Additional sensors will be added to the upper body in order to distinguish between certain actions in addition to continuous improvements of the motion classification algorithm accuracy. This will also allow the motions classified in the algorithm to be expanded to include upper body motions. The Delsys sensors also contain IMU data. However, the focus of this paper was the integration of a EMG sensor into a compression garment for the purposes of motion classification so the IMU data was not included in this analysis, but the addition of IMU data in the motion classification algorithm would likely increase the accuracy of the algorithm. Future work will also focus on improving classification accuracy when training and testing data are collected on different days.

In conclusion, the EMG embedded textile developed in this study utilized a commercially available EMG sensor attached to a commercially available compression garment via a 3D printed sensor-textile clip. The NARV of the EMG signal collected via the sensor-embedded garment was shown to be comparable to traditional EMG placement for the biceps femoris, rectus femoris, tibilias anterior, and gastronemius. The gluteus maximus, erector spinae, and rectus abdominus were not comparable for the two sensing types, likely due to movement artifacts and loss of contact with the sensing area. The variability and difference in NARV between the sensor-embedded garment and traditional sensing techniques were larger for higher velocity motions. However, the sensor-embedded garment was also shown to produce accurate motion classification results (mean accuracy = 83.4%; SD = 6.08%) compared to traditional EMG placement (mean accuracy = 86.6%; SD = 6.43%) for data tested on the same day as data collected for training over various actions determined to be operationally important to the development of exoskeletons for soldiers.

Chapter 4

Vibratory Response Characteristics of High Frequency Shape Memory Alloy Actuators

The focus of this chapter is the development of a high frequency and high deformation shape memory alloy actuator using a bimorph configuration. This actuator can be applied to robots that mimic the motion of biological creatures.

4.1 Actuator Design

A SMA actuator was designed and manufactured using a bimorph structure in order to achieve high frequency motion. The active layers of the actuator consist of 37.5 μm SMA wire (Flexinol, Dynalloy, Irvine, CA), which is adhered in a rectangular layer of 3D printed thermoplastic polyurethane (TPU, Polymaker, Savannah USA). The SMA wires were routed along the length on the top and bottom of the TPU layer and electrically connected at one end as shown in Figures 4.1-4.2. In order to achieve low power consumption, the actuator design consisted of one wire routed from the base to the tip and back to the base, which allows all electrical connections to be placed on the base of the actuator, leaving the tip free of any additional mass from crimping and wiring connections that could affect the motion. The SMA wires were held in place by a layer of room-temperature-vulcanizing silicone (RTV).

The top and bottom SMA wires were electrically isolated due to the geometry of the bimorph structure, allowing for independent heating/contraction of each side. TPU and RTV silicone was selected for this actuator design since both have an operational temperature encompassing the operational temperature of the SMA wire (20-100°C). TPU and RTV also allow the actuator to be extremely flexible and durable compared to a SMA bimorph

Figure 4.1: Bimorph SMA actuator with TPU substrate and RTV silicone coating (CAD Model - Top View)

Figure 4.2: Bimorph SMA actuator with TPU substrate and RTV silicone coating (CAD Model - Side View)

actuator with a hard plastic passive layer, such as acrylonitrile butadiene styrene (ABS) or polylactic acid (PLA).

As the goal of this study was the evaluation of the dynamic performance of a SMA bimorph actuator while trying to minimize power consumption and maintenance, the materials were not optimized for heat transfer properties. The TPU substrate had a width of 6 mm and a thickness of 0.4 mm. The width of 6 mm was used for manufacturability of the actuator. The displacement of unimorph and bimorph actuators are functions of the strain of the active layer and the distance of the midline of the active layer to the midline of the passive layer [31]. As such, thickness of the TPU passive layer was minimized to achieve high displacement.

Actuators of various lengths (20, 25, 30, 35, and 40 mm) were tested at various peak current inputs (110, 120, 130, and 140 mA) to determine the natural frequency and displacement characteristics of the proposed SMA actuators. Figures 4.3-4.4 show one of the manufactured actuators used in this study. The actuators were colored black in order to aid in point tracking during testing.

4.2 Electrical Excitation

Oscillatory motion of the actuator was achieved by alternating excitation of the two active SMA layers by a sine wave current (Figure 4.5). A sine wave current input was used

Figure 4.3: Bimorph SMA actuator with TPU substrate and RTV silicone coating and motion tracking markers (Top View)

Figure 4.4: Bimorph SMA actuator with TPU substrate and RTV silicone coating and motion tracking markers (Side View)

in order to reduce the effects of a sudden impulse on the system when the current was redirected between the SMA layers.

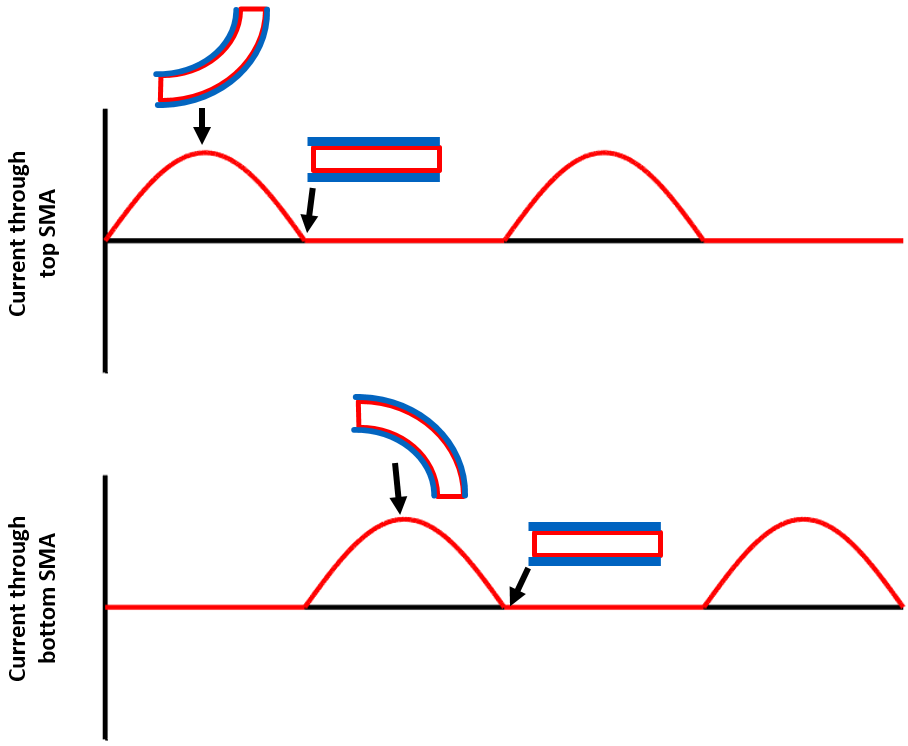


Figure 4.5: Sine wave voltage to oscillate actuator

The amplitude of the current was varied in order to determine the actuator performance and dynamic characteristics at different current inputs. A microcontroller was used to control the current and frequency of oscillations via an H-bridge circuit powered by a 0-30 Vdc power supply (BK Precision 1671A). High frequency motion results from the ability of the bimorph to be alternatively activated in opposing directions, which negates the necessity of cool down time that would otherwise be present for the actuator to return to the starting position. During actuation, one side of the bimorph is heated while the other side experiences no current input for a specified time. Then, the current is redirected to the other side of the bimorph actuator and heats the other SMA wire resulting in an oscillatory response. Experimental testing was conducted by first actuating at a frequency below the natural frequency, and then quasistatically increasing the frequency until it was above the natural frequency, as determined by previous trials. Since SMA wires can be damaged due to overheating resulting in a decrease in strain, the actuators were not tested at frequencies significantly lower than their natural frequency where overheating would be more likely.

The actuator was tested at 110, 120, 130, and 140 mA. These current values indicate the amplitude of the sine wave current input used to electrically excite the actuator and will be referred to as peak current input throughout the rest of the paper.

4.3 Equipment

An Edgertronic SC2+ camera (Sanstreak Corp., San Jose, CA) was used to collect kinematic data at 1000 frames per second by circular markers at the base (non-moving) and tip (moving) to track the movement of the actuator. The SMA actuator is realized as a cantilever, with one end fixed and the other end free. MATLAB was used for image processing to calculate the tip displacement of the actuator with respect to its initial position, allowing for the determination of the displacement of the actuator at the corresponding operating frequency. In addition to computer vision motion tracking, a slow-speed infrared camera was used to monitor the temperature of the actuator during actuation. The thermal camera

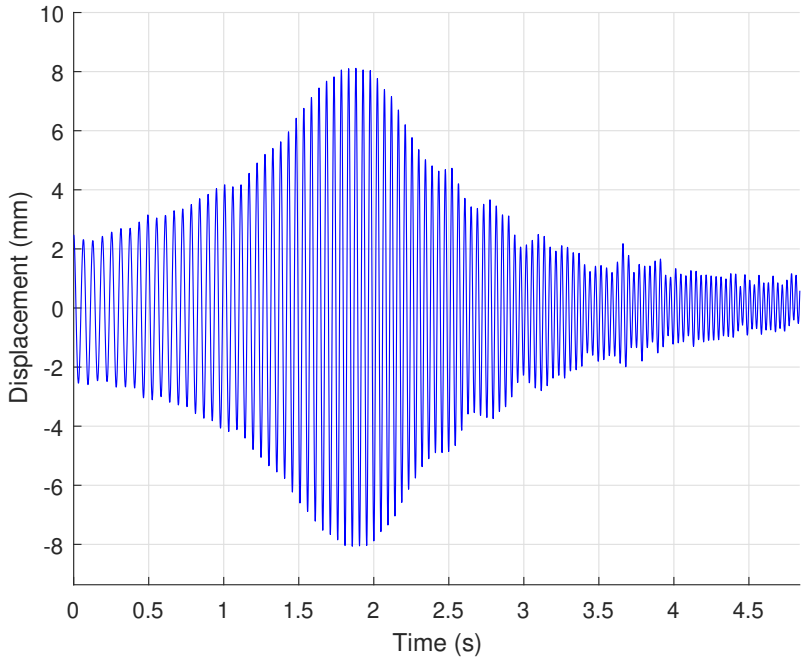


Figure 4.6: Temporal response of a 30 mm actuator with a 120 mA peak current frequency sweep.

was mounted in the plane of the actuator, so that it recorded the temperature of one of the top sides shown in Figure 4.2. The camera was used to verify that the actuator returned to room temperature before each trial. This was done to ensure that the initial temperature was the same for each trial.

4.4 Dynamic Characteristics

For a 30 mm actuator, the displacement is plotted against time using the high speed photogrammetry data (Figure 4.6). It shows that the actuator does experience resonance, as there is a substantial increase in displacement at a particular frequency. This agrees with the findings of previous studies [35].

The temporal displacement data was transformed into the frequency domain to show average displacement at excitation frequency. This data was then used to curve fit a modal

frequency response function (Equation 4.1) in order to determine the approximate spring constant, damping factor, and natural frequency of the actuator [59].

$$| G(\Omega) | = \frac{1}{k\sqrt{(1-\Omega^2)^2 + (2\zeta\Omega)^2}} \tag{4.1}$$

A least-squares nonlinear curve fit was used in MATLAB to fit the experimental data to the modal equation. For example, the results of the curve fit for the displacement data shown in Figure 4.6 can be seen in Figure 4.7.

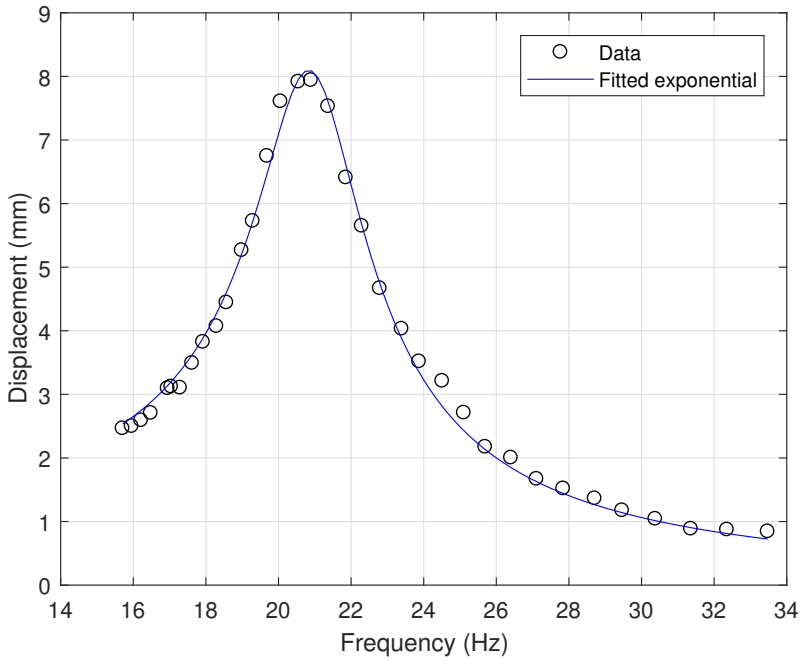


Figure 4.7: Response of 30 mm actuator with 120 mA current frequency sweep

The experimental data and estimated frequency response curve can be seen for each actuator length tested at each current input level of 110, 120, 130, and 140 mA in Figures 4.8-4.11. Displacement of the actuator was measured as the one dimensional displacement of the tip of the actuator from the neutral (un-actuated) initial position. The data shows that maximum displacement increases with increasing peak current input; in addition, the natural frequency increases with decreasing actuator length.

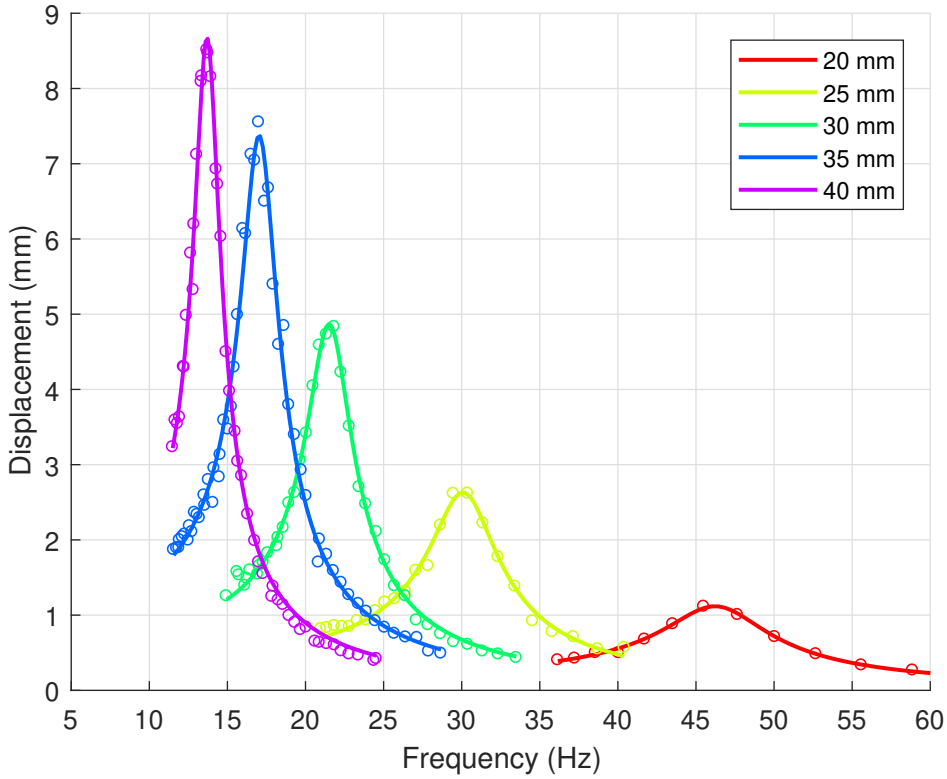


Figure 4.8: Displacement vs operating frequency for various actuator lengths at 110 mA excitation current

The peak current input and actuator length had an effect on both the natural frequency and displacement characteristics of the SMA bimorph actuator described in this paper as shown in Figure 4.12. For actuators of the same length at different peak current inputs, the natural frequency had an approximate linear change of 0.08, 0.06, 0.10, 0.10, and 0.02 Hz/mA for actuators of length 40, 35, 30, 25, and 20 mm, respectively. The displacement for the 25 mm actuator ranged from 1.12 to 2.54 mm for all peak current input levels. The relatively small difference in displacement resulted in the small difference between natural frequencies at various peak current input levels for this actuator length.

In the small-angle cantilever beam model (Equation 4.2), the natural frequency is inversely proportional to the cubed length. The actuator in this study showed similar characteristics in that the natural frequency is inversely proportional to the length raised to the -2.15, -2.24, -2.43, and -2.48 power for 110, 120, 130, and 140 mA, respectively.

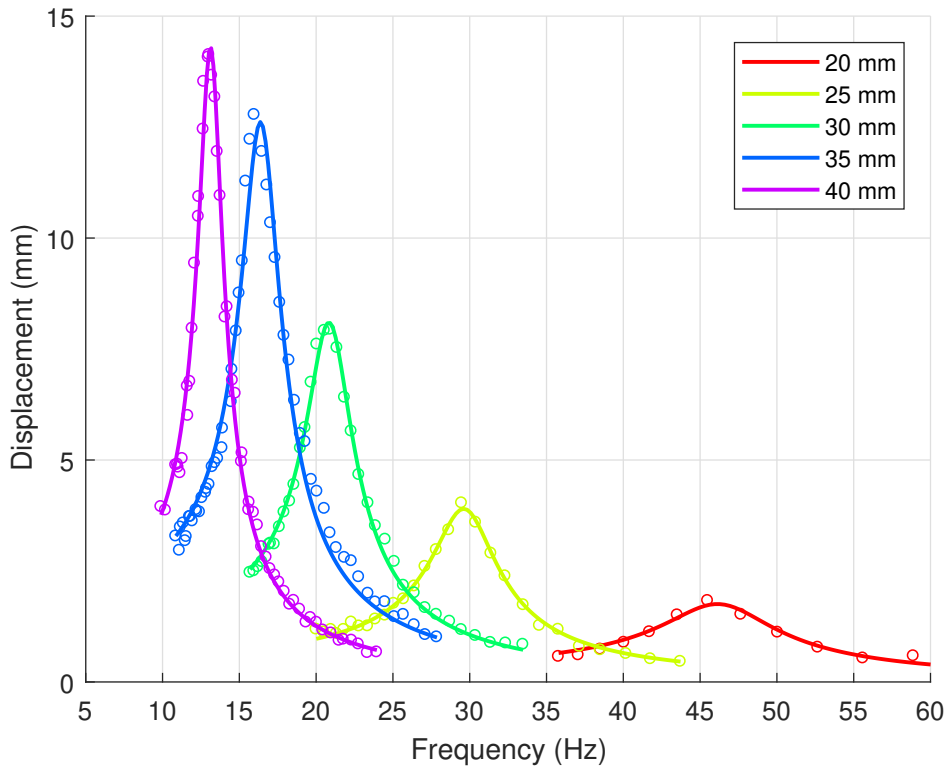


Figure 4.9: Displacement vs operating frequency for various actuator lengths at 120 mA excitation current

Maximum displacement increased proportionally with actuator length. Similar to the change in natural frequency at different peak current input levels, the maximum displacement changed linearly at 5.96, 4.93, 3.13, and 1.88 mm displacement/mm length (slope of the displacement versus actuator length in Figure 4.12). As the peak current was increased, the slope of the max displacement versus actuator length shown in Figure 4.12 increases by 0.14 mm/mA.

The estimated frequency function curve approximated the spring constant and damping factor for the system. The values can be seen in Figures 4.13-4.14. The damping factor increased with the cube of the peak current. The approximated spring constant decreased with increasing peak current. The amount of change depended on the length of the actuator. For the 20 mm actuator, the spring constant decreased from 6.28 N/m at 110 mA to 1.72 at 140 mA. The negative slope in approximated spring constant and damping ratio is likely caused by a combination of viscoelastic properties of the passive TPU layer and change in

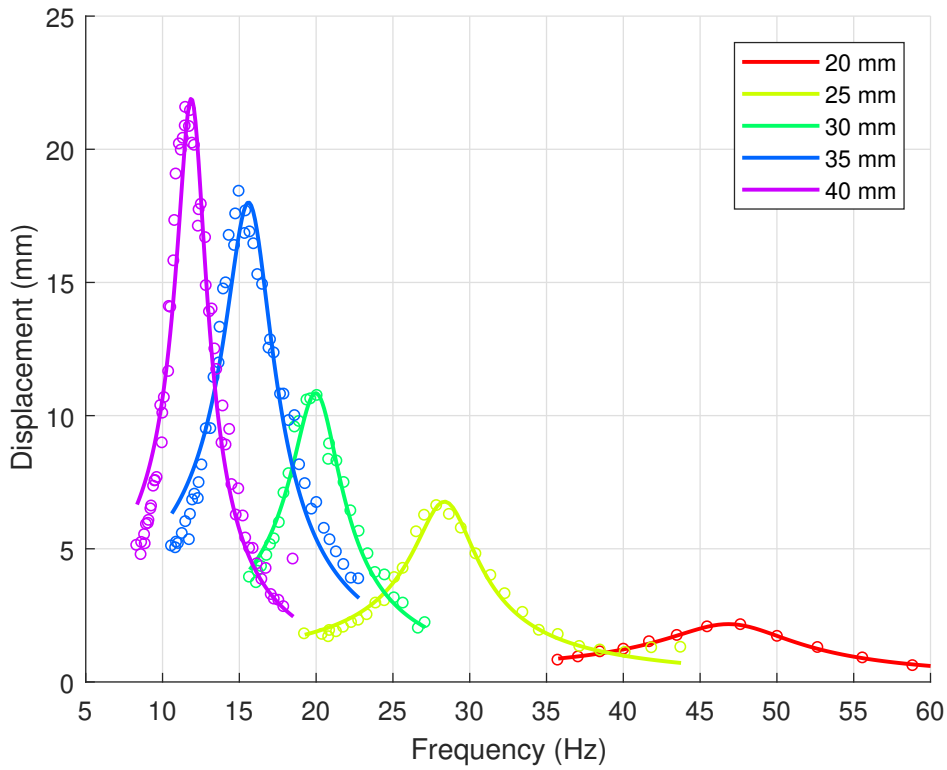


Figure 4.10: Displacement vs operating frequency for various actuator lengths at 130 mA excitation current

material properties due to the increase in temperature of the actuator since more current is being applied.

4.5 Modeling

Natural frequency modeling of cantilever beams is well studied and can be modeled by Equation 4.2 [59], assuming a cantilever beam with a rectangular cross section and homogeneous material. This model can potentially be used to predict the natural frequency of the actuator since the actuator designed in this study oscillates like a cantilever beam and the natural frequency of the actuator was shown to trend similarly to a cantilever beam as the length of the beam changes.

$$f_n = \frac{1}{2\pi}\sqrt{\frac{Ewh^3}{4ml^3}} \tag{4.2}$$

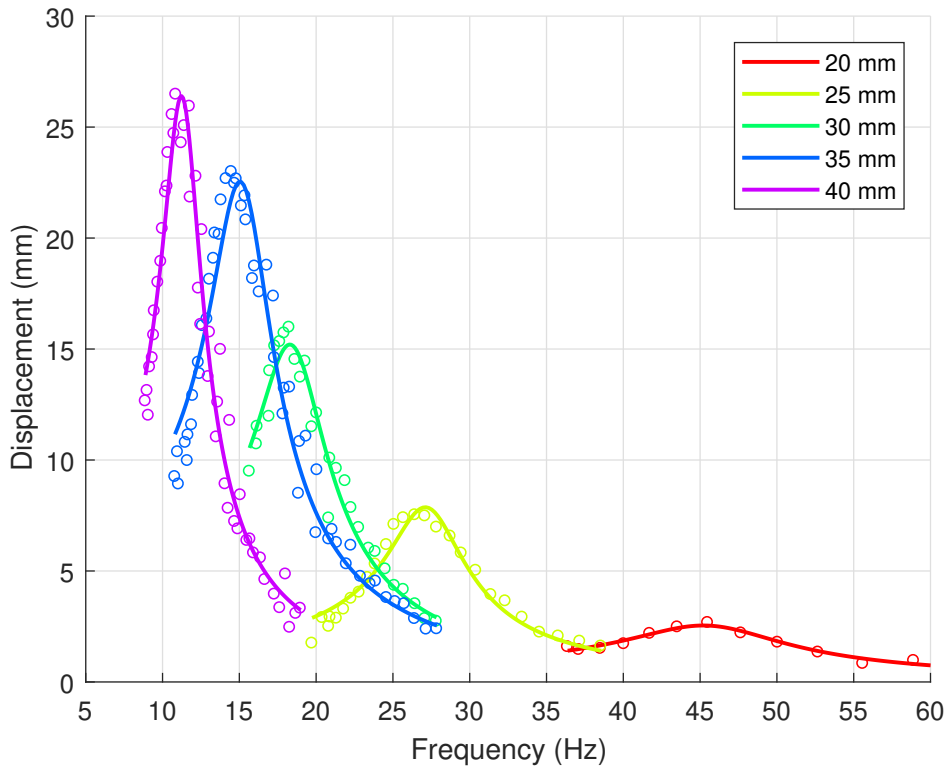


Figure 4.11: Displacement vs operating frequency for various actuator lengths at 140 mA excitation current

However, this simple cantilever beam does not accurately model this actuator since it is made up of 3 different materials (SMA, RTV, and TPU). In order to account for all the materials, the rule of mixtures was used to estimate an equivalent elastic modulus of the actuator (Equation 4.3). The equivalent elastic modulus was used in Equation 4.2 with the overall cross sectional dimensions and mass of the actuator.

$$E_{equiv} = E_{sma}\nu_{sma} + E_{tpu}\nu_{tpu} + E_{rtv}\nu_{rtv} \tag{4.3}$$

To compare with these previous two analysis, an equivalent cross sectional area, which allows the system to be modeled as one material while maintaining the stiffness of the original system, was also employed. This is done by scaling the widths of the materials based on the ratio of their elastic moduli compared to the elastic modulus of the "parent" material (the material into which the other materials are being "transformed"). This calculation was

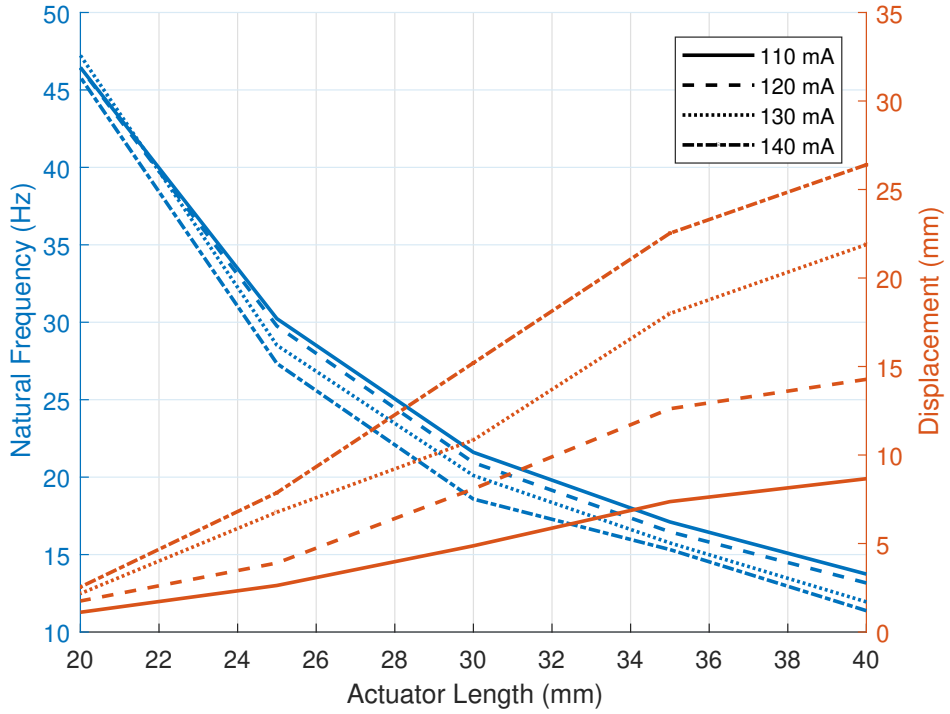


Figure 4.12: Natural frequency and max displacement of actuators at various lengths and current inputs.

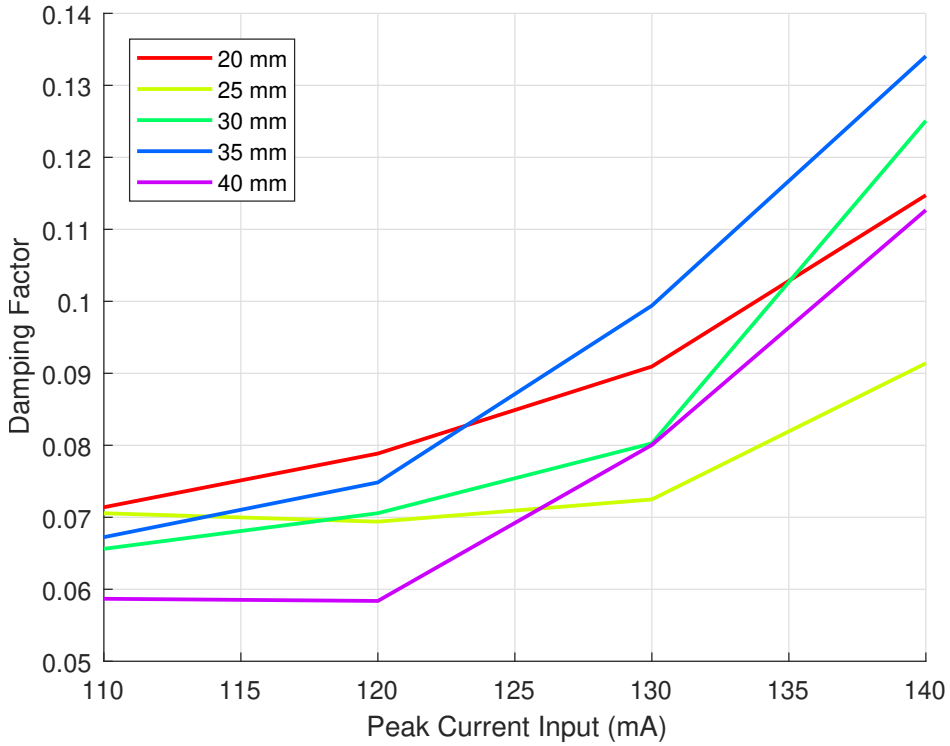


Figure 4.13: Damping factor of actuators at various lengths and current inputs.

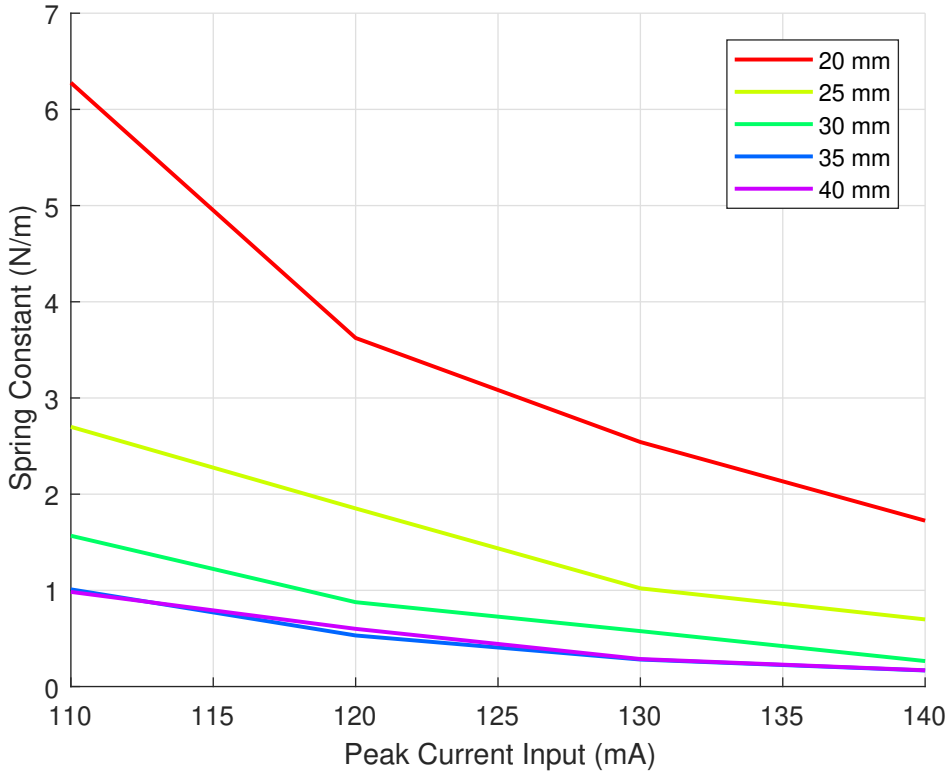


Figure 4.14: Spring constant of actuators at various lengths and current inputs.

done using TPU as the parent material. After the transformation, the moment of inertia was calculated and used to determine the spring constant and natural frequency of the system.

In order to verify and compare the various models, an ANSYS modal analysis was used to determine the natural frequency of the system based on the 3D model shown in Figures 4.1-4.2. Fixed supports were added to one end while the other remained free. The results of the standard cantilever beam equation, rules of mixtures, and equivalent area are shown in Figure 4.15 along with the experimental and simulation results.

The simulation was the most accurate with a percent difference of 5.6%. The equivalent area and rule of mixtures had a percent difference of 6.5% and 21.1% respectively.

4.6 Power Consumption

In order for untethered, soft actuated robots to become increasingly more useful and available, they need to have a power consumption that is capable of being generated by a

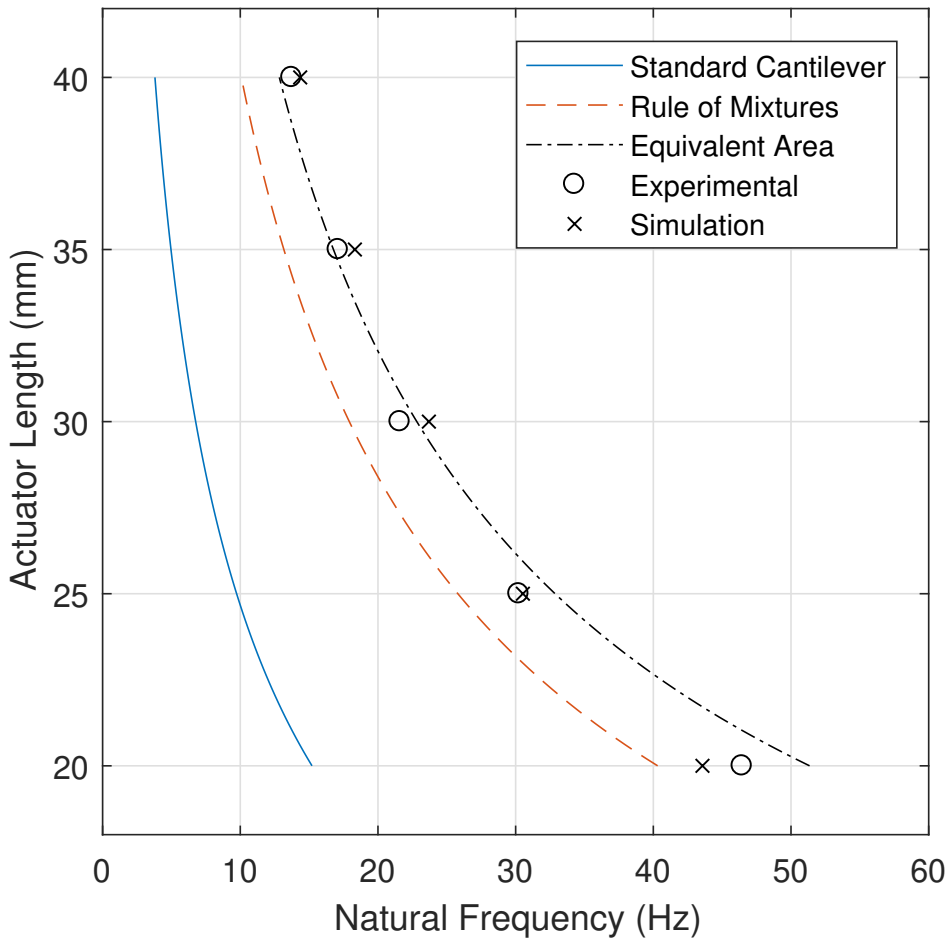


Figure 4.15: Natural frequency system identification using rule of mixtures and cantilever beam equations compared to experimental (at 110 mA peak current input) and ANSYS results.

battery pack. Since a sine wave was used for the current input to the system, the root mean square value was used to calculate the power consumption of the actuator using the following equations:

$$I_{rms} = \frac{I_{peak}}{\sqrt{2}} \tag{4.4}$$

$$P = I_{rms}^2 \cdot R_{sma} \tag{4.5}$$

Determining the resistance of the SMA wire is challenging due to the fact that the material properties of the SMA change as it is transformed from martensite to austenite and back to martensite. The resistivity of Flexinol SMA wire is 80 $\mu\Omega cm$ in the martensite phase and 100 $\mu\Omega cm$ in the austenite phase. Various mixture models have been proposed in order to

model the material properties while the material is transforming phases, but, for simplicity, a constant resistance was assumed and the resistance of the SMA in its resting state was measured and used for power consumption calculations (Table 4.1).

Table 4.1: Resistance (Ω) for the actuator lengths tested.

Length (mm)	Resistance (Ω)
45	79.9
40	65.2
35	56.1
30	45.7
25	38.1

Using Equations 4.4-4.5, the power consumption for the actuators tested was calculated and shown in Table 4.2. The power consumption ranged from 0.23 watts for the shortest (20 mm) actuator at the lowest current tested (110 mA) to 0.78 watts for the longest (40 mm) actuator at the highest current tested (140 mA).

Table 4.2: Power Consumption

Length	110 mA	120 mA	130 mA	140 mA
40 mm	0.48 W	0.58 W	0.68 W	0.78 W
35 mm	0.39 W	0.47 W	0.55 W	0.64 W
30 mm	0.34 W	0.40 W	0.47 W	0.55 W
25 mm	0.28 W	0.33 W	0.39 W	0.45 W
20 mm	0.23 W	0.27 W	0.32 W	0.37 W

4.7 Discussion

The actuator designed here used a simple rectangular piece of thermoplastic polyurethane and commercially available nickel-titanium shape memory alloy wires adhered together with

room-temperature-vulcanizing silicone into a bimorph configuration. The bimorph configuration allows for an increase in displacement and operating frequency compared to linear SMA actuators. For a 40 mm linear actuator, the typical 4% of displacement would result in 1.6 mm of movement. However, the 40 mm bimorph actuator presented in this paper was shown to achieve one-sided displacements of 26.4 mm from its initially straight position resulting in a total deformation of 52.8 mm (132%). Previous studies have shown a 40 mm SMA bimorph actuator to be capable of ∼80 mm displacement (∼200%), but at a considerably higher power consumption (>50 Watts) [35].

Along with displacement, the operating frequency of a SMA bimorph actuator is higher than traditional linear SMA actuators. Since SMAs rely on a temperature change to achieve their length change, their operating cycle is typically limited to how fast they are passively cooled. SMA actuators that rely on free convection for cooling are commonly reported with maximum operating frequencies of < 3 Hz. The bimorph SMA actuator that consists of antagonistic pairs, allow the opposing SMA layers to "pull" the actuated SMA layer out of its strained state. Since this configuration does not rely on cooling, it can achieve much higher operating frequencies. The 25 mm actuator tested in this study showed an operating frequency of 27.3 Hz while maintaining a displacement of 15.7 mm resulting in a 62.8% length change of the actuator. Both of these values are significantly larger than traditional SMA linear actuators.

As the current input was varied in the actuator, the displacement and natural frequency changed. This was likely due to the increase in strain rate of the actuator. When a higher amount of power is input into the actuator, the SMA wire transforms from martensite to austenite at a faster rate. This causes an increase in the bending velocity of the actuator, which causes an increase in momentum as the actuator bends. This momentum contributes to the increase in displacement as the peak current input is increased. TPU, which is a viscoelastic material, was used as the passive layer in the bimorph since it was able to withstand the high temperatures of the SMAs during their phase transformation. Since it

is a viscoelastic material, its properties are strain rate dependent. Therefore, when a higher current was passed through the actuator, a higher strain resulted in a decrease in the natural frequency of the actuator. This can be seen in Figure 4.12. For the 20 mm actuator, the one-sided displacements are relatively the same (max displacement of 1.12, 1.75, 2.17, and 2.54 mm for 110, 120, 130, and 140 mA, respectively). Similar displacement profiles mean that the actuators have relatively similar strain rates at all peak current inputs. This results in the 20 mm actuator having similar natural frequencies at all power levels (46.4, 46.4, 47.2, and 45.8 Hz for 110, 120, 130, and 140 mA, respectively). The contrasting case is for the 40 mm actuator, where there is a significant change in displacement profiles (max displacement of 8.66, 14.28, 21.89, and 26.40 mm for 110, 120, 130, and 140 mA, respectively). The change in strain rate between peak current levels is higher resulting in a more significant change in natural frequency (13.7, 13.2, 11.9, and 11.4 Hz for 110, 120, 130, and 140 mA, respectively).

The models presented in this paper (rule of mixtures and equivalent area) were shown to produce accurate results of the natural frequency at various actuator lengths (Figure 4.15). The error in the models is likely due to differences in the physical actuators compared to the 3D model used for the calculations. For example, the model calculations assume a constant cross-sectional area along the length of the actuator and that the actuator is perfectly symmetric about the mid-plane of the passive TPU layer. However, when the actuator is made, variations in the thickness of the RTV layer could result in nonuniform cross-sectional areas and alter the results of the experimental data.

One of the most important characteristics of soft actuators and their advancement is power consumption. Assuming a constant resistance and using the RMS value for the sine wave current input, the power consumption ranged from 0.23 watts to 0.78 watts for the actuator design. The actuator design in this study used the smallest diameter SMAs that were available (37.5 μm) and the length of the SMA was kept as short as possible by routing the wire from the base to the tip and back to the base so that all electrical connections were located at the base. This keeps the mass as low as possible to allow the temperature

change to happen as quickly as possible while consuming a low amount of power. Keeping all electrical connections at one end also eliminates any effects on the actuator from electrical wires crimped on the moving end. The low power consumption of the actuator constructed in this study would allow for the use of this technology in unthethered applications. A single actuator could be powered by a battery typically used in embedded systems.

Along with high displacement, high operating frequency, and low power consumption, the materials used to construct the actuator are inexpensive. SMA wires trained to shorten in length are commercially available along with the RTV silicone to adhere the SMA wires to the TPU substrate (a popular filament used in 3D printers). Overall, the 40 mm actuator proposed in this design costs approximately $2.50 to manufacture.

The force output of the actuators was not investigated in this current work as the focus was to develop a SMA actuator with high displacement and operating frequency characteristics with a low power consumption. The force output of a bending SMA actuator is less than traditional SMA linear actuators. However, since linear SMA actuators are $\sim$500 times stronger than human skeletal muscles, a decrease in force would likely still be comparable to human skeletal muscles along with other forms of soft actuators. It is also worth noting that only 37.5 μm diameter SMAs were used in this study in order to allow the wire to transform phases as quickly as possible. A larger diameter wire would likely increase the force capabilities of the actuator, but have an adverse effect on the operation frequency. The passive materials used in this actuator design were not optimized for heat transfer in this study. A more detailed analysis on how the material properties affect the actuator performance is necessary and could possibly improve the performance of the design. In addition to materials used, alternate cross-sectional areas of the passive TPU layer were not investigated. Previous studies have shown that the displacement of the unimorph and bimorph actuators are functions of both the strain of the active material and the distance of the active layer midline from the midline of the passive layer. Therefore, the thickness (0.4 mm used in this

study) could be reduced in order to achieve more displacement. However, this would result in a decrease in natural frequency of the actuator based on Equation 4.2.

The dynamic characteristics of the actuator presented in this paper are functions of material properties, geometry, and current input. While this adds to the complexity of analysis, it also allows for a wide range of customizable behavior based on the specific application of the bimorph actuator. Future work will look at determining the force capabilities of the actuator along with how force production affects the displacement and frequency responses of the actuator.

Chapter 5

Conclusions and Future Work

The literature presents various methods of exoskeleton designs and use of electromyography to dictate the motion along with various biomimetic robots that use soft actuators in order to produce biologically inspired motion. The electromyography sensor embedded suit developed showed to produce accurate motion classification results from a bank of actions using only the electromyography sensor data. This allows for minimal technical knowledge for the application of EMG sensors with an exoskeleton. For biomimetic actuators, a bimorph shape memory alloy actuator was constructed and shown to have an increase in operational frequency and displacement compared to traditional linear shape memory alloy actuators. This actuator also consumed minimal amounts of power due to the small diameter of the SMA wire used and limited length. This actuator could potentially improve SMA biomimetic robots.

Future work will look at optimizing the bimorph SMA actuator design along with quantifying the force output of the actuator. The object of the shape memory alloy research was to test the bimorph configuration at various lengths and observe the change in natural frequency, displacement, and other dynamic characteristics. As a result, force output was not yet quantified and tested, but is important for use in biomimetic applications. Additionally control of the oscillations and implementation of the actuator in biomimetic robots will be pursued.

Bibliography

[1] T. Yan, M. Cempini, C. M. Oddo, and N. Vitiello, "Review of assistive strategies in powered lower-limb orthoses and exoskeletons," Robotics and Autonomous Systems, vol. 64, pp. 120–136, 2015.

[2] L. M. Mooney, E. J. Rouse, and H. M. Herr, "Autonomous exoskeleton reduces metabolic cost of human walking," Journal of neuroengineering and rehabilitation, vol. 11, no. 1, p. 151, 2014.

[3] Z. Xu and E. Todorov, "Design of a highly biomimetic anthropomorphic robotic hand towards artificial limb regeneration," in Robotics and Automation (ICRA), 2016 IEEE International Conference on. IEEE, 2016, pp. 3485–3492.

[4] M. Bortole, A. Venkatakrishnan, F. Zhu, J. C. Moreno, G. E. Francisco, J. L. Pons, and J. L. Contreras-Vidal, "The h2 robotic exoskeleton for gait rehabilitation after stroke: early findings from a clinical study," Journal of neuroengineering and rehabilitation, vol. 12, no. 1, p. 54, 2015.

[5] A. M. Khan, D.-w. Yun, J.-S. Han, K. Shin, and C.-S. Han, "Upper extremity assist exoskeleton robot," in Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on. IEEE, 2014, pp. 892–898.

[6] A. B. Zoss, H. Kazerooni, and A. Chu, "Biomechanical design of the berkeley lower extremity exoskeleton (bleex)," IEEE/ASME Transactions on mechatronics, vol. 11, no. 2, pp. 128–138, 2006.

[7] T. Noda, T. Teramae, B. Ugurlu, and J. Morimoto, "Development of an upper limb exoskeleton powered via pneumatic electric hybrid actuators with bowden cable," in 2014 IEEE/RSJ International conference on intelligent robots and systems. IEEE, 2014, pp. 3573–3578.

[8] M. P. De Looze, T. Bosch, F. Krause, K. S. Stadler, and L. W. O'Sullivan, "Exoskeletons for industrial application and their potential effects on physical work load," Ergonomics, vol. 59, no. 5, pp. 671–681, 2016.

[9] G. Li, Y. Geng, D. Tao, and P. Zhou, "Performance of electromyography recorded using textile electrodes in classifying arm movements," in Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE. IEEE, 2011, pp. 4243–4246.

[10] A. B. Ajiboye and R. F. Weir, "A heuristic fuzzy logic approach to emg pattern recognition for multifunctional prosthesis control," IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 13, no. 3, pp. 280–291, 2005.

[11] J.-U. Chu, I. Moon, and M.-S. Mun, "A real-time emg pattern recognition system based on linear-nonlinear feature projection for a multifunction myoelectric hand," IEEE Transactions on biomedical engineering, vol. 53, no. 11, pp. 2232–2239, 2006.

[12] K. Englehart and B. Hudgins, "A robust, real-time control scheme for multifunction myoelectric control," IEEE transactions on biomedical engineering, vol. 50, no. 7, pp. 848–854, 2003.

[13] A. Young, T. Kuiken, and L. Hargrove, "Analysis of using emg and mechanical sensors to enhance intent recognition in powered lower limb prostheses," Journal of neural engineering, vol. 11, no. 5, p. 056021, 2014.

[14] H. Huang, T. A. Kuiken, R. D. Lipschutz et al., "A strategy for identifying locomotion modes using surface electromyography," IEEE Transactions on Biomedical Engineering, vol. 56, no. 1, pp. 65–73, 2009.

[15] M. T. Farrell and H. Herr, "A method to determine the optimal features for control of a powered lower-limb prostheses," in Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE. IEEE, 2011, pp. 6041–6046.

[16] M. Meng, Z. Luo, Q. She, and Y. Ma, "Automatic recognition of gait mode from emg signals of lower limb," in Industrial Mechatronics and Automation (ICIMA), 2010 2nd International Conference on, vol. 1. IEEE, 2010, pp. 282–285.

[17] H. Huang, F. Zhang, L. J. Hargrove, Z. Dou, D. R. Rogers, and K. B. Englehart, "Continuous locomotion-mode identification for prosthetic legs based on neuromuscular–mechanical fusion," IEEE Transactions on Biomedical Engineering, vol. 58, no. 10, pp. 2867–2875, 2011.

[18] Y. Huang, K. B. Englehart, B. Hudgins, and A. D. Chan, "A gaussian mixture model based classification scheme for myoelectric control of powered upper limb prostheses," IEEE Transactions on Biomedical Engineering, vol. 52, no. 11, pp. 1801–1811, 2005.

[19] H. Lee, K. Kim, and S. R. Oh, "Development of a wearable and dry semg electrode system for decoding of human hand configurations," in Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. IEEE, 2012, pp. 746–750.

[20] N. U. Ahamed, K. Sundaraj, R. B. Ahmad, M. Rahman, and M. A. Islam, "Analysis of right arm biceps brachii muscle activity with varying the electrode placement on three male age groups during isometric contractions using a wireless emg sensor," Procedia Engineering, vol. 41, pp. 61–67, 2012.

[21] T. W. Beck, T. J. Housh, J. T. Cramer, J. R. Stout, E. D. Ryan, T. J. Herda, P. B. Costa, and J. M. Defreitas, "Electrode placement over the innervation zone affects

the low-, not the high-frequency portion of the emg frequency spectrum," Journal of Electromyography and Kinesiology, vol. 19, no. 4, pp. 660–666, 2009.

[22] S. H. Roy, G. De Luca, M. Cheng, A. Johansson, L. D. Gilmore, and C. J. De Luca, "Electro-mechanical stability of surface emg sensors," Medical & biological engineering & computing, vol. 45, no. 5, pp. 447–457, 2007.

[23] C. Kendell, E. D. Lemaire, Y. Losier, A. Wilson, A. Chan, and B. Hudgins, "A novel approach to surface electromyography: an exploratory study of electrode-pair selection based on signal characteristics," Journal of neuroengineering and rehabilitation, vol. 9, no. 1, p. 24, 2012.

[24] S. L. Colyer and P. M. McGuigan, "Textile electrodes embedded in clothing: A practical alternative to traditional surface electromyography when assessing muscle excitation during functional movements," Journal of sports science & medicine, vol. 17, no. 1, p. 101, 2018.

[25] T. Finni, M. Hu, P. Kettunen, T. Vilavuo, and S. Cheng, "Measurement of emg activity with textile electrodes embedded into clothing," Physiological measurement, vol. 28, no. 11, p. 1405, 2007.

[26] S. M. Mirvakili and I. W. Hunter, "Artificial muscles: Mechanisms, applications, and challenges," Advanced Materials, vol. 30, no. 6, p. 1704407, 2018.

[27] I. W. Hunter and S. Lafontaine, "A comparison of muscle with artificial actuators," in Solid-State Sensor and Actuator Workshop, 1992. 5th Technical Digest., IEEE. IEEE, 1992, pp. 178–185.

[28] P. Brochu and Q. Pei, "Advances in dielectric elastomers for actuators and artificial muscles," Macromolecular rapid communications, vol. 31, no. 1, pp. 10–36, 2010.

[29] H.-I. Kim, M.-W. Han, S.-H. Song, and S.-H. Ahn, "Soft morphing hand driven by sma tendon wire," Composites Part B: Engineering, vol. 105, pp. 138–148, 2016.

[30] S. Kim, C. Laschi, and B. Trimmer, "Soft robotics: a bioinspired evolution in robotics," Trends in biotechnology, vol. 31, no. 5, pp. 287–294, 2013.

[31] A. Villanueva, C. Smith, and S. Priya, "A biomimetic robotic jellyfish (robojelly) actuated by shape memory alloy composite actuators," Bioinspiration & biomimetics, vol. 6, no. 3, p. 036004, 2011.

[32] A. Villanueva, K. Joshi, J. Blottman, and S. Priya, "A bio-inspired shape memory alloy composite (bismac) actuator," Smart Materials and Structures, vol. 19, no. 2, p. 025013, 2010.

[33] C. Smith, A. Villanueva, K. Joshi, Y. Tadesse, and S. Priya, "Working principle of bio-inspired shape memory alloy composite actuators," Smart Materials and Structures, vol. 20, no. 1, p. 012001, 2010.

[34] H.-J. Kim, S.-H. Song, and S.-H. Ahn, "A turtle-like swimming robot using a smart soft composite (ssc) structure," Smart Materials and Structures, vol. 22, no. 1, p. 014007, 2012.

[35] S.-H. Song, J.-Y. Lee, H. Rodrigue, I.-S. Choi, Y. J. Kang, and S.-H. Ahn, "35 hz shape memory alloy actuator with bending-twisting mode," Scientific reports, vol. 6, p. 21118, 2016.

[36] Z. Wang, G. Hang, Y. Wang, J. Li, and W. Du, "Embedded sma wire actuated biomimetic fin: a module for biomimetic underwater propulsion," Smart Materials and Structures, vol. 17, no. 2, p. 025039, 2008.

[37] K. J. Kim and M. Shahinpoor, "A novel method of manufacturing three-dimensional ionic polymer–metal composites (ipmcs) biomimetic sensors, actuators and artificial muscles," Polymer, vol. 43, no. 3, pp. 797–802, 2002.

[38] C. Jo, D. Pugal, I.-K. Oh, K. J. Kim, and K. Asaka, "Recent advances in ionic polymer–metal composite actuators and their modeling and applications," Progress in Polymer Science, vol. 38, no. 7, pp. 1037–1066, 2013.

[39] R. Pelrine, R. Kornbluh, Q. Pei, and J. Joseph, "High-speed electrically actuated elastomers with strain greater than 100%," Science, vol. 287, no. 5454, pp. 836–839, 2000.

[40] S. Shian, K. Bertoldi, and D. R. Clarke, "Dielectric elastomer based "grippers" for soft robotics," Advanced Materials, vol. 27, no. 43, pp. 6814–6819, 2015.

[41] T. Li, G. Li, Y. Liang, T. Cheng, J. Dai, X. Yang, B. Liu, Z. Zeng, Z. Huang, and Y. Luo, "Fast-moving soft electronic fish," Science Advances, vol. 3, no. 4, p. e1602045, 2017.

[42] S. Barbarino, E. S. Flores, R. M. Ajaj, I. Dayyani, and M. I. Friswell, "A review on shape memory alloys with applications to morphing aircraft," Smart materials and structures, vol. 23, no. 6, p. 063001, 2014.

[43] H. Rodrigue, W. Wang, B. Bhandari, M.-W. Han, and S.-H. Ahn, "Sma-based smart soft composite structure capable of multiple modes of actuation," Composites Part B: Engineering, vol. 82, pp. 152–158, 2015.

[44] L. Brinson and M. Huang, "Simplifications and comparisons of shape memory alloy constitutive models," Journal of intelligent material systems and structures, vol. 7, no. 1, pp. 108–114, 1996.

[45] C. Liang and C. A. Rogers, "One-dimensional thermomechanical constitutive relations for shape memory materials," Journal of intelligent material systems and structures, vol. 8, no. 4, pp. 285–302, 1997.

[46] L. Li, Q. Li, and F. Zhang, "One-dimensional constitutive model of shape memory alloy with an empirical kinetics equation," Journal of Metallurgy, vol. 2011, 2011.

[47] N. Zotov, V. Marzynkevitsch, and E. J. Mittemeijer, "Evaluation of kinetic equations describing the martensite–austenite phase transformation in niti shape memory alloys," Journal of Alloys and Compounds, vol. 616, pp. 385–393, 2014.

[48] H. Prahlad and I. Chopra, "Comparative evaluation of shape memory alloy constitutive models with experimental data," Journal of Intelligent Material Systems and Structures, vol. 12, no. 6, pp. 383–395, 2001.

[49] Y. Ivshin and T. J. Pence, "A thermomechanical model for a one variant shape memory material," Journal of intelligent material systems and structures, vol. 5, no. 4, pp. 455–473, 1994.

[50] K. Tanaka and S. Nagaki, "A thermomechanical description of materials with internal variables in the process of phase transitions," Ingenieur-Archiv, vol. 51, no. 5, pp. 287–299, 1982.

[51] K. Ikuta, "Micro/miniature shape memory alloy actuator," in Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on. IEEE, 1990, pp. 2156–2161.

[52] A. Nespoli, S. Besseghini, S. Pittaccio, E. Villa, and S. Viscuso, "The high potential of shape memory alloys in developing miniature mechanical devices: A review on shape memory alloy mini-actuators," Sensors and Actuators A: Physical, vol. 158, no. 1, pp. 149–160, 2010.

[53] D. L. Christensen, E. W. Hawkes, S. A. Suresh, K. Ladenheim, and M. R. Cutkosky, "μtugs: Enabling microrobots to deliver macro forces with controllable adhesives," in Robotics and Automation (ICRA), 2015 IEEE International Conference on. IEEE, 2015, pp. 4048–4055.

[54] H.-T. Lee, M.-S. Kim, G.-Y. Lee, C.-S. Kim, and S.-H. Ahn, "Shape memory alloy (sma)-based microscale actuators with 60% deformation rate and 1.6 khz actuation speed," Small, vol. 14, no. 23, p. 1801023, 2018.

[55] T. Andriacchi, E. Alexander, M. Toney, C. Dyrby, and J. Sum, "A point cluster method for in vivo motion analysis: applied to a study of knee kinematics," Journal of biomechanical engineering, vol. 120, no. 6, pp. 743–749, 1998.

[56] H. J. Hermens, B. Freriks, C. Disselhorst-Klug, and G. Rau, "Development of recommendations for semg sensors and sensor placement procedures," Journal of electromyography and Kinesiology, vol. 10, no. 5, pp. 361–374, 2000.

[57] S. H. Chung and C. A. Giuliani, "Within-and between-session consistency of electromyographic temporal patterns of walking in non-disabled older adults," Gait & Posture, vol. 6, no. 2, pp. 110–118, 1997.

[58] M. Arvetti, G. Gini, and M. Folgheraiter, "Classification of emg signals through wavelet analysis and neural networks for controlling an active hand prosthesis," in Proceedings of the IEEE 10th International Conference on Rehabilitation Robotics, 2007, pp. 13–15.

[59] B. Balachandran and E. B. Magrab, Vibrations. Cengage Learning, 2008.

Appendices

Appendix A
Motion Classification Algorithm

Critical to the K-nearest neighbor classifier is the notion of distance. One 50 ms collection of all EMG sensors, $\mathcal{X}$, is defined as a series of N sensor measurements. $\mathcal{X} = \{X^{(1)}, X^{(2)}, ..., X^{(N)}\}$ where each $X^{(i)}$ is a sensor consisting of m_i data points $X^{(i)} = \{x_1^{(i)}, x_2^{(i)}, ..., x_{m_i}^{(i)}\}$. The distance, $d(\mathcal{X}, \mathcal{Y})$, is defined between two sensor-collections $\mathcal{X}$ and $\mathcal{Y}$ via the following:

$$d(\mathcal{X}, \mathcal{Y}) \quad = \sum_{i=1}^{N} \frac{1}{A^{(i)}} (f_i(X^{(i)}, Y^{(i)})) \tag{A.1}$$

where

$$f_i(X^{(i)}, Y^{(i)}) = \begin{cases} \sum_{j=1}^{m_i} \mid x_j^{(i)} - y_j^{(i)} \mid & \text{if } A^{(i)} > \varepsilon^{(i)} \\ 0 & \text{otherwise} \end{cases}$$

with $A^{(i)} = \sum_{j=1}^{m_i} \mid x_j^{(i)} \mid + \sum_{j=1}^{m_i} \mid y_j^{(i)} \mid$ and $\varepsilon^{(i)}$ an appropriately chosen constant for each sensor. Here the term $A^{(i)}$ serves as a normalization in order to equalize the weights of sensors with varying magnitudes, and the inclusion of the constant $\varepsilon^{(i)}$ is to avoid similar "small" signals from producing large distances due to that normalization.

Appendix B
Arduino SMA Bimorph Actuating Code

This code is the arduino code used to osscilate the SMA bimorph actuator.

```
#include <Wire.h>

// Pins for H bridge circuit
#define input1 3
#define input2 4
#define enable1 10
#define input3 6
#define input4 7
#define enable2 9

// Pin for button that starts actuator
#define button 12

// counter variable
int count = 1;

// Initialize variables
float freq; // operating frequency of actuator
float delay_time; // delay time until circuit flips to send current
   through other side of bimorph
int buttonState; // state of button pin
float pwm_out; // PWM signal
float wave_counter = 1; // number of sine waves counter
int freq_count = 0; // number of time that the frequency has been swept
   through


// VARIABLES /////////
float freq_initial = 30; // initial frequency of actuator (Hz)
int num_per_cycle = 5; // number of cycles at each frequency
float freq_inc = 0.5; // increment of frequency sweep (Hz)
float freq_max = 65; // max frequency (Hz)
float freq_min = freq_initial; // min frequency (Hz)
int bit_res = 32; // resolution of PWM sine wave
uint16_t i; // counter for PWM sine wave
unsigned long prevMicros = 0;
unsigned long currentMicros;

// arrays for storage of sine wave PWM signal
const PROGMEM uint16_t DACLookup_FullSine_8Bit[256] =
{
0,4,8,11,15,18,22,25,
25,29,33,36,40,43,47,50,
50,54,57,61,64,68,71,75,
```

```
75,78,81,85,88,91,95,98,
98,101,105,108,111,114,118,121,
121,124,127,130,133,136,139,142,
142,145,148,151,154,157,159,162,
162,165,168,170,173,176,178,181,
181,183,186,188,191,193,195,198,
198,200,202,204,206,208,211,213,
213,214,216,218,220,222,224,225,
225,227,229,230,232,233,235,236,
236,237,239,240,241,242,243,245,
245,246,246,247,248,249,250,251,
251,251,252,252,253,253,254,254,
254,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,254,
254,254,253,253,252,252,251,251,
251,250,249,248,247,246,246,245,
245,243,242,241,240,239,237,236,
236,235,233,232,230,229,227,225,
225,224,222,220,218,216,214,213,
213,211,208,206,204,202,200,198,
198,195,193,191,188,186,183,181,
181,178,176,173,170,168,165,162,
162,159,157,154,151,148,145,142,
142,139,136,133,130,127,124,121,
121,118,114,111,108,105,101,98,
98,95,91,88,85,81,78,75,
75,71,68,64,61,57,54,50,
50,47,43,40,36,33,29,25,
25,22,18,15,11,8,4,1,
};

const PROGMEM uint16_t DACLookup_FullSine_2Bit[64] =
{
0,13,26,39,51,63,76,88,100,
111,122,133,144,154,164,174,
183,192,200,208,215,221,227,
233,238,242,246,249,252,254,255,
255,255,255,254,252,249,246,242,
238,233,227,221,215,208,200,192,183,
174,164,154,144,133,122,111,100,88,
76,63,51,39,26,13,1
};
const PROGMEM uint16_t DACLookup_FullSine_1Bit[32] =
{
0,26,52,77,101, 124, 146, 167,
185, 202, 217, 229, 240, 247,
253, 255, 255, 253, 247, 240,
229, 217, 202, 185, 167, 146,
124, 101, 77,  52,  26,  1
};

void setup() {
  // Initalize the pin modes for all pins
  pinMode(input1,OUTPUT);
```

```
97    pinMode(input2,OUTPUT);
98    pinMode(enable1,OUTPUT);
99    pinMode(enable2,OUTPUT);
100   pinMode(input3,OUTPUT);
101   pinMode(input4,OUTPUT);
102   pinMode(button,INPUT_PULLUP); // use pullup resistor so that code is
       triggered when it goes LOW
103   Serial.begin(9600); // begin Serial communication
104
105   // set inital pin modes H-bridge as LOW/0 so that no current goes
       through actuator
106   digitalWrite(input1,LOW);
107   digitalWrite(input2,LOW);
108   digitalWrite(input3,LOW);
109   digitalWrite(input4,LOW);
110   analogWrite(enable1,0);
111   analogWrite(enable2,0);
112
113   freq = freq_initial; // set operating frequency as initial frequency
114 }
115
116 void loop() {
117   delay_time = ((1/freq)*1000000)/2; // find the delay time based on the
       current operating frequency (microseconds)
118   buttonState = digitalRead(button); // read the button state
119   if (buttonState == LOW) // if buttonState is pulled LOW, begin actuating
120   {
121     // if the frequency is equal to the max, cut off current to the
      actuator
122     if (freq == freq_max)
123     {
124       while (1)
125       {
126         analogWrite(enable1,0);
127         analogWrite(enable2,0);
128         digitalWrite(input1,LOW);
129         digitalWrite(input3,LOW);
130       }
131     }
132
133     // if the actuator has hit the number of cycles per frequency,
      increase the operating frequency by the increment
134     if (count % num_per_cycle == 0)
135     {
136       freq = freq+freq_inc;
137     }
138
139     //Oscillate enable pin 1
140     analogWrite(enable1,0);
141     digitalWrite(input1,HIGH);
142     i = 0;
143     analogWrite(enable1,pgm_read_word(&(DACLookup_FullSine_1Bit[i])));
144     prevMicros = micros();
145     i++;
```

```
    while (i < bit_res)
    {
      currentMicros = micros();
      if (currentMicros - prevMicros >= delay_time/bit_res)
      {
        analogWrite(enable1,pgm_read_word(&(DACLookup_FullSine_1Bit[i])));
        i++;
        prevMicros = currentMicros;
      }
    }
    analogWrite(enable1,0);
    digitalWrite(input1,LOW);

    // Oscillate enable pin 2
    analogWrite(enable2,0);
    digitalWrite(input3,HIGH);
    i = 0;
    analogWrite(enable2,pgm_read_word(&(DACLookup_FullSine_1Bit[i])));
    prevMicros = micros();
    i++;
    while (i < bit_res)
    {
      currentMicros = micros();
      if (currentMicros - prevMicros >= delay_time/bit_res)
      {
        analogWrite(enable2,pgm_read_word(&(DACLookup_FullSine_1Bit[i])));
        i++;
        prevMicros = currentMicros;
      }
    }
    analogWrite(enable2,0);
    digitalWrite(input3,LOW);

    count++;
    }
    // if buttonState is pulled HIGH, reset counter and freq
    if (buttonState == HIGH)
    {
        count = 1;
        freq = freq_initial;
    }
}
```

Appendix C
MATLAB High Speed Camera Processing Code

This MATLAB code imports the video file that captured SMA movement and output displacement.

```
clear; close all;
%find the video file and make a video object
video_name = '35mm_140ma_02_Trim';
v = VideoReader([pwd,'\',video_name,'.mp4']);
videoPlayer = vision.VideoPlayer;
videoPlayer.Position = [136 50 1280 720];

% initialize
actuator_length = 35;
frame_rate = 1000;
frameCount = 0;
iwant = cell([],1); % for easy matrix access

%reads in video and makes it bw
while hasFrame(v)
    video = readFrame(v);
    video = imcrop(video, [1 100 1281 501]);
    thresh      = graythresh(video);
    imageBW     = im2bw(video,0.35);
    BW2         = bwareaopen(imageBW,50);
    stats = regionprops('struct',BW2,'Centroid','MajorAxisLength','
   MinorAxisLength');
    centers = stats.Centroid;
    diameters = mean([stats.MajorAxisLength stats.MinorAxisLength],2);
    radii = diameters/2;
    %hold on
    %viscircles(centers,radii);
   % hold off
    step(videoPlayer, BW2);
    frameCount = frameCount + 1;
%     iwant{frameCount} = video;
%     iwantoriginal(frameCount).video = original;
    %iwant(frameCount).processed = video;
    %iwant(frameCount).mirror_processed = mirror_vid;
    %iwant(frameCount).original = original_;

    %check for extra circles


    %smaller y is origin
    x1 = stats(1).Centroid(1);
    y1 = stats(1).Centroid(2);
    x2 = stats(2).Centroid(1);
```

```
    y2 = stats(2).Centroid(2);

    if y2 > y1 %y1 is fixed point
        %coordinates in relation to fixed point
        tip_x(frameCount) =  x2-x1;
        tip_y(frameCount) =  y2-y1;
    else
        tip_x(frameCount) =  x1-x2;
        tip_y(frameCount) =  y1-y2;
    end
end

tip_x = tip_x .* (actuator_length/tip_y(1));
tip_y = tip_y .* (actuator_length/tip_y(1));
csvwrite(['displacement_data_',video_name,'.csv'],tip_x');
```

Appendix D
MATLAB SMA Displacement Processing Code

This MATLAB code processes the displacement data from the high speed camera and determines the operational frequency.

```
%% PROGRAM DESCRIPTION
%  Name:    nonlinear_analysis.m
%  Author:  Scott Kennedy
%  Sources: N/A
%  Purpose: Inputs motion capture data, determines displacement during
%           various frequencies.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc, clear all, close all
warning('off','all')
set(0, 'DefaultTextInterpreter', 'none')

% Map current directory and locate CSV files
folder_dir = pwd;
source_files = dir(fullfile(folder_dir,'*.csv'));
mkdir plots

% Motion capture sampling frequency (240 Hz)
sample_freq = 1000;
final_mat = [];

% Loop through each source files contatining displacement data
for count1 = 1:length(source_files)
    % Read in data and set time vector
    raw_data = detrend(csvread(source_files(count1).name));
    time = 0:1/sample_freq:(length(raw_data)-1)*(1/sample_freq);
    % Loop through the data of the current source file and determine
    % starting point of activation
    for aa = 1:length(raw_data)
        % Once displacement is greater than this limit, activation has
        % begun and set time and displacement vectors
        if raw_data(aa) > 0
            start_time = time(aa);
            new_time = time(aa:length(raw_data));
            disp_data = raw_data(aa:length(raw_data));
            break
        end
    end
    disp_data = raw_data;
    new_time = time;
    % Find peaks of displacement data
    [pks,pks_loc] = findpeaks(disp_data,'MinPeakDistance',12,'
   MinPeakHeight',0.01);
```

```
    % Initialize storage matrices
    freq_instant = zeros(1,length(pks)-1);
    disp_freq = zeros(1,length(pks)-1);

    % Loop through peaks and calculate frequency and average the
    % displacements
    for count2 = 1:length(pks)-1
        time_lap = (new_time(pks_loc(count2+1))-new_time(pks_loc(count2)))
    ;
        freq_instant(count2) = 1/(time_lap);
        disp_freq(count2) = (disp_data(pks_loc(count2))+disp_data(pks_loc(
    count2+1)))/2;
    end
    % Store the positives values
    freq_instant_pos = freq_instant;
    disp_freq_pos = disp_freq;

    % Flip the data to find peaks on the negative displacement side
    disp_data = -1.*disp_data;
    [pks,pks_loc] = findpeaks(disp_data,'MinPeakDistance',11,'
    MinPeakHeight',0.02);
    % Re-initialize storage matrices
    freq_instant = zeros(1,length(pks)-1);
    disp_freq = zeros(1,length(pks)-1);

    % Loop through data again
    for count2 = 1:length(pks)-1
        time_lap = (new_time(pks_loc(count2+1))-new_time(pks_loc(count2)))
    ;
        freq_instant(count2) = 1/(time_lap);
        disp_freq(count2) = (disp_data(pks_loc(count2))+disp_data(pks_loc(
    count2+1)))/2;
    end
    % Store it in negative (neg) matrices
    freq_instant_neg = freq_instant;
    disp_freq_neg = disp_freq;

    % Put the neg and pos matrices together
    freq_instant = [freq_instant_pos,freq_instant_neg];
    disp_freq = [disp_freq_pos,disp_freq_neg];

    % Create a matrix with the displacement per frequency data
    nat_freq_mat = [freq_instant;disp_freq];
    % Sort the data based on the frequency
    nat_freq_mat = sortrows(nat_freq_mat');

    % Initalize storage matrices and set initial frequencies to first
    frequency
    prev_freq = nat_freq_mat(1,1);
    average_disp_mat = [];
    average_disp = [];
    average_freq = [];
    average_freq_mat = [];

```

```
    % Loop through displacement vs frequency data and average displacement
    % data at same frequency
    count4 = 1;
    for count3 = 1:length(disp_freq)
        if (nat_freq_mat(count3,1) >= prev_freq-0.01 && nat_freq_mat(
    count3,1) <= prev_freq+0.01)

        else
            average_freq_mat = [average_freq_mat,mean(nat_freq_mat(count4:
    count3,1))];
            average_disp_mat = [average_disp_mat,mean(nat_freq_mat(count4:
    count3,2))];
            if (count3 == length(disp_freq))
                count4 = count3;
            else
                count4 = count3+1;
            end
        end
        prev_freq = nat_freq_mat(count3,1);
    end
    % add in final frequency data
    %average_freq_mat = [average_freq_mat,mean(nat_freq_mat(count4:count3
    ,1))];
    %average_disp_mat = [average_disp_mat,mean(nat_freq_mat(count4:count3
    ,2))];
    final = [average_freq_mat;average_disp_mat]';

    % Polyfit
    fit_parameters = fit(average_freq_mat',average_disp_mat','
    smoothingspline','SmoothingParam',0.97);

    struct(count1).name = num2str(source_files(count1).name);
    struct(count1).data = final;
    struct(count1).fit = fit_parameters;

    % Pull out length of actuator
    length_loc = strfind(struct(count1).name,'mm');
    struct(count1).length = str2num(struct(count1).name(length_loc-2:
    length_loc-1));

    % Pull out increase or decrease
    if (strfind(struct(count1).name,'increase') ~= 0)
        struct(count1).direction = 1;
    elseif (strfind(struct(count1).name,'decrease') ~= 0)
        struct(count1).direction = 0;
    end

    % Pull out specific parts of a string
    current_loc = strfind(struct(count1).name,'ma');
    struct(count1).current = str2num(struct(count1).name(current_loc-3:
    current_loc-1));

    % Find natural frequency of actual data
    [max_disp,max_disp_loc] = max(average_disp_mat);
```

```
137     natural_freq = average_freq_mat(max_disp_loc);
138
139     % Find nautral frequency of spline data
140     spline_freq = linspace(average_freq_mat(1),average_freq_mat(length(
    average_freq_mat)),1000);
141     spline_disp = fit_parameters(spline_freq);
142     [spline_max,spline_max_loc] = max(spline_disp);
143     struct(count1).spline_nat_freq = spline_freq(spline_max_loc);
144
145     % Plot
146     cd([folder_dir,'\plots'])
147     figure
148     %plot(new_time(pks_loc),pks,'o')
149     hold on
150     grid on
151     %plot(new_time,disp_data,'b')
152     plot(disp_data)
153     xlabel('Time (s)')
154     ylabel('Displacement (mm)')
155     title(num2str(source_files(count1).name),'Interpreter', 'none')
156     saveas(gcf,['raw_data_',num2str(source_files(count1).name) ,'.fig']);
157     saveas(gcf,['raw_data_',num2str(source_files(count1).name) ,'.png']);
158
159     figure
160     plot(average_freq_mat,average_disp_mat,'.')
161     hold on
162     grid on
163     plot(fit_parameters,average_freq_mat',average_disp_mat')
164     s=findobj('type','legend');
165     delete(s)
166
167     plot(natural_freq,max_disp,'*b')
168     line([natural_freq natural_freq],[0 max_disp],'Color','b')
169     line([spline_freq(spline_max_loc) spline_freq(spline_max_loc)],[0
    spline_max],'Color','b')
170 %     set(0, 'DefaultTextInterpreter', 'tex')
171 %     text(spline_freq(spline_max_loc),spline_max,[' \leftarrow ',num2str(
    round(spline_freq(spline_max_loc),1)),' Hz'])
172     xlabel('Frequency (Hz)')
173     ylabel('Displacement (mm)')
174     axis manual
175     axis auto
176     title(num2str(source_files(count1).name),'Interpreter', 'none')
177     saveas(gcf,['disp_per_freq_',num2str(source_files(count1).name) ,'.fig
    ']);
178     saveas(gcf,['disp_per_freq_',num2str(source_files(count1).name) ,'.png
    ']);
179     cd(folder_dir)
180     set(0, 'DefaultTextInterpreter', 'none')
181
182 end
183
184 %% Plot all data
185 cd([folder_dir,'\plots'])
```

```
colormap winter;
cmap = hsv(length(struct));
figure
hold on
grid on
h = zeros(4,1);
for count1 = 1:length(struct)
    %plot(struct(count1).data(:,1),struct(count1).data(:,2),'.')
    struct_plot = plot(struct(count1).fit,struct(count1).data(:,1)',struct(count1).data(:,2)');
    set([struct_plot(1) struct_plot(2)],'color',cmap(count1,:))
    set([struct_plot(1)],'DisplayName',struct(count1).name)
    h(count1) = plot(NaN,NaN,'Color',cmap(count1,:));
    s=findobj('type','legend');
    set(s,'Interpreter','none');
    delete(s)
end
legend(h,'110 mA','120 mA','130 mA','140 mA')
xlabel('Frequency (Hz)')
ylabel('Displacement (mm)')
title('45mm')
save('45mm_data','struct')

saveas(gcf,['displacement_vs_frequency.fig']);
saveas(gcf,['displacement_vs_frequency.png']);
saveas(gcf,'displacement_vs_frequency.eps','epsc');

figure
hold on
grid on
for count1 = 1:length(struct)
    struct_plot = plot(struct(count1).current,struct(count1).spline_nat_freq,'.','MarkerSize',20);
    set(struct_plot,'color',cmap(count1,:))
end
xlabel('Current Input (mA)')
ylabel('Natural Frequency (Hz)')
saveas(gcf,['frequency_vs_current.fig']);
saveas(gcf,['frequency_vs_current.png']);
cd(folder_dir)
```