2019

Wavelet-Eigen Analysis for Customizable Flexion Classifier Based on Electromyography Signals

Luke S. Bonnen
luke.bonnen@gmail.com

ABSTRACT

WAVELET-EIGEN ANALYSIS FOR CUSTOMIZABLE FLEXION CLASSIFIER BASED ON ELECTROMYOGRAPHY SIGNALS

Luke Bonnen, M.S.
Department of Electrical Engineering
Northern Illinois University, 2019
Dr. Benedito Fonseca, Director

This research covers electromyography (EMG) and its use in prosthetics. Many upper limb amputees use EMG to control a prosthetic limb. While many of these prostheses are proportionally controlled, meaning they can control one movement proportional to the EMG amplitude, there is a continual need to improve function for quality of life. Proportional control is traditionally utilized for opening and closing of the hand. It is the desire of engineers to find ways to add more functionality without sacrificing reliability. There are systems that perform these additional actions, but they have difficulty with reliability because of using a fixed predefined set of features. Through using wavelet transforms and an eigen analysis of the feature space, this research explores the use of a method to develop a customized feature set. By selecting components of a signal that lend themselves to an optimum energy distribution of classes, a custom feature can be extracted. This extracted feature is based on the user's data, allowing customization of the prosthetic control and improving trust in the device.

Keywords: electromyography (EMG), wavelet transform (WT), short-time Fourier transform (STFT), prosthetics, eigenvectors, eigenvalues, statistical detection theory, Neyman-Pearson, Bayesian classification

NORTHERN ILLINOIS UNIVERSITY
DEKALB, ILLINOIS

AUGUST 2019

WAVELET-EIGEN ANALYSIS FOR CUSTOMIZABLE FLEXION CLASSIFIER BASED ON ELECTROMYOGRAPHY SIGNALS

BY

LUKAS S. BONNEN

A THESIS SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE

MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL ENGINEERING

Thesis Director:
Dr. Benedito Fonseca

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

A person subjected to traumatic injury that results in voluntary or involuntary amputation seeks to improve quality of life post-op. For the most functional outcome, an amputee is often fitted with a prosthetic. In some cases, this is a purely mechanical device, which can return function of the absent extremity, although this is often oriented to a few fundamental tasks. To improve quality of life, a prosthetic needs to replicate the functions of an able-bodied limb. Electromyography (EMG) can be utilized for providing this expanded functionality.

Upper extremity prosthetics are the focus of this research. This is because more functionality is lost with the amputation of an arm or hand than with the amputation of a lower extremity.

Electromyography (EMG) is commonly used in prosthetic devices and is one of the leading ways that a user can communicate with a device naturally. The signal is generated by using two or three electrodes to observe electrical potential across a portion of the forearm. This potential will change, depending on the underlying muscle behavior (Webster & Clark, 2010).

Charged molecules and atoms in the tissues allow for potential gradients to exist. When cellular structures allow for depolarization, like in the contraction of a muscle, electrodes placed on the surface of the skin can be used to measure this potential. This potential is projected in the direction of electrode placement and is called a lead (Webster & Clark, 2010).

EMG comes with its own set of problems, as will be discussed, but fundamentally the idea is this: when a person undergoes amputation, the remaining skeletomuscular structures are sufficiently intact for a signal to be measured. For many amputations, this is the case.

To improve quality of life, the function of the prosthetic needs to be more inclusive of those functions provided by an able-bodied hand. For example, some of the desired hand functions would include flexion/extension, pronation/supination, and the ability to grasp.

For a system to classify or detect an action from a signal, some features of the signal must be extracted. Intuitive features could simply be the amplitude of a signal against time. More elaborate features have been extracted from EMG signals and used with classifiers to determine specific actions from the user. For example, mean absolute value (MAV) is a commonly mentioned feature that is used in EMG pattern recognition (Phinyomark, Limsakul, & Phukpattaranont, 2009). MAV is a time domain feature, but there are also many features extracted from the frequency domain of a signal. An example of a frequency domain feature is modified mean frequency (MMNF) (Phinyomark et al., 2009). MMNF is a feature that species where the mean of the frequency spectrum is located (Phinyomark et al., 2009). A frequency transform can be thought of as breaking the signal down into components that are members of some frequency bin. By calculating the mean of this distribution, a feature can be extracted (Phinyomark et al., 2009).

As stated, features like MAV, MMNF, or frequency-based approaches have been used to build EMG classifiers with additional functionality. However, one difficulty that arises in producing commercial prostheses is the variation of need between individual users. A given set of features may work well in distinguishing the different movements of one amputee, but not work well for another. This is the problem with classification vs. completion accuracy (Roche, Rehbaum, Farina, & Aszmann, 2014). In other words, it is difficult to build a global classifier with a set of features that works well for all amputees. Because all amputees have different conditions, they require different, customized solutions for determining the best set of features for their use.

This research will look at using a wavelet transform (WT) and eigen analysis to create features based on an individual's EMG data. Because EMG signals vary from one person to another, it is desirable to have a procedure that produces a customized detection scheme for individual amputees. It is the goal of this thesis to present a possible method based on features retrieved from a wavelet decomposition. The hope is that this method could then be scaled to an increased number of people with similar results, creating a way to dynamically optimize a feature for an individual.

Literature Review

Electromyography has a broad range of applications. This research was focused on an area regarding limb loss, and more specifically the loss of an upper extremity.

Myoelectric control of a prosthesis is a common control scheme for many upper limb amputees (Roche et al., 2014). Types of control can be categorized as proportional control and classifier control (Roche et al, 2014). Proportional control is any control that relates the user's EMG amplitude with a physical output of the prosthesis, such as force, velocity, or position

(Roche et al., 2014). Classifier control relates to classifying a given action using some set of features (Roche et al., 2014).

To date, many prosthetic hands have taken advantage of proportional control because it is robust (Roche et al, 2014). As well as being robust, using proportional control allows the user to feel more in control of the hand. Proportional control systems respond in the time domain and are usually limited in actions that can be controlled (Roche et al., 2014). In most systems, the amputee can control the opening and closing of the hand proportionally using one or two sensors (Roche et al., 2014). When using one sensor, the open and close actions are controlled based on the amplitude of the EMG signal (Roche et al., 2014). If the signal is strong, the hand will open; if the signal is weak, the hand will close; and if no signal is detected, the hand will stay in its current state (Roche et al., 2014).

When using some type of classifier control, a common method is the use of pattern recognition (Roche et al., 2014). In pattern recognition, a set of features are used to determine the given actions of the limb (Roche et al., 2014). The system is trained to interpret the given action through observing changes or patterns in the features selected for the signal (Roche et al., 2014). Early pattern recognition started in the 1970s but did not emerge in practice until the early 1990s (Roche et al., 2014). This was in part due to hardware and software limitations (Roche et al., 2014). Even today, pattern recognition still has difficulty finding its way into commercial devices (Roche et al., 2014). One of the primary reasons that adoption of pattern recognition has been slow is due to its lack of robustness (Roche et al., 2014). Electrode movement, arm fatigue, and arm posture can all effect the EMG signal, and in a more advanced classifying system, these disturbances can result in errors (Roche et al., 2014).

Furthermore, in classifier control there is a discrepancy between academic results and clinical practice (Roche et al., 2014). In the literature, it is common for results to be reported with respect to classification accuracy, but in practice the ability of an amputee to control a hand to execute a given command (known as completion) did not correlate strongly with classification accuracy of the system (Roche et al., 2014). This is due in part to some of the issues with robustness stated earlier. It is for a combination of these reasons that more complex control strategies, like pattern recognition, have struggled to be adopted in commercial products (Roche et al., 2014). The need for reliable control is the primary requirement of a prosthetic hand. Regardless how many actions the hand can classify, the hand must be accurate in classifying the user's command correctly; otherwise, using the hand could be unsafe (Roche et al., 2014). If a patient is concerned about the hand spilling hot water on them, they will not use their prosthetic (Roche et al., 2014). It is for this reason that many prosthetic hands are still reliant on simple proportional control (Roche et al., 2014).

Though control strategies have been advancing academically, as we have seen, commercial products have been slow to adopt (Roche et al., 2014). Roche, Rehbaum, Farina, and Aszmann noted in their literature review that while technology has been advancing, rejection rates for prosthetics have not declined, "with 35% of [pediatric] and 23% of adult amputees discontinuing use of their prosthetics" (Roche et al., 2014).

The nature of EMG is inherently random and the amplitude of EMG occurs in the range of 0–10 mV (Reaz, Hussain, & Mohd-Yasin, 2006). EMG acquires noise as the signal propagates through the tissues (Reaz et al., 2006). There are many different types of noise that can corrupt the signal, most notably ambient noise and motion artifacts (Reaz et al., 2006). Ambient noise comes about from externally produced electromagnetic radiation (Reaz et al., 2006). Electromagnetic noise is all around us and is not something that can be easily controlled. It is for this reason that it is considered a noise source. The other type of noise is due to motion artifacts, which arise because the sensor is attached to the measurement device via leads (Reaz et al., 2006). If these leads are moving, they can add transients into the system, due to their representative component values shifting.

EMG is a nonstationary signal in practice (Reaz et al., 2006). The non-stationarity can be broken down into two main categories: slow and fast non-stationarity (Reaz et al., 2006). Slow non-stationarity comes about from the accumulation of metabolites, charged by products of fatigue in the muscle (Reaz et al., 2006). Fast non-stationarity is caused by some biomechanical event, such as lifting or purposefully activating a muscle (Reaz et al., 2006). It should also be noted that changes in muscle force can cause a change in the EMG frequency spectrum (Reaz et al., 2006).

There are many proposed features of EMG signals. This paper will focus on some of the popular features and some novel ones, discussed in the literature.

As stated earlier, MAV and MMNF are features that have been used as EMG features (Phinyomark et al., 2009). MAV is defined as follows:

$$MAV = \frac{1}{N}\sum_{i=1}^{N}|x_i| \tag{1}$$

where x_i is the sample at index ‘i,’ and N is the length of the sample vector. This can be used with a rectified EMG signal, because it requires the use of the absolute value operator (Phinyomark et al., 2009). Being able to use a rectified EMG signal is useful because it does not require re-referencing of the signal before sampling. When rectified, the signal can be readily quantized by the ADC.

According to Phinyomark et al., 2009, there are two other versions of MAV; one is the modified mean absolute value 1 (MMAV1) and the other is the modified mean absolute value 2 (MMAV2). MMAV1 uses a weight for the window that is used, as follows:

$$MMAV1 = \frac{1}{N}\sum_{i=1}^{N} w_i|x_i| \tag{2}$$

where $w_i = \begin{cases} 1, if\ 0.25N < i < 0.75N \\ 0.5, otherwise \end{cases}$

MMAV2 uses a similar window, but treats the beginning and end regions of the signal differently.

$$MMAV2 = \frac{1}{N}\sum_{i=1}^{N} w_i|x_i| \tag{3}$$

where $w_i = \begin{cases} 1, if\ 0.25N < i < 0.75N \\ \frac{4i}{N}, if\ 0.25N > i \\ \frac{4(i-N)}{N}, if\ 0.75N < i \end{cases}$

These are some examples of time domain features (Phinyomark et al., 2009). Some other time domain features are zero crossing, EMG variance, waveform length, and histograms. (Phinyomark et al., 2009).

A feature is just some way to distinguish one signal from another. Just as there are time domain methods for this, there are also frequency domain methods (Phinyomark et al., 2009).

Some frequency domain methods found in literature are the mean frequency median frequency, and the modified mean (MMNF) and median (MMDF) frequencies.

$$\sum_{i=1}^{MMDF} A_i = \sum_{i=MMDF}^{N} A_i = \frac{1}{2}\sum_{i=1}^{N} A_i \tag{4}$$

Where A_i is the value of the amplitude spectrum at frequency bin ‘i’ (Phinyomark et al., 2009), MMDF is then the feature which is used. The MMNF is another technique for feature generation from a signal and is defined as:

$$MMNF = \frac{\sum_{i=1}^{N} f_i A_i}{\sum_{i=1}^{N} A_i} \tag{5}$$

where f_i is the value of the frequency at frequency bin ‘i’. MMNF is the feature given by calculating the mean of the amplitude spectrum of the signal (Phinyomark et al., 2009). Note that the MMNF looks much like the definition of expected value, assuming a uniform distribution for A_i. These features came from using the amplitude spectrum of the signal. A similar set of features can be found using the power spectrum of the signal. These features are known as the median frequency and the mean frequency. They are defined as above where A_i is replaced by P_i, P_i representing the power spectrum of the signal (Phinyomark et al., 2009).

EMG variance is a common feature that is also evident in the literature (Phinyomark et al., 2009). In the case of EMG signals, the mean of the signal is approximately zero, so we can estimate the variance as follows: (Phinyomark et al., 2009)

$$VAR = \frac{1}{N}\sum_{i=1}^{N} {x_i}^2 \tag{6}$$

where x_i are the realizations of N total data.

As mentioned earlier, EMG signals are non-stationary, meaning they change randomly in time. This random change can result in the frequency spectrum changing (Phinyomark et al., 2009). Due to non-stationarity, frequency transforms like the Fourier transform are less appropriate, because the frequency content can change over time.

Time-Frequency Transforms

The non-stationarity of the signal motivates the use of time-frequency analysis methods such as the short-time Fourier transform (STFT) and the WT (Pattichis & Pattichis, 1999). The STFT is computed by windowing time data (which gives time localization), then computing the Fourier transform of the windowed signal. By windowing in time, the time domain operation is a multiplication, which reflects itself as a convolution by the windowing function in the frequency domain. So, by doing the STFT, the frequency information of the signal is altered due to the window that is used. However, this method does allow for time-frequency resolution. The window length can be selected such that the signal captured by the window can be assumed stationary over the length of the window.

The WT is like the STFT, in that a window function (in this case called a scaling and wavelet function), is correlated with the signal at different levels. A level can be thought of as just a scaled version of the mother scaling and wavelet function. The advantage of the WT is that it decomposes the signal into an orthonormal basis (Pattichis & Pattichis, 1999), meaning that every level of coefficient is a unique signal component (Pattichis & Pattichis, 1999). Many signals can be used as wavelet functions. In this research, the Haar wavelet will be used to decompose the EMG signal at different levels. In the continuous wavelet transform (CWT), there are an infinite number of scales to completely recreate a signal (Burrus, Gopinath, & Guo, 1997). In the discrete wavelet transform (DWT), the scale can only go up to the level were individual samples are represented. This means that a sampled signal can be completely reconstructed from the wavelet and scaling coefficients (Burrus et al., 1997).

The DWT can completely reconstruct the sampled signal through the sum of the scaling coefficients and the wavelet coefficients at different scales (Burrus et al., 1997). In order to determine the wavelet or scaling coefficients, the signal is projected onto the wavelet of a certain position and scale (Burrus et al., 1997). To see this, start with the representation of the scaling function dependent on the next scale. A derivation of scaling coefficients, starting with the recursive representation of the scaling function (Burrus et al., 1997), is shown as follows:

$$\varphi(t) = \sum_{n} h(n)\sqrt{2}\varphi(2t - n) \tag{7}$$

$$\begin{aligned} \varphi\left(2^{i}t - k\right) &= \sum_{n} h(n)\sqrt{2}\varphi\left(2\left(2^{i}t - k\right) - n\right) \\ &= \sum_{n} h(n)\sqrt{2}\varphi(2^{i+1}t - 2k - n) \end{aligned} \tag{8}$$

$$\varphi\left(2^{i}t - k\right) = \sum_{n} h(m - 2k)\sqrt{2}\varphi(2^{i+1}t - m) \tag{9}$$

where $\varphi(t)$ is the scaling function and $h(n)$ are the coefficients for the next scale. The coefficients can be found by using an inner product (Burrus et al., 1997).

$$c_j = \langle f(t), \varphi_{j,k}(t)\rangle = \int f(t)2^{\frac{j}{2}}\varphi(2^{j}t - k)dt \tag{10}$$

Equation (9) can be substituted into Equation (10) (Burrus et al., 1997).

$$c_j(k) = \sum_m h(m - 2k) \int f(t)\, 2^{(j+1)/2} \varphi(2^{j+1}t - m)dt \tag{11}$$

$$c_j(k) = \sum_m h(m - 2k) c_{j+1}(m) \tag{12}$$

Equation (12) shows that the current scale can be expressed in terms of the coefficients of the higher scale. Being able to completely represent the original signal under the DWT at the highest scale becomes advantageous, because the highest scales are the samples themselves (Burrus et al., 1997). By starting at the highest scale, just the samples—filter banks expressed in terms of the coefficients h(-n)—can be used with downsampling by 2 to produce the next lower scale (Burrus et al., 1997). This is a common way of looking at a WT and is how it is utilized in this research. Note that while the equations show this process using the scaling coefficients, a similar process gives the wavelet coefficients. Once the coefficients are determined, the discrete signal f[n] can be reconstructed by the following (Burrus et al., 1997):

$$f[n] = \sum_k c_{j0} \varphi_{j0,k}[n] + \sum_k \sum_{j=j0}^{J-1} d_j(k) \varphi_{j,k}[n] \tag{13}$$

The power in different levels of the signal can be computed or the power of the entire signal f[n] can be computed from the coefficients by the following (Burrus et al., 1997):

$$\sum_{n=-\infty}^{\infty} |f[n]|^2 = \sum_{l=-\infty}^{\infty} |c(l)|^2 + \sum_{j=0}^{\infty} \sum_{k=-\infty}^{\infty} \left|d_j(k)\right|^2 \tag{14}$$

In the DWT, the level does not approach infinity. Because of this, the power can be computed by summing the square of the scaling coefficients at the lowest level and the square of the wavelet coefficients up to the highest level, i.e., the level of the samples themselves (Burrus et al., 1997).

The difference between the WT and the STFT is that in the STFT, the signal of a window is decomposed into orthogonal sinusoidal components (composing a basis) where the WT uses a different basis function. In the WT, the signal is decomposed into orthonormal basis functions that are limited in time, as opposed to the case of the sinusoidal basis used in Fourier analysis (Pattichis & Pattichis, 1999). The advantage here is while for different window overlaps the STFT might not be orthogonal; the WT will always give orthogonal coefficients (Pattichis &Pattichis, 1999). This becomes useful for creating a basis with which to evaluate features.

The WT uses a set of windows to correlate with the signal (Pattichis & Pattichis, 1999). The windows enable time-frequency resolution; however, the more frequency resolution of the wavelet, the less its time resolution, and vice versa (Burrus et al., 1997; Pattichis & Pattichis, 1999). This is because to have time resolution the wavelet window needs to be small. If the window is small, then the frequency information is limited to only higher frequencies, due to being unable to represent lower frequencies (Burrus et al., 1997; Pattichis & Pattichis, 1999). Likewise, if the window is large then the occurrence of an event in time is hard to localize; however, the frequency content of the window is a better representation of the frequency content of the signal (Burrus et al., 1997). In contrast, the STFT uses a predetermined fixed window size to sweep the signal (Pattichis &Pattichis, 1999). The WT scaling functions for finding $c_{j,k}$ do not overlap; they are orthogonal at each level, as stated earlier (Pattichis & Pattichis, 1999).

The STFT window does have overlap, and overlap is used to improve time resolution, but it comes at the cost of non-orthogonality (Pattichis & Pattichis, 1999). To understand more fully how the WT compares to the STFT, consider what a basis function is. A basis function is some function used to represent components of a signal. For example, a Dirac delta function could be a basis function of a signal. A signal could be thought of as a linear combination of scaled and shifted deltas (Burrus et al., 1997). The frequency domain representation of the signal would then be the linear combination of the Fourier transform of the shifted and scaled Dirac deltas. Note that in this case, the time resolution is the best; however, in the frequency domain there is very poor frequency localization (Burrus et al., 1997). In comparison to the Fourier basis, which is composed of sinusoids, the Dirac basis represents only an instant in time (Burrus et al., 1997). The advantage of using a sinusoidal basis is that in the frequency domain the localization is the most effective acting as a Dirac delta in the frequency domain (Burrus et al., 1997).

To understand more clearly how wavelets work, let us first look at the discrete STFT (DSTFT) in detail. Here, the basis function can be represented as follows:

$$w_{j,k}(t) = w(t - k\tau_0)e^{-jw_0lt} \tag{15}$$

where $w(t - k\tau_0)$ represents a shifted window and e^{-jw_0lt} represents the Fourier basis function (Burrus et al., 1997). This means that as the window size is increased towards infinity, the Fourier transform of that window will approach a Dirac delta (Burrus et al., 1997). In the Fourier domain, the Fourier basis is represented by deltas at the corresponding frequency locations (Burrus et al., 1997). Note, there is no uncertainty in frequency in this case, but there is a very large (approaching infinite) uncertainty in time (Burrus et al., 1997).

As the windowing function length is decreased in time, the frequency representation of that windowing function becomes more and more spread. If this continues to occur, the convolution in the frequency domain of a wide function and the Fourier bases representation of a signal will result in a wider, less resolute function being expressed at the frequency locations of interest. These frequency locations (previously just the deltas representing the Fourier basis components) are now spread in frequency (Burrus et al., 1997). By decreasing the window size, in effect what has occurred is the time localization has improved but the frequency resolution has become more uncertain (Burrus et al., 1997). In fact, there is an uncertainty relation between time and frequency, such that the more resolution there is in time, the less resolution there is in frequency, and vice versa (Burrus et al., 1997).

Eigenvectors and Eigenvalues

Eigenvalues and eigenvectors allow for a k-dimensional space to have an orthogonal representation where one axis is along the average signal direction. Points in a space represented by some basis can be represented by a different orthogonal basis, such that the points are not altered. In other words, the points in the space stay the same but the description of the points can change. This is the advantage of eigenvectors. The eigenvectors compose an eigenspace where all the entries of the matrix are linearly independent and span the space. The eigenvectors are also orthogonal, which is advantageous when describing the space. Any point in the space can be described in terms of a combination of these orthogonal vectors, which is used in this research to improve class separability. In essence, an eigenvector is any vector that satisfies the following (Lay, 2012):

$$A\boldsymbol{x} = \lambda\boldsymbol{x} \tag{16}$$

where A is any matrix transformation and λ represents a constant known as an eigenvalue (Lay, 2012). An interpretation of the above equation is that an eigenvector $\mathbf{x}$ is any vector, under transform results in a dilation or contraction of the same vector **x** (Lay, 2012). From Equation (16), the eigenvector can be interpreted as the null space of the following (Lay, 2012):

$$(A - \lambda I)\boldsymbol{x} = \mathbf{0} \tag{17}$$

This expression can be used to find the eigenvectors for a specific eigenvalue (Lay, 2012). Thus, once the eigenvector has been found, it can be interpreted as dilating or contracting the eigenspace (Lay, 2012).

In this research, eigenvectors and eigenvalues will be used to find the best combination of wavelet coefficients to represent each class. Multiple signals will be collected from each class and the wavelet coefficients will be extracted from these signals. The set of wavelet coefficients for each class will form a column in matrix A, which will be called the "feature matrix." Then, eigenvectors and values will be computed for $A^T A$. This is key to obtaining an orthogonal space. The only way we can be guaranteed to achieve an orthogonal eigenspace is if our transform matrix is square symmetric.

CHAPTER 2

PROBLEM STATEMENT

Improving quality of life is arguably the most important feature of a prosthetic. For the user to live with more functionality from a prosthetic, that device needs to be able to replicate more actions of the able-bodied limb. Traditionally, as stated previously, most upper limb prosthetics are proportionally controlled, meaning those devices have limited functionality (Roche et al., 2014). This is done to bolster predictability and reliability of the prosthetic. If the patient loses trust in the device, they will be more likely to reject it (Roche et al., 2014). Many upper limb prosthetics use classifiers to increase function (Roche et al., 2014); however, these have been slow to be adopted in commercial products because of a lack of reliability (Roche et al., 2014).

There are many features used in pattern recognition, some of which have already been discussed in detail. The question is what makes a good feature? In this case, many features show drift over time, which results in increased error by the classifier. It is for this reason that many academic detection schemes are not adopted into commercial products (Roche et al., 2014).

An approach of interest would be to determine whether a method could be implemented to generate features and their associated classifier to reflect a specific patient's signals in an optimal way. The goal is not to determine a set of features to evaluate all patients; it is to propose a method that allows the data to help determine a feature set for a specific patient. It is the goal of this research to test such a method, using wavelet decomposition to extract features from the signal.

CHAPTER 3

PROCEDURE

The signal processing technique used was a WT. The wavelet decomposition was implemented using Python’s wavelet packet library. These wavelet coefficients at all levels (to a maximum of 18 levels) were then collected to produce a feature set for a given class. These feature vectors represent information collected from the signal over a duration of 500 samples, or 62.5 ms. Over the 500 samples, the decomposition gives scaling and wavelet coefficients. These arrays of coefficients were averaged, and their variance, power, was taken as features representing the 500 samples. This was done for the entire 10 s file for each class. The total set of these feature vectors represents a matrix. To produce a square matrix, the current feature matrix was multiplied by its transpose. This operation will produce an 18 × 18-dimensional matrix that is symmetrical. This symmetry allows for the eigenvalue decomposition.

This new feature matrix was then treated as a representation of some feature space. This feature space, which is an orthogonal space, was re-represented using the eigenvector representation of the space, the goal being to re-orient the space in such a way that the strongest eigenvector was where most of the signal was expressed. In other words, the strongest eigenvector represents the vector along which the largest energy in the signal is seen. By removing the other eigenvectors, the signal is in effect being de-noised. From here, the eigenvectors associated with the most powerful eigenvalues were used to determine the necessary eigenvectors to represent the signal. Then, the feature vectors were projected into this new space, distributions observed, and a method for building a classifier/detector implemented to evaluate the method.

Method

The goal of the detection and classification system is to get the device to “learn” the user. Simple threshold EMG prosthetic systems exist that rely on a prosthetist to tune a threshold. However, this threshold is set in a clinical environment and may not accurately represent the amputee’s usage of the arm in multiple environments over long periods.

Prosthetics systems exist that allow for pattern recognition, and have thresholds that the system can calibrate. However, these may not be looking at the best set of features and may not be selecting the best threshold.

The goal of this research was to investigate methods for selecting the best subset of features given some initial set. In this case, the original set of features were the wavelet and scaling coefficients taken from a time-sampled forearm EMG signal. To find the best threshold with these features, a Bayes risk classifier was implemented, which takes the most powerful signal features and uses them to build a detector. Then, a classifier was built to determine the best theoretical threshold set for an individual’s classes.

To simulate a user’s data over time, the original features were collected subjectively. Instead of setting up predetermined tasks or acceptable ranges for the data, the data was instead collected based on what seemed like a relatively good measure for a class. The person collecting the data decided ahead of time what class the recorded data would belong to. Then, based on what felt like a good representation for that class, the signal would be generated. This subjective method of data collection was known to be error prone and it was of interest to determine whether the wavelet-eigen approach would improve the ability to determine a threshold.

Measurement and Data Collection

A single channel EMG sensor and AVR microcontroller was used to collect data from a subject. This data was then stored and used for processing on a computer. The signal processing tool used was PyCharm Community, which is an open source IDE using Python. To conduct the wavelet decomposition, the Pywavelets library functions were implemented. These functions were specifically: the dwt using the Haar wavelet, the wavelet function to build the Haar, and maxlevel was used to determine how many rounds of decomposition would be needed. This library allows for a wavelet packet representation at any level, and the ability to conduct the decomposition level by level. This ability was used to determine whether one or more levels of decomposition should be used to generate features.

Once the feature set was determined for one set of samples, it was repeated through all samples to represent an ensemble of that class. The classes were as follows: resting, light flexing, medium flexing, and heavy flexing. Data was collected for each of these classes and features were constructed.

Once the features where constructed, different techniques were used to determine how well that set of features were able to separate the classes. One preliminary technique to evaluate features is to represent the classes simply in terms of their features on a scatter plot. The scatter plot enables the researcher to see the classes in terms of their features. This technique only works out to three features on a 3D plot but it is still useful, because it gives a quick and simple way to visualize large amounts of data, showing how the classes are spread based on the features chosen.

The final evaluation technique for the features was via a detector. The data was split into a training group and a testing group. The training group set up where the threshold should be, and the testing group was used to evaluate that threshold via a Monte Carlo simulation. The results then gave a probability of detection to quantify how often, when using that threshold, the detector decided for the correct action. Once the system detects for action, it will then decide for one of the three classes (light flex, medium flex, and heavy flex). Some modifications were made to compare results between all three classes and using just two classes. Justification for this will be presented in a later chapter.

The heart of the measurement device is an instrumentation amplifier. This is necessary to provide a large input impedance for safety and a large common mode rejection ratio (CMRR). The following stages are utilized to condition the signal further before sampling. Common stages to include would be a gain stage and (in some cases) a filter to remove high frequency noise (Webster & Clark, 2010).

Two electrodes are necessary to capture the signal across a muscle. In some systems, a third electrode is used to set a reference and to provide protection for the user (Webster & Clark, 2010). A third electrode system is commonly known as a “driven right leg” (DRL) (Webster & Clark, 2010). DRL comes from electrocardiogram systems in which an electrode is attached to the right leg of a patient to provide current limiting and control the common mode voltage (Webster & Clark, 2010).

DRL replaces just grounding an electrode with an active circuit that is used to limit current drawn from the electrode when the active circuit saturates. When the active device is not saturated, the circuit provides a low-impedance path to ground that is much less than it would be with just a lead connected directly to ground. The overall effect is that the common mode voltage is shorted out, which helps improve the CMRR (Webster & Clark, 2010). A typical biopotential amplifier can be realized with just general-purpose operational amplifiers (Webster & Clark, 2010).

Many researchers use multiple EMG sensors to produce a multiple channel representation of an action (Phinyomark et al., 2009). In this research, only a single EMG sensor was considered, and the focus was on a method for creating a feature set to improve class separability and detection. The EMG sensor used was an Advancer Technologies Muscle Sensor V2 (Webster & Clark, 2010). To conduct this research, data was first collected using a Teensy 3.6 MCU. The data was sampled at 8 Ksamples/s for a total of 10 s, resulting in 80 Ksamples of data for each file. The literature comments on sampling frequencies of 1 KHz (Phinyomark et al., 2009). The reason for using 8 KHz was out of interest in seeing a better representation of the analog data spectrum.

Data collection was attempted with an ATMEL MCU, which had 2 K of RAM; however, the storage of an 8-bit value per sample quickly filled the on-board memory. For this research, there was a need to view sampled data over a continuous 10 s interval of time. For this reason, a new MCU was chosen with 256 KB of on-board memory. The on-board ADC was a 10-bit converter which, if stored in memory at the desired sampling frequency, was still over-flowing on-board memory. The solution was to have a location in the memory to store the 10-bit converted value from the ADC, then the storage array was located at a different location. Data was converted into the 10-bit value; then, when stored in the array, the data was transformed into an 8-bit representation. This allowed all 10 s for 8 KHz sampled data to be stored on-board. Figure 1 illustrates the process of this system.

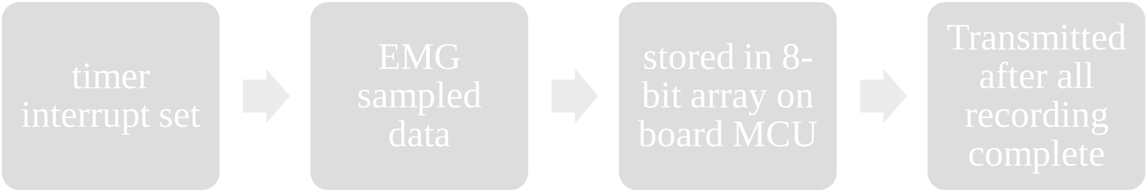
timer interrupt set
EMG sampled data
stored in 8-bit array on board MCU
Transmitted after all recording complete

Figure 1. System diagram of the measurement and storage device.

Feature Construction

Before getting into using the eigenvalue decomposition it is important to show why the wavelet coefficients were not sufficient. Some features are extracted here using the WT and the DSTFT. This is done to create a series of scatter plots to show how the classes initially interact.

Data collection was achieved via the method stated earlier and a DSTFT and DWT were used. The DWT resulted in the cleanest feature set, because it results in orthogonal features.

The DWT was first done at level 8, which resulted in the scaling coefficients and the wavelet coefficients at this level. These coefficients where taken as features for the signal and the scatter plot illustrated in Figure 2 was created.

Notice in Figure 2 that at this level there is poor class separability for all four classes. This is true for all levels of decomposition when only looking at one level. Next, look at two different levels of decomposition. To achieve this, two sets of wavelet coefficients must be observed, because the scaling coefficients between two different levels may not be orthogonal but all wavelet coefficients are orthogonal.

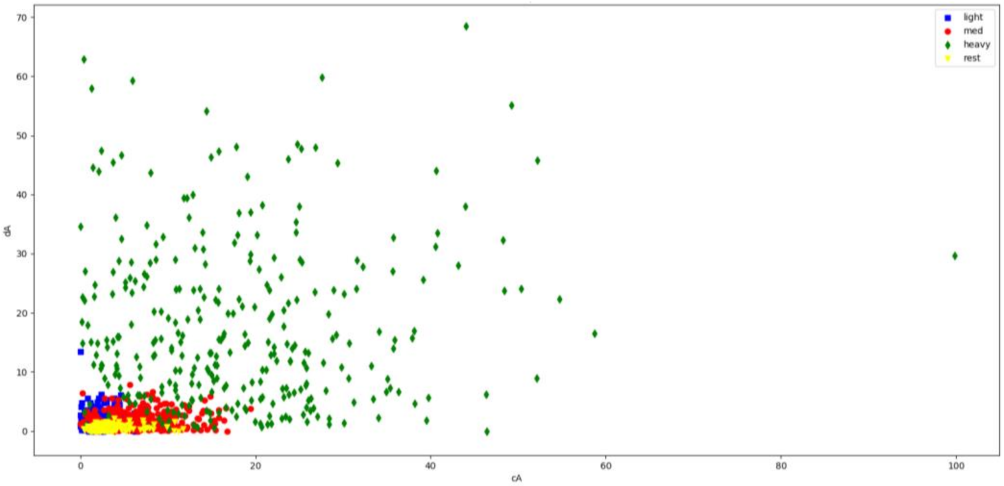

Figure 2. WT level 8 scaling vs. wavelet coefficients for four signal classes.

Notice in Figure 3 that here class separability has not improved. This same decomposition was tried for a few different levels, with the same results. The next step was to check whether the DSTFT would give any better class separability. It was expected to give a similar result because it is similar to a DWT at only a single level.

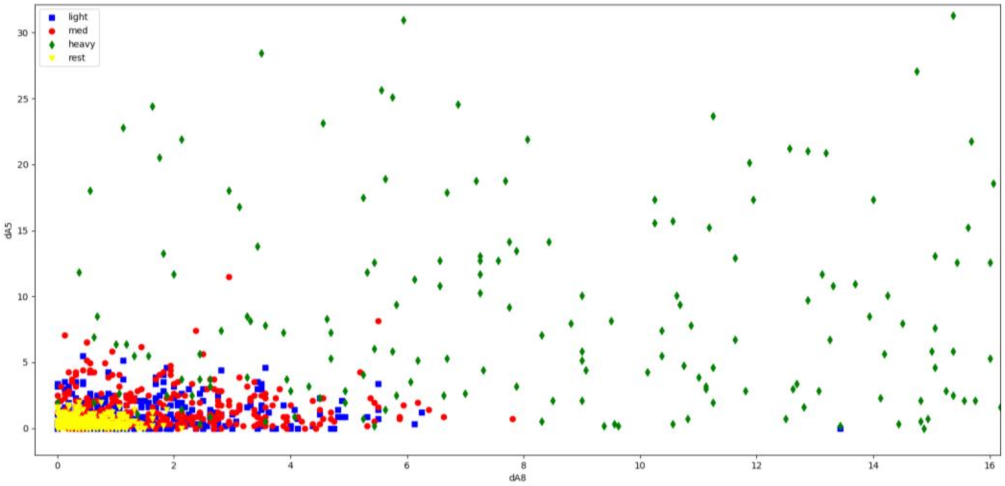

Figure 3. WT of wavelet coefficients at level 5 vs. coefficients at level 8.

In Figure 4 it can be seen that the class separability has not improved, which was expected. The features have a large degree of variance. There was an interest in seeing whether the class separability would improve if the variance was decreased. The DWT features at level 8 where averaged over a window. That window was 20 coefficient samples in length. The idea of this was to reduce the variance by 1/20, and hopefully improve class separation.

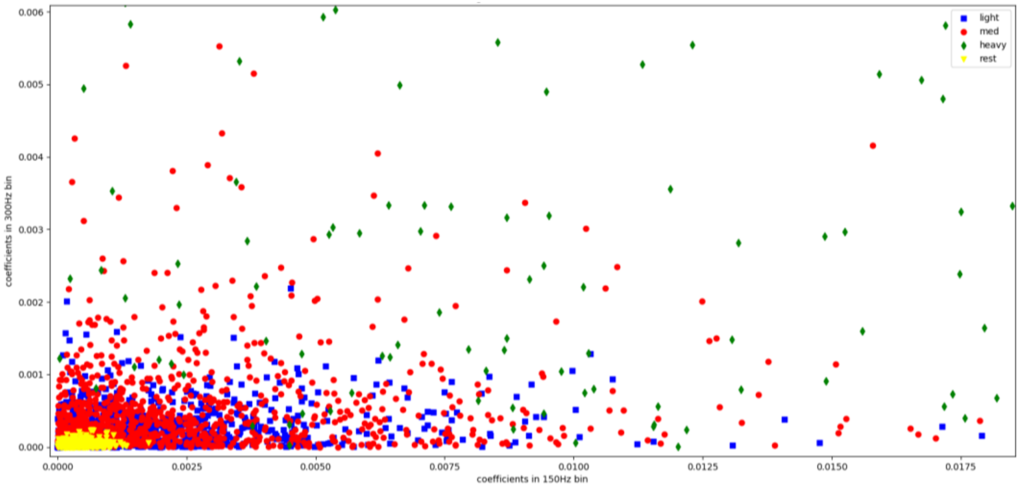

Figure 4. DSTFT of coefficients at 300 Hz vs 150 Hz.

As can be seen in Figure 5, while the variance did decrease, the separation did not improve enough to make a good classifier. This was also done with the wavelets at levels 5 and 8 from Figure 3 shown earlier with similar results. It is apparent with these results, more needed to be done to improve class separation.

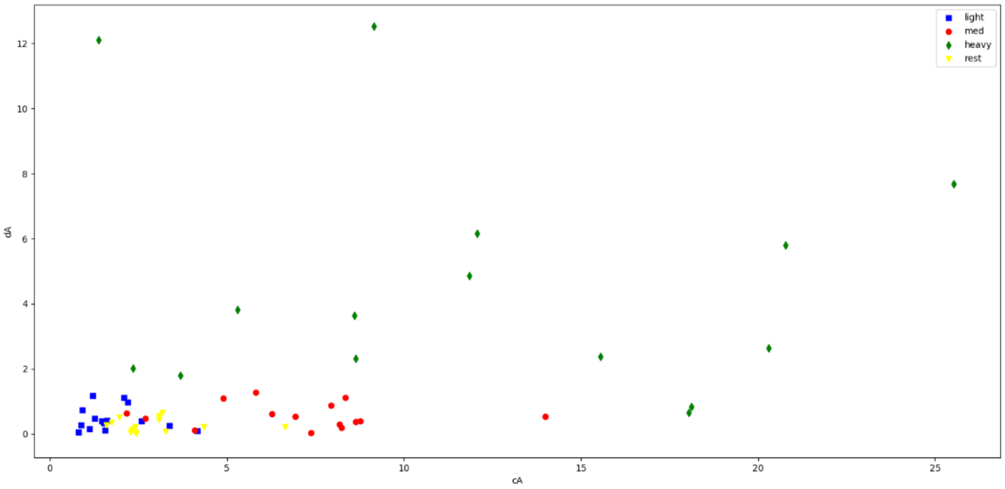

Figure 5. DWT at level 8 of wavelet coefficients vs scaling coefficients when averaged over a window of length 20.

Eigenvalue Analysis

A matrix could be constructed describing the space in terms of recorded signal wavelet vectors. This would lead to a customized coordinate system, to express an individual's signals in all classes.

All coefficients at all levels of decomposition over a 500-sample long window were taken. Then, computing their averages and variances over that window, a wavelet feature was made. This wavelet feature describes the individuals signal over the time window. The wavelet features, representing the entire 10 s recording period, now represent a vector space, where each column vector describes the average and variance of the wavelet coefficients at all levels, up to the maximum level, for the 500-sample long time slice. This new matrix was built for each class and then all matrices were concatenated. The concatenated matrix represents the signal under all conditions. This matrix is then multiplied by its transpose in order to create a square matrix. This square matrix can be thought of as a power matrix, representing the signal under all conditions. The importance of building a square symmetric matrix is to be sure that the eigenvectors, representing that space, will be orthogonal to each other.

A power matrix was constructed to represent the new features in a square, symmetric space, at which point the eigenvalues were found. It was important to find which were the largest eigenvalues. This gives an indication of significance for the corresponding eigenvector, when it comes to expressing the space. The expectation was that there would be some eigenvalues that were much stronger than the others, and that those weaker eigenvectors could be omitted.

In Figure 6, the magnitudes of the corresponding eigenvalues for all four classes are shown: resting, light flexing, medium flexing, and heavy flexing. Here, it can be seen that the first eigenvector is the most common dimension of the signal in all four classes. There are other non-zero eigenvalues corresponding to the eigenvectors shown in the plot. However, the effect of these dimensions in describing the signal may be only over complicating the model. The signal in the first eigenvector was so strong that the other values relative to it could be considered negligible. This is not a certainty, so here the first two eigenvectors have been chosen to construct a model of the signal. (It will be shown later that it is only necessary to consider the first eigenvector.)

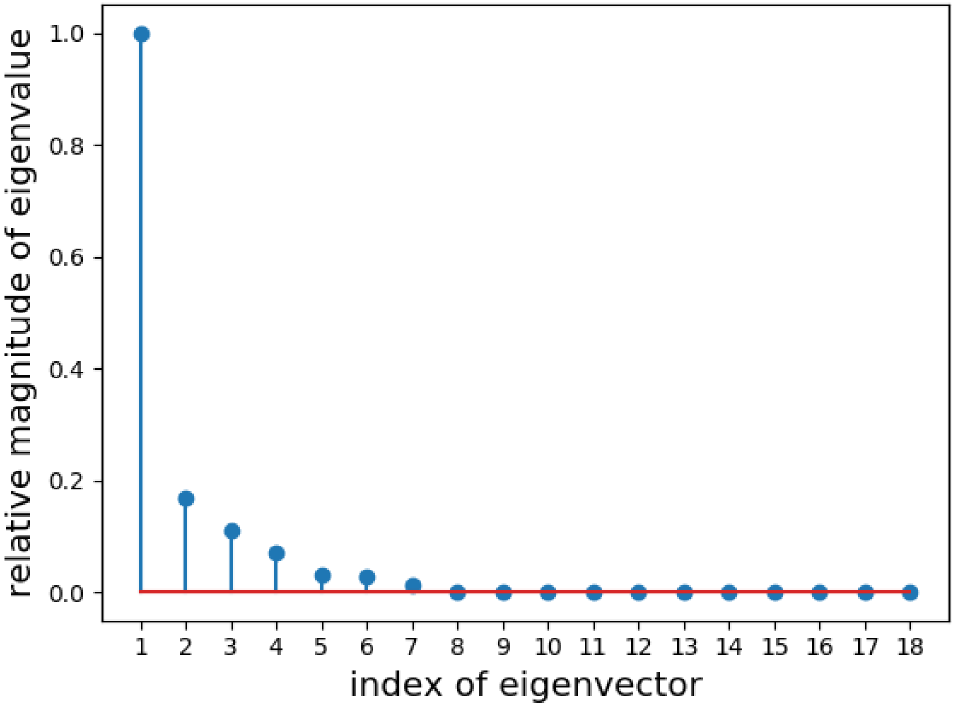

Figure 6. Absolute relative eigenvalues for all classes.

Once the model has been chosen—in this case describing the signal in terms of the first two eigenvectors—it is necessary to represent the signal in terms of the new model. This is done by projecting the original signal vectors onto the new basis given by the strongest two eigenvectors.

In Figure 7, all four classes of features, after re-representation, can be seen. With respect to the original features from the wavelet transform, there is much better class separation here. Our goal is to build a detector and classifier to determine which action was performed. For this reason, a measure of improvement is to increase the geometric separation of the classes with respect to some basis.

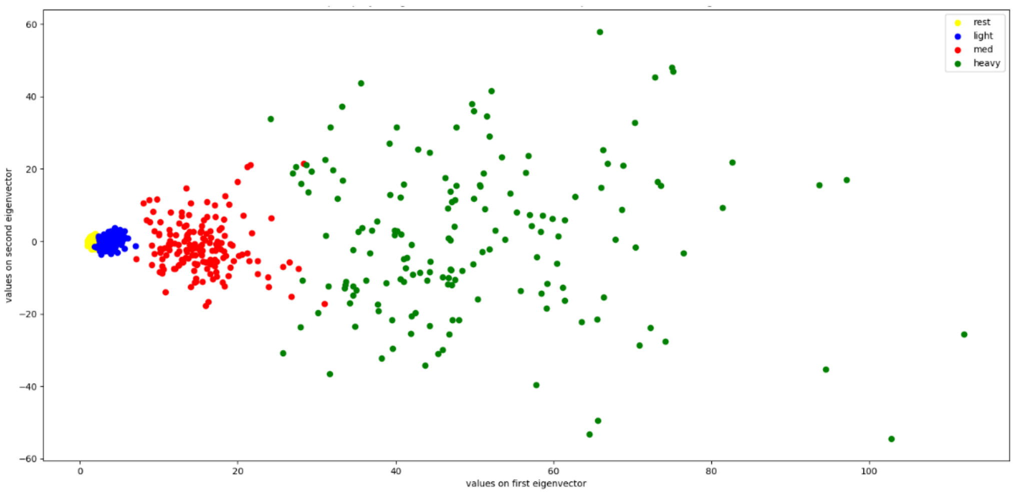

Figure 7. Features represented by the new model, using the first two eigenvectors.

Figure 8 is a magnified version of Figure 7. Here, it can be seen that the light and medium classes show good separation. It will be shown later that this just happens to be a special case of well-behaved data.

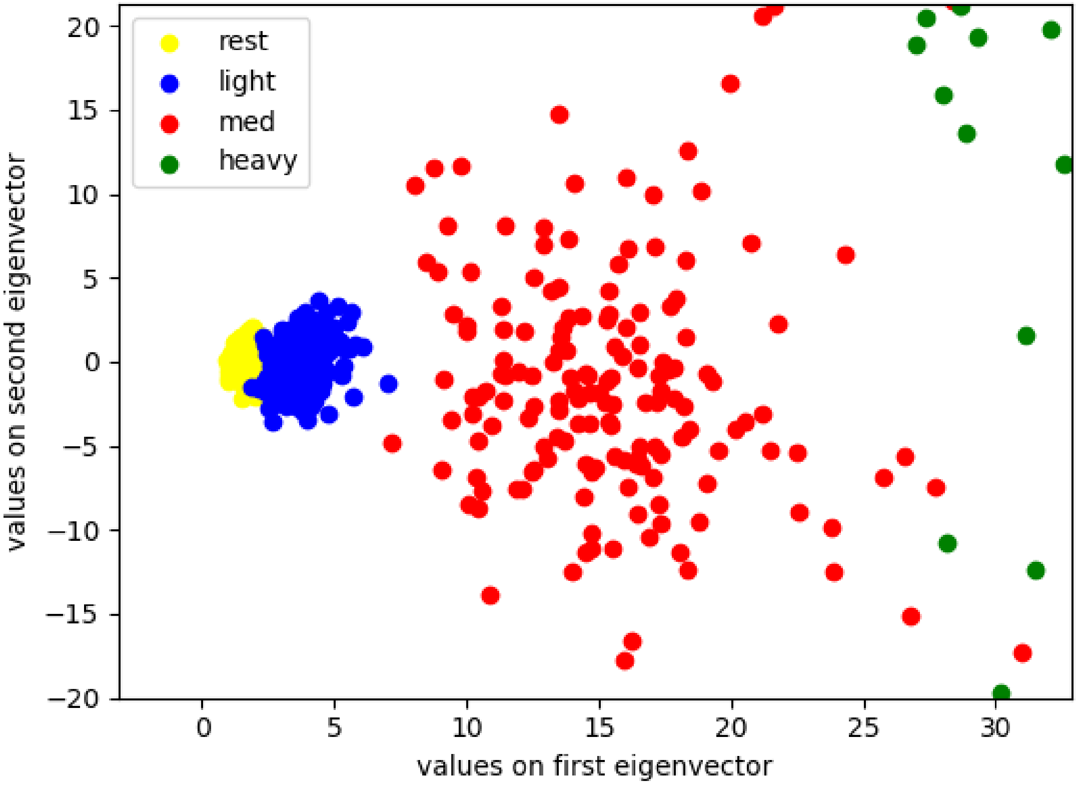

Figure 8. Magnified view of transform of rest, light, and medium class features.

Detector and Classifier

Data from the subject was collected, features extracted, and a transform to optimize the class separability was implemented. The next step was to build a statistical model for the data in the classes, to detect and classify actions. As stated previously, a histogram can be used as an estimation of a PDF. Now that the classes show promise, they will result in class separability upon transform, but the development of a detector is of practical concern.

There are a couple of ways to develop a detector. If the statistical model of the data is known a priori, then a likelihood ratio test can be conducted. This can be shown through the Neyman-Person lemma to be a most powerful test (MPT). The Neyman-Pearson detector is based on a likelihood ratio, and is a test with a constraint on a type of error. In the case of Neyman-Pearson, the constraint on the type of error is with respect to the probability of false alarm. If it is assumed that a detector must have a probability of false alarm less than some value α, then the detector is as follows (Kay, 1993):

$$\emptyset(x) = \begin{cases} 1, & \dfrac{f_1(x)}{f_0(x)} > t \\ 0, & \dfrac{f_1(x)}{f_0(x)} < t \end{cases} \tag{18}$$

where $\emptyset(x)$ is the detector, $\frac{f_1(x)}{f_0(x)}$ is the likelihood ratio, and "t" is some threshold that would satisfy the probability of false alarm criteria.

For example, consider two Gaussian distributions that are identical, except for their means. In this case, take the value for "t" to be "1." This simply means that the detector will decide for Hypothesis 1 when the PDF of the first hypothesis is greater than the null hypothesis by a certain amount. This is an intuitive way to examine this type of detector. In some cases, inequalities can be manipulated to get down to a test statistic with a known distribution (i.e. Gaussian). In the case of a Gaussian distribution, a Q-function can be used to determine the appropriate value for "t." The most important piece of information from this type of tool is that some threshold with regard to the random variable (r.v.) will be determined. This means that, given realizations of some r.v., the threshold value indicates at which point in the data set the regions for H_0 and H_1 will be defined.

A useful way to look at the total performance of a detector is to gauge the probability of detection against the probability of false alarm. This is known as a receiver operating curve (RoC). An RoC is a plot of probability of detection against the probability of false alarm. It is useful to have this information to determine whether a detector can be realized with the desired requirements. It can also be used to determine whether one detector is better than another. These two detectors would examine the same hypotheses, but could look at them using a different set of features relating to the two decisions to be made. One detector could also require a larger number of samples than another to obtain the desirable probability of false alarm and detection. In practical cases of Neyman-Pearson detectors, it is common to increase the number of samples required in order to make the detector perform as desired.

Another type of detector is the Bayes risk detector. In the Bayes risk detector, the threshold value of the Neyman-Pearson-style detector is replaced by the a priori probabilities of the hypotheses involved, and is weighted by the associated costs of making an error. The definition of Bayes risk is as follows (Kay, 1993):

$$\emptyset(x) = \begin{cases} 1, & \dfrac{f_1(x)}{f_0(x)} > \dfrac{(C_{10} - C_{00})P[H_0]}{(C_{01} - C_{11})P[H_1]} \\ & 0, \; otherwise \end{cases} \tag{19}$$

where C_{ij} is the cost of making a mistake. Notice that, if the cost of making the right decision is 0, and the cost of making the wrong decision is 1, then the Bayes risk detector minimizes the probability of making an error. In this case, the detector is just known as a Bayes detector, and can be thought of as a special case of the Bayes risk detector.

A detector decides between two hypotheses (which is the only concern when the question is between action or inaction), but what about where something must be classified? In this case, a classifier must be designed. A classifier has a function similar to the function of the detector. The key difference is, with the classifier, there are multiple possible hypotheses for a particular realization. In this case, it is not enough to just select one type of error to design the threshold. The designer should be concerned with the probability of making an error in general. Suppose there are three hypotheses to classify, based on realizations of some feature. In this case, the designer would be concerned with not only selecting H_{i+1} when H_i is true, but also the error in selecting H_{i-1} when H_i is true, and vice-versa. This is also known as multiple hypothesis testing, and the detector commonly used to decide between multiple hypotheses is the Bayes multiple hypothesis risk detector (Kay, 1993).

The Bayes multiple hypothesis detector is what is used in this research to classify the transformed features as members of a specific class. The Bayes multiple hypothesis detector is defined for an N-hypotheses system as:

$$Decide\ for\ H_i\ if\ \sum_j C_{ij}\, f_j(\underline{x}) < \sum_j C_{kj}\, f_j(\underline{x}),\ \ \forall\, k \qquad (20)$$

where C is the cost matrix, associated with the cost of making a type of error (Kay, 1993). In many practical cases, this can be thought of as a "1's" matrix, with a 0 diagonal. What this means is that the cost of making the correct decision is 0. The cost of making any other decision, an incorrect classification, will be 1. It is also a reasonable assumption that in cases were the system does not have a bias towards a class, the a priori probabilities will be equal. With this assumption, the multiple hypothesis Bayes detector can be described as follows:

$$Decide\ for\ H_i\ if\ f_i(\underline{x}) > f_k(\underline{x}), \qquad \forall\ k \tag{21}$$

This formulation was utilized in this research. The justification for the above model is that no error is more or less significant than any other error, and the probability of being under a given class is equivalent among the classes. Note that this could be changed if a priori information was shown to favor a specific class. Further customizability of the method could be implemented by biasing the a priori probabilities and/or adjusting the cost matrix. The reason for doing this would be to accommodate users who are more likely to be under a certain class.

Model Selection

In the above discussion on detectors and classifiers, $f_i(x)$ was used to represent some probability density function (PDF) under the hypothesis H_i. To have an accurate classifier, model selection is an important consideration. To examine the types of statistical models that best fit some collection of data, a histogram can be utilized. In this research, the features along the most significant eigenvectors are treated as random variables. Histograms of realizations of these random variables are presented in the following figures.

Notice in Figure 9 that, along the first eigenvector, the features appear to be approximately Gaussian. This is an important observation, because optimal detectors for these actions can be found explicitly. By making an educated assumption as to the likely distribution, the most powerful test statistic is able to be computed to give an accurate estimate of the best threshold for a given error requirement. Note that the class of inaction is identical to the resting class, and the class for action, is the concatenation of the light, medium, and heavy classes. This was done to estimate the model of the detector. Once a detector is built, the action classes are looked at closely to classify which action occurred.

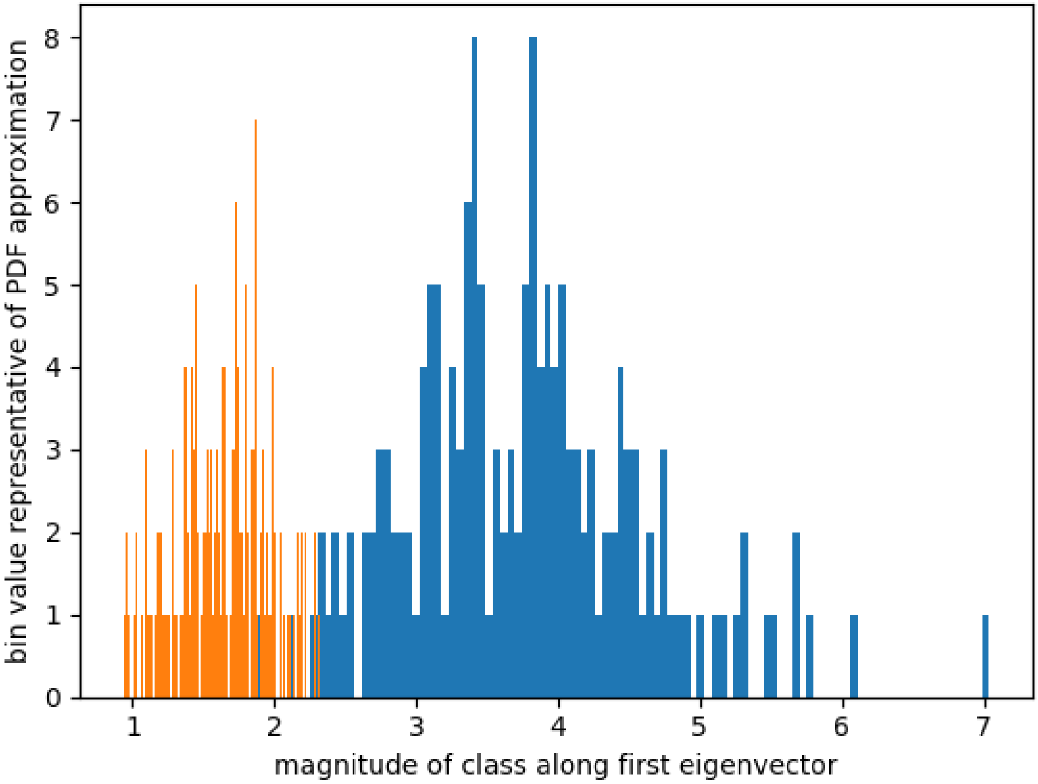

Figure 9. Histogram for action and inaction classes along first eigenvector. Number of bins 100.

In Figure 10, a histogram is presented using data from all 4 classes. Notice that, along the first eigenvector, the features appear to produce Gaussian-like distributions with each distribution having a different mean and variance. While there is class overlap, it is much improved from distributions of the untreated class features themselves (as shown in Figures 2 to

5). This is where the inherent advantage of an eigenvector feature space representation becomes apparent: it is capable of re-representing the orthogonal signal space in a way that is custom to the input data. This characteristic of the eigenvectors allows a signal to be projected onto its most powerful dimension, increasing class separation, and resulting in better detector performance.

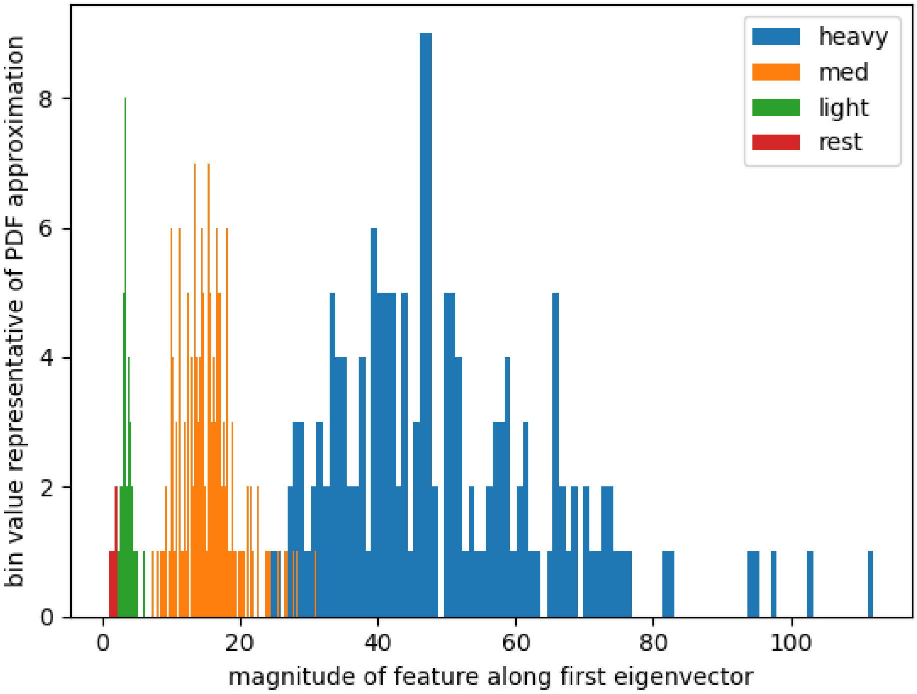

Figure 10. Histograms of all classes along first eigen vector. Number of bins 100. With 160 samples for each class.

As stated earlier, the second eigenvector was also taken to represent the signal space. This was done because there was still some significant energy in the second eigenvalue, though it was small relative to the first eigenvalue. For the sake of comparison, the distributions of the classes are presented with respect to the second eigenvector projections in Figure 11.

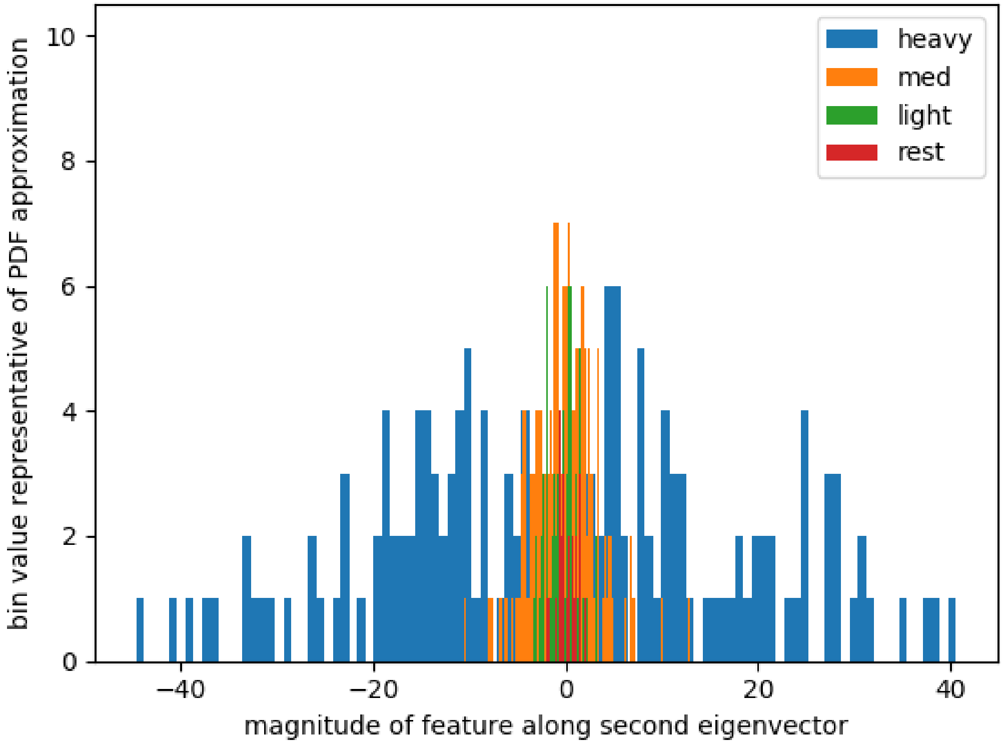

Figure 11. Histogram along second eigen vector for all classes. Number of bins 100.

Notice that the variance is similar, but the means are much closer than in the case of the first eigenvector. This again highlights what the eigenvector representation is doing. There is more energy in the signal along the first eigenvector than in the second eigenvector. This is because there is less energy in the signal along the second eigenvector, but the noise is the same. In other words, the signal-to-noise ratio (SNR) along the first eigenvector is much larger than the SNR along the second, which leads to an optimal condition for a detector along the first eigenvector.

Validation of Model

There are many techniques to validate a model's accuracy. One such technique is known as cross validation. Two types of cross validation are "leave-one-out" and "leave-p-out" cross validation. In this research, leave-p-out cross validation was used. Due to the impractical number of combinations necessary in leave-p-out cross validation, a compromise was selected. In this thesis, all the data comprising the classes were put into sets, with each set representing one of the four possible classes: resting, light flex, medium flex, and heavy flex. The first test took just one subset of the main class set and used it as training data. The complements of the training set were used to test the classifier. It was noticed that the statistics on the probability of error were too spread, and one round of this was not enough to estimate the classifier's probability of error. After some experimentation, run counts of 100 iterations were found to yield acceptable statistics on the probability of error. It was ensured that the training and testing sets of each iteration were different in order, to not corrupt the results of the test.

CHAPTER 4

RESULTS

A function to generate both wavelet-eigen features and DSTFT features was developed and used to generate both training and testing data sets. The training data sets were chosen at random, and for each training set its complement was used as the testing set. This was done so that the testing would not use any sample that was used during training, which would have given an inaccurate measure of the detector and the classifier. This is similar to leave-p-out cross validation. In this case, four sets of data for each class were left out for testing, and the complement of the set was used to train. The process of selecting separate training and testing data sets was completed over 20 iterations. The purpose was to gather statistics on the P[error] in order to validate the model.

Wavelet-Eigen Detector Analysis

To determine the threshold for the detector, a probability of false alarm (PFA) requirement was set and a threshold was found using the training data. Then, a Monte Carlo simulation was conducted with the testing data to evaluate the threshold and determine the corresponding probability of detection (PD).

In order to create the RoC, the PFA requirement on the training set was changed incrementally from 100% to 0% and a threshold was computed at each increment. A Monte Carlo simulation on the testing set was again run at each threshold to determine the PFA and PD for the given threshold. This process was completed over 20 combinations of possible training and testing sets. The detector was implemented and used to generate the RoC, as shown in Figure 12.

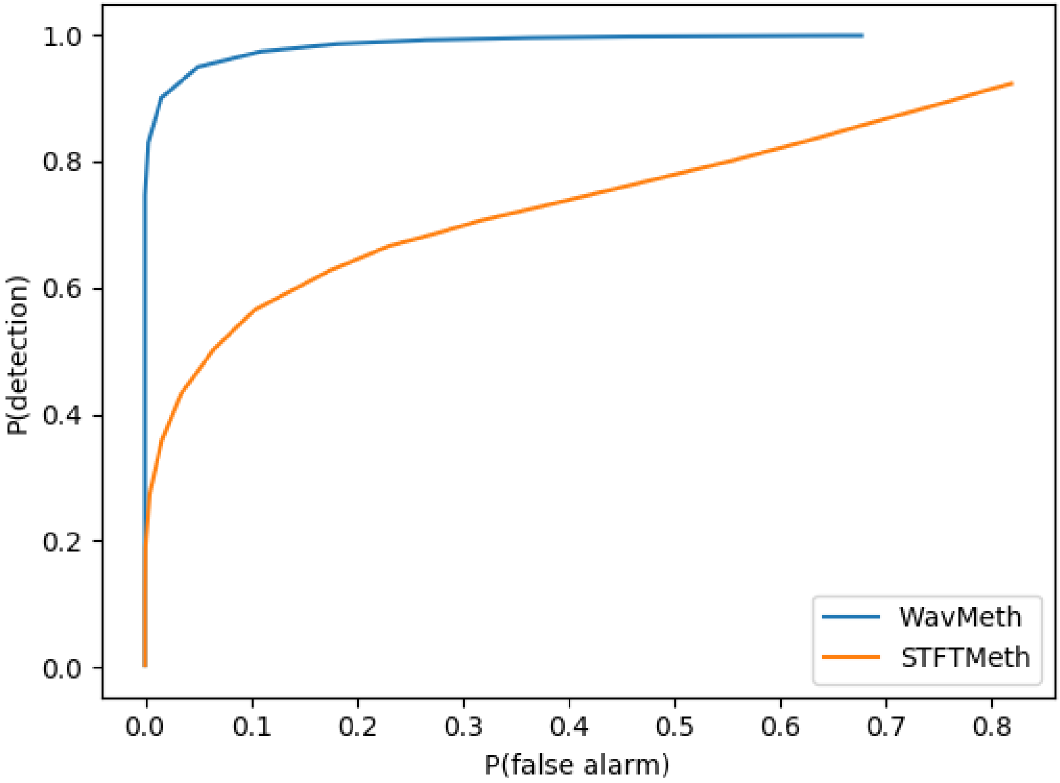

Figure 12. RoC for wavelet-eigen and DSTFT methods.

From Figure 12, it can be seen that for any given probability of false alarm requirement, the probability of detection with the wavelet-eigen method will always be higher than in the DSTFT case. It can be seen that the wavelet features treated with the eigenvalue decomposition performed better than the untreated features extracted from the DSTFT. The purpose of the detector is to decide if an action occurred. Once an action is detected it will be classified.

Wavelet-Eigen Classifier Analysis

As before with the detector, the training and testing sets were established and statistics on the training data were computed. The means and standard deviations on the training data allowed class data ranges to be selected. This was completed based on a Gaussian model. The testing set was then used to evaluate the accuracy of the trained data ranges. This was completed over 100 iterations where new testing and training sets were being produced to evaluate statistics on the P[error]. Once the P[error] statistics were found, a confusion matrix could be created. This confusion matrix was created over just one random selected training and complement testing set. The performance of the classifier using wavelet features is shown in Figure 13, and Table 1, as follows:

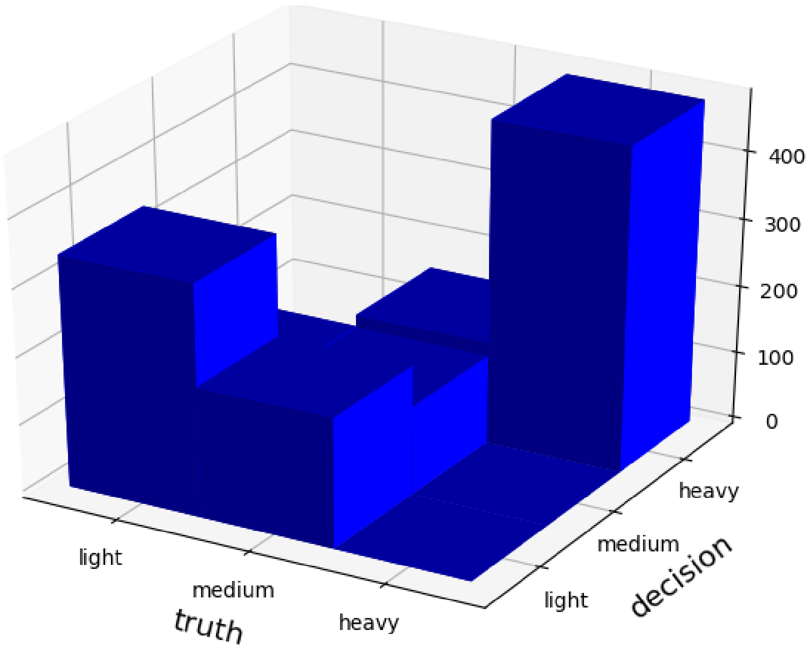

Figure 13. Confusion chart using all 3 action classes for the wavelet-eigen method.

Table 1 Type of error and its occurrence for the confusion chart in Figure 13.

DECISION **TRUTH**	**LIGHT**	**MEDIUM**	**HEAVY**
LIGHT	23.9%	9.4%	0.0%
MEDIUM	13.5%	9.3%	10.6%
HEAVY	0.0%	0.0%	33.3%

Notice in this case that the classifier was more likely to take the light class as the decision under the condition where the medium class was the truth, and vice versa.

DSTFT Classifier Analysis

To illustrate the advantage of the wavelet-eigen features, a comparison was conducted with an untreated sinusoidal basis. Sinusoidal frequency components are used in the literature. However, when performing the DSTFT, the original features were not orthogonal. Orthogonal frequency components were chosen; specifically, 150 and 300 Hz. The classifier was built on the features representative of the 150 Hz component, as shown earlier. Figure 14 and Table 2 present the confusion chart and data for the DSTFT feature based classifier.

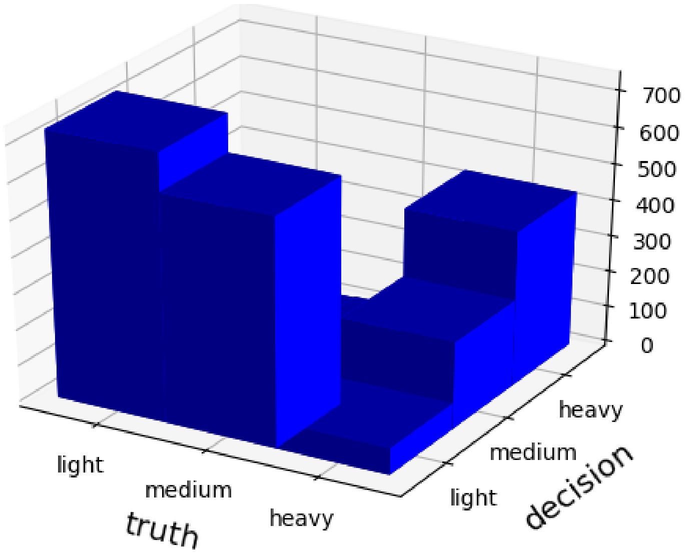

Figure 14. DSTFT-based feature classifier confusion chart.

Table 2. Type of error and its occurrence for the confusion chart in Figure 14.

DECISION **TRUTH**	**LIGHT**	**MEDIUM**	**HEAVY**
LIGHT	32.7%	0.7%	0.0%
MEDIUM	28.0%	5.2%	0.1%
HEAVY	3.3%	10.8%	19.2%

Comparing Figure 14 with Figure 13, it is possible to observe that more errors were made using the DSTFT-based classifier than in the wavelet-eigen-based classifier. Also, notice that those errors are of the same type. Regardless of the method used, the classifier was most confused by the medium flex class. As was evident previously with the other set of features from the wavelet-eigen analysis, this was due to the overlap in the feature space between the medium flex class and light flex class.

A statistical comparison of the two methods with respect to their mean, probability of error, and the standard deviation of the probability of error is presented in Table 3 below.

Table 3. Statistical comparison of Wavelet-Eigen and the DSTFT methods and classifier results over 100 iterations.

P[ERROR]	**MEAN**	**STAN. DEV.**
WAVELET-EIGEN	0.31	0.04
DSTFT	0.40	0.05

Notice that the standard deviations are low in comparison to the means. The mean probability of error for the wavelet-eigen method was ~0.31, whereas for the STFT method it was ~0.40. These are still both relatively high allowances. In an effort to bring the probability of error down to more acceptable levels, the classes were combined.

Combined Class Analysis

Recall that when classifying among all classes, the most common error was between the light and medium classes. This was due to the overlap in features between the medium and the light flex classes. To create a classifier with a better probability of error, it was decided to combine them into one class. The heavy class remained the same. Now, instead of three possible classes, with three distributions, there are only two possible distributions to choose from, the one containing the light and medium flex data, and the one containing the heavy flex data. Figure 15 presents a plot of the confusion chart, using the wavelet-eigen method. This and Table 4 illustrate deciding between the light-medium class, and the heavy class.

Figure 15. Wavelet-eigen classifier result with combined features.

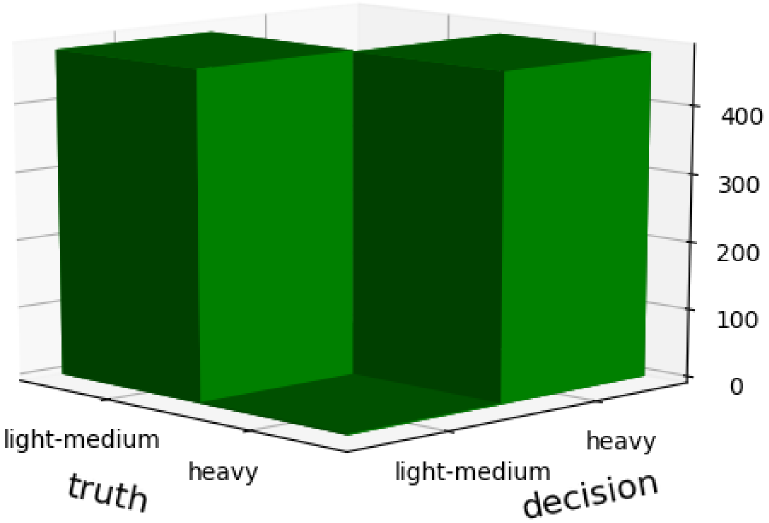

Table 4. Type of error and its occurrence for the confusion chart in Figure 16

DECISION TRUTH	**LIGHT-MEDIUM**	**HEAVY**
LIGHT-MEDIUM	50.0%	0.0%
HEAVY	0.2%	49.8%

Notice that compared with the previous result of Figure 13, the number of errors made between the light-medium and heavy classes are substantially decreased when using the wavelet-eigen method. This combination of classes was also performed with the DSTFT features. Figure 16 and Table 5 present the results of the combined features on the classifier.

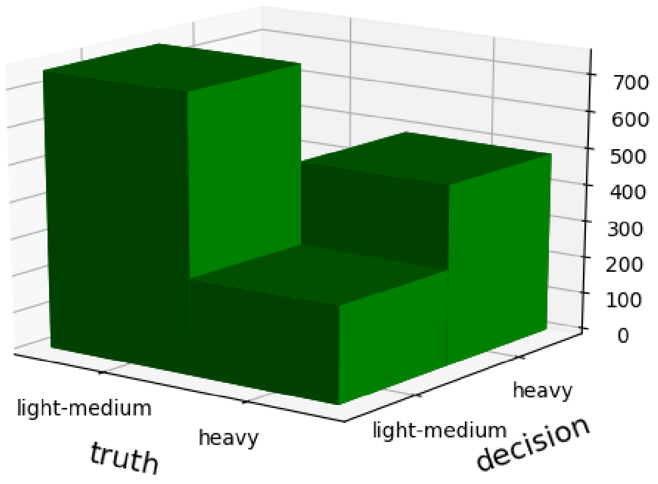

Figure 16. Confusion chart for DSTFT method, with combined features.

Table 5. Type of error and its occurrence for the confusion chart in Figure 17.

DECISION TRUTH	LIGHT-MEDIUM	HEAVY
LIGHT-MEDIUM	50.0%	0.0%
HEAVY	17.5%	32.5%

Notice that in Figure 16, the number of errors is higher than in the combined features case when using the wavelet-eigen method. There is a greater chance that the classifier makes the incorrect decision, but it is still improved from the STFT case shown in Figure 14, which uses all three classes. Table 6 shows the new probability of error statistics, with the combined light and medium classes.

Table 6. Statistical comparison of combined wavelet-eigen and the DSTFT methods and classifier results over 100 iterations.

P[ERROR]	MEAN	STAN. DEV.
WAVELET-EIGEN	0.01	0.01
DSTFT	0.13	0.05

Recall from Table 3 that the statistics on the probability of error when trying to classify among three classes had a mean probability of error for the wavelet-eigen method of ~0.31. In this case, with the light and medium classes combined, the mean probability of error decreased to 0.01+/-0.01 , which is an order of magnitude better than in the prior wavelet-eigen case. The STFT method statistics also improved, but they still have a relatively large probability of error of 0.13 +/-0.05.

To summarize, when considering classification between two actions, the classifier based on the wavelet-eigen features was able to classify the desired action correctly with an average probability of error of 0.01, which is lower than the average probability of error of 0.13 obtained with STFT features.

Well-Behaved Data Result

It is worth noting that a probability of error of 0.31 +/- 0.04 is relatively high. Notice further that the errors are not equal. For instance, the classifier is more likely to make an error deciding for the light class when the truth is the medium class. To understand the performance of the classifier better, the iterations were run, and a plot for the transformed features is shown in Figures 17 and 18. Notice in these figures that the medium class looks to be clustered over the light classes data by comparison with Figure 19.

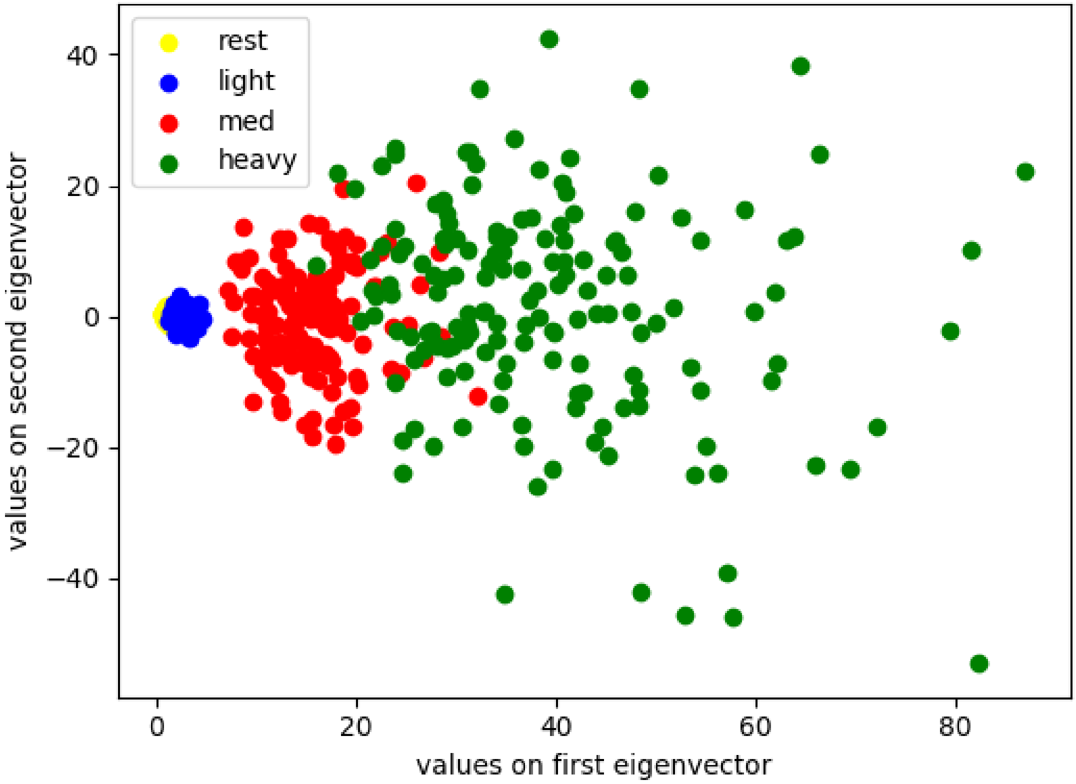

Figure 17: Well-behaved data after wavelet-eigen representation on the first two eigenvectors.

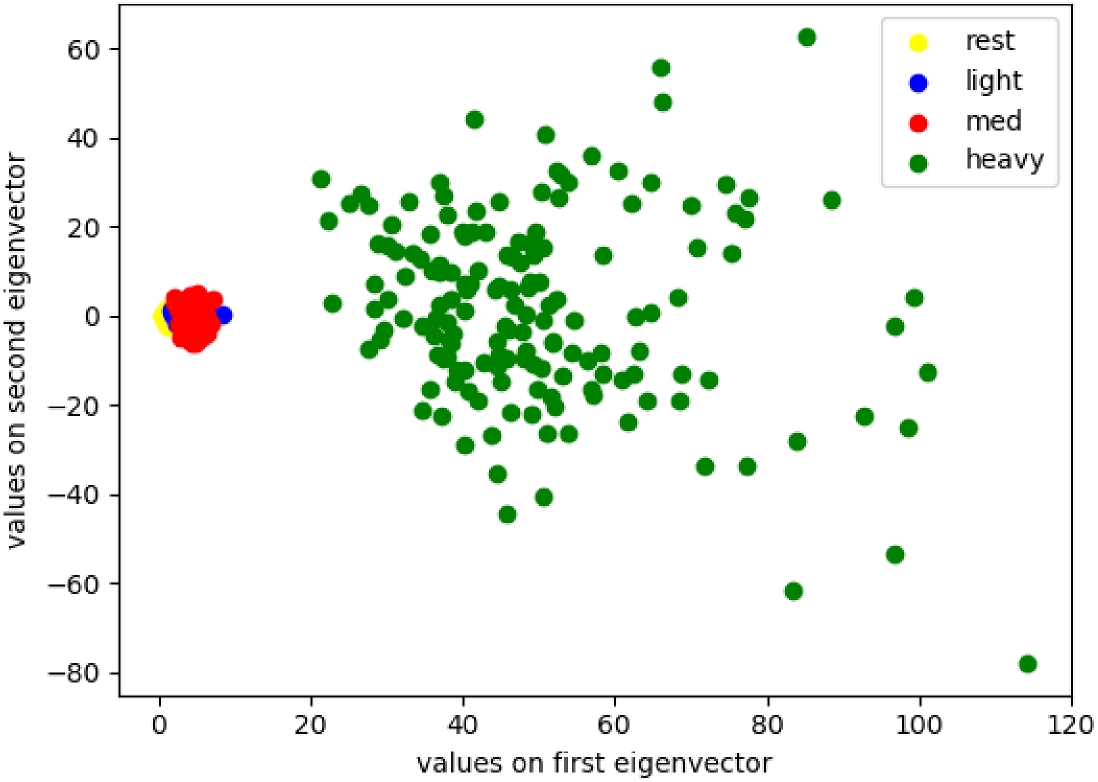

Figure 18. Scatter plot of first and second eigenvector features.

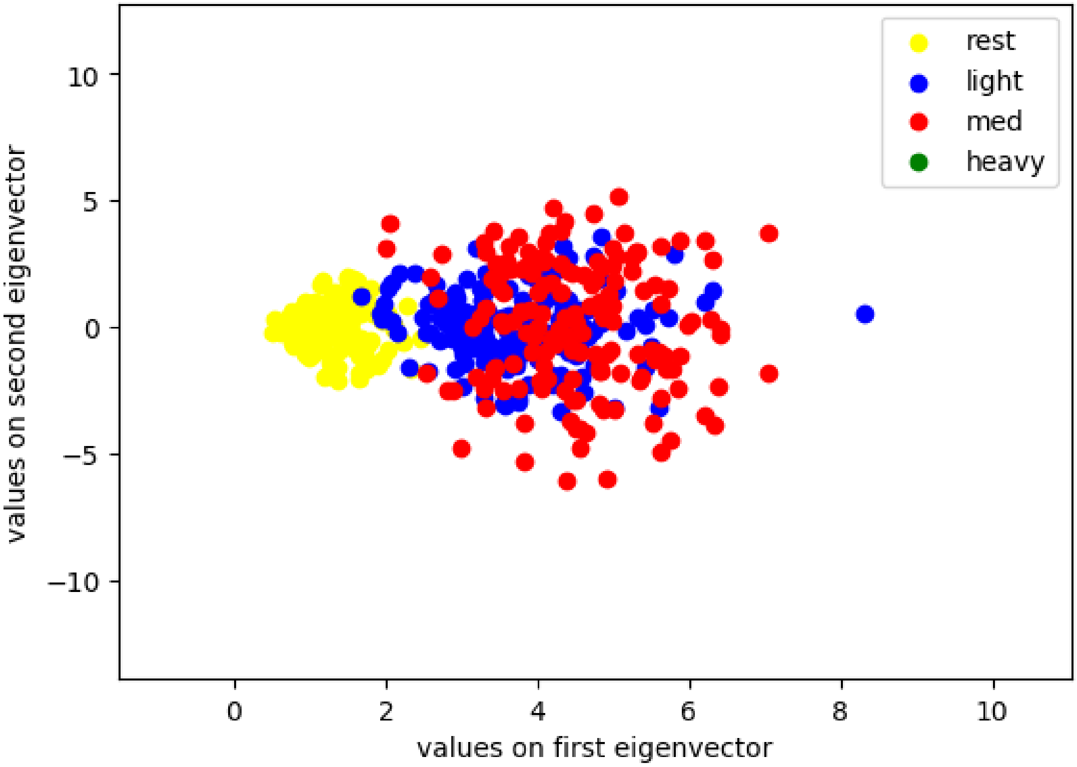

Figure 19. The resting, light, and medium classes from Figure 18.

In Figure 19, it can be seen that the medium class is over-lapping the light class. This explains the high error when deciding between these two classes, as seen in the confusion charts presented in Tables 1 and 2. From the collected data the medium class appears to be more likely to drift into the region of the light data than the converse. This was the motivation behind combining the light and medium classes.

The well-behaved data has a very clear spatial separation on the first eigenvector. The resulting confusion chart has a very low probability of error (Figure 20 and Table 7). This is the result we could have expected from an objective test for collecting data.

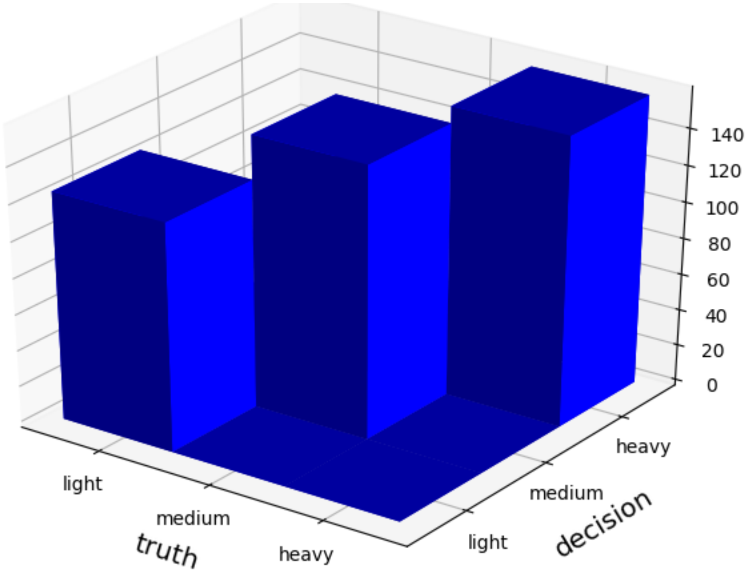

Figure 20. Confusion chart for well-behaved data using wavelet-eigen features.

Table 7

Type of error and its occurrence for the confusion chart in Figure 20.

DECISION **TRUTH**	**LIGHT**	**MEDIUM**	**HEAVY**
LIGHT	26.4%	7.0%	0.0%
MEDIUM	0.0%	31.4%	1.9%
HEAVY	0.0%	0.0%	33.3%

As before, the observation that the implemented classifier has a low probability of error indicates that the features showed good spatial separation in the feature space. In the plot of Figure 17, this can be seen.

Notice in Figure 17 the medium class data is not over-lapping with the light flexing data as it is in Figure 18. For this reason, the classifier does well with making accurate decisions.

Figure 17 also shows why the first eigenvector is the most powerful in classifying the features. Notice that along the first eigenvector distributions are spread out with different means, but along the second eigenvector the data has nearly the same mean. While both vectors could be used to build a classifier, the first eigenvector was chosen because it offered the best feature separation.

CHAPTER 5

DISCUSSION AND CONCLUSION

Discussion

This thesis is not making a claim that the wavelet method is superior to the DSTFT method on its own. Given the same eigenvector treatment that the wavelet decomposition goes through, the results for the DSTFT based classifier could be just as good. The reason for the DSTFT method being presented is to show the difference, using a current method, of the possible advantage in the eigenvector representation of a feature set.

In the customizable approach, a person's signal is decomposed into some basis. In this case, the basis is produced by a wavelet transform, using the Haar wavelet. Once the basis is established, features are created from the windowed-averaged coefficients of the wavelet and scaling coefficients and a matrix is built. This matrix is representative of that person's signal. Subsequently, the strongest eigenvector is used to increase that person's class separation. The key idea here is that the method "finds" a unique, best combination of original features for the given signals for the given person. This is where the key difference lies. The assumption is that two people will have different signals, and when the matrix is built, the eigenvectors of any two peoples' feature matrices will be different.

It should be mentioned that these are preliminary results, and further study is needed to determine whether the method proposed in this thesis is able to provide better action classification in a large population of subjects. If, for example, a large study was conducted and it was found that by using the same wavelet transform to generate features most of the eigenvectors found were the same, this would suggest only using that vector to generate features and there would not be a need for the customizable approach. If the eigenvectors found are different among individuals, then the customizable approach is of practical consideration. Over multiple sets of the collected data the first eigenvectors were nearly the same.

Data was not gathered from an upper extremity amputee in this study. It would be of interest to study this particular population to explore whether different amputations may result in different signals. In fact, this may prove to be the real power of a method such as this, as many amputees experience different levels of muscle atrophy, or in some cases do not have the same muscular topology as an able-bodied person.

Conclusion

EMG is an inherently random signal that exhibits non-stationarity. This research looked to treat this signal using an eigenvector space representation to determine which signal vector had the most energy present. This was of interest for detecting and further classifying that action. A broad set of orthogonal features were generated by using a wavelet decomposition.

The EMG data was received from a single-channel sensor located over the brachioradialis muscle. The reason a single channel was used was to better gauge the effectiveness of the wavelet eigen-analysis approach along a single EMG signal vector. If multiple channels had been used, those separate channels could have potentially increased signal separation and given better results.

These results show why thresholding the EMG signal could still be error prone, even if the threshold is just to distinguish between three different actions. In the case of this research, it was noticed that even after treatment by wavelet decomposition and an eigenvector representation of the signal space, there was still not enough energy in the treated signal to decrease the probability of error to a desirable value. The probability of error was not decreased by much more than approximately 31%. The type of error that occurred was biased towards an interference between two classes. For that reason, the two classes causing the most interference were combined. When this was done, the total probability of error decreased to approximately 1%.

It could be argued that if the data had been collected in a more systematic and measurable way, then the results for the three classes could have been more easily separated by features For instance, if the data had been collected with different action classes defined based on a fixed loading of the muscle, then there may have been an improvement in the probability of error for the three classes. However, it may be important to consider data measurement with respect to practical field implementations. To illustrate this point, consider users of prosthetic hands calibrating the hand to their EMG signal using an objective measure of effort, in which the user of the hand must exert a specified force on an object to record the three classes. Although the measurements would be objective and everyone who calibrates to the hand would produce the same result, such an objective calibration would not be suitable to other amputees, such as those with exhaustion or muscle fatigue. The problem is that the maximum effort from such users may only be enough for them to get into the objective class measurement of a medium flex, while their intent was to produce a signal for a heavy flex action. The research here suggests that a subjective measure of the EMG signal could be useful in the case of practical classification.

This research also evaluated the idea of using the wavelet-eigen-analysis to generate custom features and hence custom eigenvectors for those features (the goal being to build detectors and classifiers on those features). The idea here is that different amputees may build different eigenvectors. To use just any feature may not result in the best test statistic for a detector or classifier.

An approach where the amputee's EMG signal samples are used to discover the optimal feature dimension may result in more consistency and reliability from user to user. As noted in the literature review, one of the significant problems for amputees is acceptance of a device. If the device is not reliable for the user, it will be rejected. The rejection rate specified in the literature review was 23% for adult patients and was higher yet for children (Roche et al., 2014). Because of this, creating customizable features could reduce rejection rates.

The signals used in the research were from only one person who was not an arm amputee. The demonstration of this idea as an improvement over existing techniques for prosthetic hands has not been completed here. A large, multi-participant study across age, race, sex, and type of amputation would be needed to validate any of the above claims.

This research was undertaken to investigate the possibility of using a wavelet-eigen approach to create custom features with subjective data classes. The benefit of using such a method, where features of the signal are used to build new features, lies in how the new features are constructed. By using the wavelet-eigen approach, new features can be extracted representing a combination of the most energetic original features. This allows for the analysis to produce new features of a signal that are much more distinguished than if features had been chosen directly.

In conclusion, this thesis has provided evidence that signal processing techniques can be practically implemented in prosthetic devices to improve upon classification accuracy. Further research into validation of the customizable approach should be undertaken in order to validate the technique.

Future Work

A more expansive patient study using subjective data should be completed to confirm the results presented in this thesis and to confirm that customizable detectors can indeed offer additional reliability and precision. In this study, data was gathered from the principal researcher and the results are representative of the researchers recorded data.

As mentioned, the eigenvector is oriented along the direction of most energy. This is the property that allowed for the classifier to be realized with improved performance. In this research, it was shown that the probability of error decreased when the light and medium classes were combined. One interpretation of this improvement could come from looking at the distributions for each class. As the overlap between classes increase, this could be interpreted as that class decreasing in energy. In a real system, this could occur for a variety of reasons, such as sensor mobility away from the muscle, muscle fatigue, or the sensor could be impacted by dead cell build-up. A person’s hygiene could also affect the system performance. In all cases, these reasons might be representable by an increase in the overlap among classes.

A technique to allow the system to dynamically adjust to maintain some requirement on reliability would be of interest. This system could dynamically adjust the number of classes based on the SNR, similar to a communications system. In detail, the system would decrease the number of available classes if the SNR decreased and would increase the number of available classes if the SNR increased.

This same concept was implemented in this thesis in order to improve upon the classification accuracy. The P[error] for the case of all three classes was too high, or its SNR was too low. That motivated the combination of classes, which significantly reduced the P[error] of the classifier. The proposed system would do the same, except it would decide to do the combination automatically.

It has been noticed during the research here that the first eigenvector shows up as being the strongest vector for representing the space. It is worth investigating why this may be, or if there is a physiological reason for this vector showing up consistently. More research should be done in order to determine if there is some scientific reason behind this result.

REFERENCES

Burrus, C., Gopinath, R. & Guo, H. (1997). *Introductory to wavelets and wavelet transforms: a primer.* Upper Saddle River, NJ: Prentice Hall.

Kay, S. M. (1993). *Fundamentals of Statistical Signal Processing Detection Theory.* Upper Saddle River, NJ: Prentice Hall.

Lay, D. C. (2012). *Linear algebra and its applications*. Boston: Addison-Wesley.

Pattichis, C., & Pattichis, M. (1999). Time-scale analysis of motor unit action potentials. *IEEE Transactions on Biomedical Engineering,46*(11), pp. 1320-1329. doi:10.1109/10.797992

Phinyomark, A., Limsakul, C., & Phukpattaranont, P. (2009). A Novel Feature Extraction for Robust EMG Pattern Recognition. *Journal of Computing, 1*(1), pp. 71-80. https://sites.google.com/SITE/JOURNALOFCOMPUTING/

Reaz, M. B., Hussain, M. S., & Mohd-Yasin, F. (2006). Techniques of EMG signal analysis: Detection, processing, classification, and applications (Correction). *Biological Procedures Online,8*(1), pp. 163-163. doi:10.1251/bpo124

Roche, A. D., Rehbaum, H., Farina, D., & Aszmann, O. C. (2014). Prosthetic Myoelectric Control Strategies: A Clinical Perspective. *Current Surgery Reports,2*(3). doi:10.1007/s40137-013-0044-8

Webster, J. G., & Clark, J. W. (2010). *Medical instrumentation: Application and design*. Hoboken (New Jersey): Wiley.