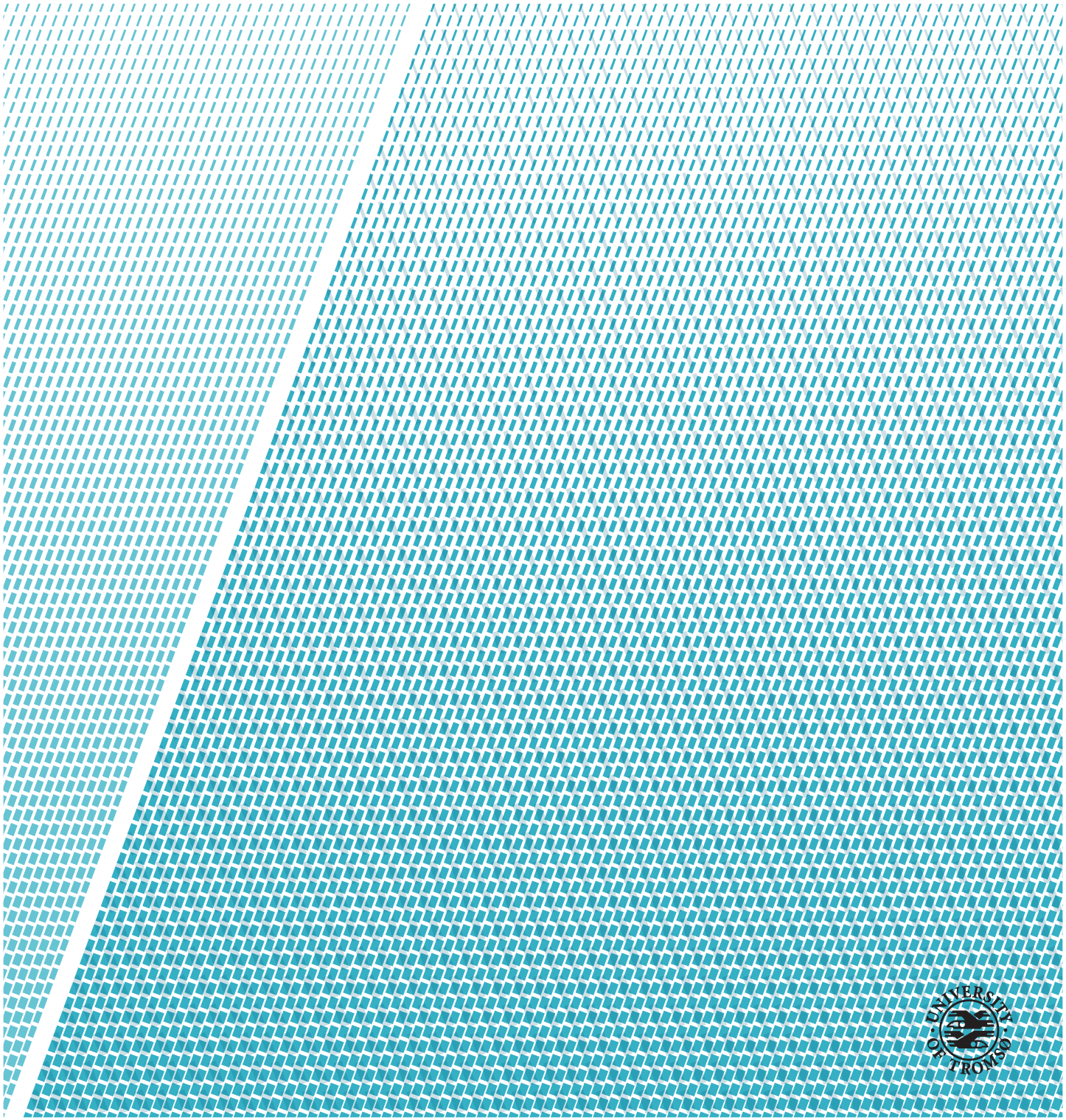
UiT
THE ARCTIC
UNIVERSITY
OF NORWAY

Faculty of Science and Technology
Department of Physics and Technology

Analyzing Behavioral Biometrics of Handwriting Using Myo Gesture Control Armband

Brynjulv Tveit

FYS-3941 Master's Thesis in Applied Physics and Mathematics



“Til lags åt alle kan ingen gjera; det er no gamalt og vil so vera”
–Ivar Aasen

Abstract

Through the last few decades, computer technology has gradually merged into our everyday lives. Computers and sensors are embedded in an increasing amount of household items, enabling us to monitor and remotely control our connected devices from apps on our smartphones. The technology interfaces are also evolving along with new technologies. Among the up and coming digital interfaces are wearable technology. The Myo gesture control armband (GCA) is an example of tools which aims to make the communication from computer to human more seamless and intuitive. The Myo GCA is a multi sensor armband containing 8 surface electromyography sensors which measure electrical activity originating from skeletal muscles in the upper forearm. It is also equipped with a 9-axis inertial measurement unit which can provide information on spatial arm movements of the users. Together these sensors enable its user to pass 6 configurable commands to a smart phone or Blue-tooth connected computer. In this thesis we explore the Myo armbands potential as a multi sensor for handwriting recognition. Data are sampled and manually extracted through a cumbersome time consuming process, using recorded video as a reference to the sampled Myo data. The subjects are given the task of writing 10 repetitions each, of the four capital letters: E, L, O, and R. A strong positive correlation between same class letters within subjects has been proven in all of the four sensor types, where the orientation data yields the highest correlation coefficient values, while the sEMG data yields the lowest. Statistical similarity between same class letters has been found through singular value decomposition, where again orientation data yields the highest values, while sEMG scores the lowest of all sensor types. In an attempt to cross subject classification though k-NN, with $k = 1$, $k = 3$, and $k = 5$, the 1-NN classifier yields a minimum success rate of 58% across the four letters. This is considerably better that what we would expect from a random assignment of letter classes. In the last part of the results, a similarity search by DTW is attempted. This yield poor results, with a classification success rate of around 10% on average across letters.

Acknowledgements

This thesis would not be possible without the help and encouraging meetings from my supervisor Professor Puneet Sharma, at the department of Engineering and Safety, University of Tromsø. Thank you! I would also like to thank Thalmic Labs, which makes the Myo armband, and enables developers to explore all it's possibilities, through providing an open source environment.

Så vil eg takke alle vennar i Tromsø, samt foreldra og syskena mine, som har støtta opp og hatt trua på meg. Ein særskild takk til Sara, Torgeir og Ingrid som har vore mine akademiske førebileter ved UiT. Personen eg kan takke høgast når det kjem til studiet, er Hallvard Haugen, medstudent, god ven og livbøye. Takk for alle dei fine stundene.

Den største takken er retta mot min sambuar og kjærast Anita Finne. Du er det beste som har skjedd meg, og det som gjer at eg står opp om morgonen.

Contents

List of Figures

List of Tables

My list of definitions

GCA	Gesture Control Armband
sEMG	surface ElectroMyoGraph
ACF	Autocorrelation Function
CCF	Crosscorrelation Function
SVD	Singular Value decomposition
IMU	Inertial Measurement Unit
CSV	Comma Separated Values
PCA	Principal Component Analysis
KNN	K-Nearest Neighbors
Jolt	An abrupt, rough or violent movement
fps	Frames per second
SDK	Software Development Kit
d	d

1 Introduction

This chapter is a short description of the motivation behind this project, along with the problem description and the structure of the given thesis.

1.1 Motivation

The motivation for this thesis is to explore the Myo armband's potential as a multi sensor for handwriting recognition. Wearable technologies, like Google Glass and smart watches aspires to make interaction between humans and machines easier and more intuitive. The nature of these technologies makes them a sought after tool for machine interaction in the Virtual Reality (VR) and Augmented Reality (AR) market. In VR/AR the users vision is partially or completely replaced with a computer generated environment, which aims to aid or entertain the user through enabling tools or features to interact with. In this configuration, a traditional way of human-computer interaction, like the mouse and keyboard may in some cases feel awkward and/or may restrict the level of engagement for the user. Communication between humans is to a large extent based gestures and body language. A similar form of communication between humans and machines could be favorable in the future, and one of the means of achieving this goal could be to let computer process data from wearable technologies as a way of body language interpretation.

Throughout numerous professions handwriting is still the most efficient form of writing. Teachers are still using blackboards around the world, and many students are still taking notes in lectures using pen and paper. Although taking notes on laptops and tablets have become more common in universities and colleges, some institutions of higher education along with most primary schools and high schools prohibits the use of these tools during class, as studies has found that some students find them distractive [Fried, Carrie B., 2008]. The clear advantage of documents written and stored on computers is the ease of organization and accessibility. If all hand written documents of individual students could be translated into digital writing and accessed by teachers and parents, and grammars spelling and hand writing technique could be assessed by machine learning tools, to give a faster more detailed feedback.

1.2 Problem Description

The objective of this thesis is to investigate the Myo armbands potential as a tool for handwriting recognition. The Myo's nine-axis inertial measurement unit and eight surface electromyography (sEMG) sensors provides data, which might be sufficient in information to correctly label individual written letters into respective classes. Before the data can be analyzed, an accurate method for extracting data corresponding to the written letters, has to be developed. This is necessary as there are no large public datasets containing captured IMU and sEMG sequences of individual letter recorded by Myo. Another challenge in this thesis is to collect enough data from a participating individual, as letters might not be classifiable across subjects, due to the variation in timing, and writing techniques.

2

Theoretical Background

2.1 Inertial Measurement Unit (IMU)

An IMU measures spatial information and consists usually of a gyroscope, an accelerometer, and often a magnetometer. The perfect and ideal IMU will provide continuous information on orientation and acceleration, thus, it could provide perfect spatial coordinates (x, y, z) at any time t by a double integration of the accelerometer data. In reality a IMU can be used to provide coordinates for short time periods, but have to be updated frequently by GPS, for the values not to accumulate error and become incorrect. In this thesis we will not focus on the navigation aspects of the IMU, but rather search for patterns in the sensor data, which may enable us to classify the letters written by subjects. The accelerometer provides data along the x,y, and z-axes, where the unit is given in $[m/s^2]$. Combined with a gyroscope which measures orientation in a 3-dimensional coordinate system, a orientation vector on quaternion form can be calculated. The quaternion has an added fourth dimension which represents the IMUs rotated angle about the axis given by the three first orientation coordinates.

2.2 Myo Armband

2.2.1 Introduction

Myo is a wearable multi sensor armband produced by Thalmic Labs for hand movement and gesture recognition. The armband measures spatial movement and muscle activity in the upper forearm and transmits a live feed via Bluetooth to a connected device such as a smart phone or a computer fitted with the USB Bluetooth adapter. Data is registered by means of eight surface electromyography sensors and a nine-axis inertia measurement unit (IMU) consisting of an accelerometer, a gyroscope and a magnetometer, governing three axes each.

Table 2.1: Myo GCA Hardware

Sensors 1	Medical Grade Stainless Steel EMG sensors, Highly sensitive 9-axis IMU containing 3-axis gyroscope, 3-axis accelerometer, 3-axis magnetometer.
LEDs	Dual indicator LEDs.
Processor	ARM Cortex M4 Processor.
Haptic Feedback	Short, medium, long vibrations.

The Myo comes installed with a set of recognizable hand gestures for the Myo Connect app, which is the software for Myo on Microsoft and Mac OSX platforms. These gestures and movements (see figure) can be used to move between slides in Power Point, navigate and scroll in web browsers.The Myo is marketed as a presentation tool for lecturers, as a controller for radio controlled drones, cars etc., and as a controller for computer applications. All the data collected in this thesis is from the Myo gesture control armband.

Figure 2.1: Figure illustrates the default recognizable gestures and movements of the Myo armband.
Source: [Thalmic Labs, 2018a].

2.2.2 Technical aspects

The Myo kit comes equipped with the following items:

- Myo armband
- 10 Myo sizing clips
- Micro USB cable
- Bluetooth adapter

The Myo Armband consists of eight rigid rectangular pods connected with a flexible material, as shown in figure 2.2.

Figure 2.2: Myo Armband with pods numerated on the right image.
Source: [Bernhardt, Paul, 2015a] and [Thalmic Labs, 2018a].

The material used in the Myo armband is a flexible type of elastomer, similar to materials used in other wearables. Circumference range of the armband is 19-34 cm. This range is due only to the armbands elasticity. Total weight of the armband is 93 grams, and it is fitted with two LED lights which pulsate in different frequencies and colors based on current status. A micro USB-port is used to charge the Myo's two 2.5 volt batteries located in pod 3 and 5. One charge-up is supposedly sufficient for one full days use [Thalmic Labs, 2018b].

2.2.3 Myo Sensor Information

Each pod is equipped with one surface electromyography sensor, which measures difference in electric potential between muscles directly under the skin. Pod number 4 is the main pod, and hold the micro USB port and the nine-axis IMU. This is also where the processor is located, which calculates the

orientation data based on the accelerometer and gyroscope. Thalmic Labs has, although advertising the magnetometer, chosen to disable the access to this data from the Myo SDK. The current (Dec. 2017) available raw data output from the Myo-Data-Capture is sEMG, accelerometer, gyroscope and orientation, where the latter is given in both quaternions and Euler angles.

Table 2.2: Additional to the dimensions of sensor data given in this table, each file contains a vector with timestamps corresponding to individual data samples. *Source: [Bernhardt, Paul, 2015b]*

Name of Output Files	Data Dimensionality	Sampling Frequency
Accelerometer	3	50 Hz
(EMG	8	200 Hz
Gyro	3	50 Hz
Orientation	4	50 Hz
OrientationEuler	3	50 Hz

The Euler angle representation of the orientation data is omitted in the rest of this thesis as the quaternion representation was easier to visualize.

2.2.4 Data Capture

Data capture from the Myo Armband is accessible though Thalmic Labs app called Myo-Data-Capture[1]. This is a simple executable command line which logs data from the Myo sensors and stores them as comma-separated values(csv) in five individual csv-files. Each of the five files have a timestamp in the fist column, and recorded data in the following columns. Since the chosen computational engine for this thesis is Matlab, a Matlab function was created to send commands to "Terminal" which is the command-line interface for UNIX-based operating systems. The recordMyo Matlab function takes time in seconds as input, and this determines the number of seconds recorded by Myo. When recordMyo has terminated, the resulting csv-files are stored in the current Matlab folder.

2.3 Data Processing Methods.

When working with high dimensional data from multiple sensors, chances are that some of the data is redundant. When recording data from hand movements

1. Available at: https://market.myo.com/app/55009793e4b02e27fd3abe79/myo-data-capture

while writing, the forearm is in contact with the table surface trough writing. This implies that accelerometer data describing non-parallel movement in respect to the table surface, could be less valuable for later classification. If this would be the case, than dimensionality reduction through Principal Component Analysis could be used to identify and remove redundant data, without loss of information.

2.4 Principal Component Analysis(PCA)

PCA applies a linear transformation to a data samples $\mathbf{X}$ of dimensionality $[n \times m]$ where n is the number of samples (e.g. over a given time), and m is the number of observations. The linear transformation matrix $\mathbf{A}$ is found through eigen-decomposition of the sample covariance matrix $\mathbf{R}$, where the corresponding eigenvectors are columns in the transformation matrix $\mathbf{A}$.

$$\mathbf{R} = \sum_{i=1}^{N-1} \mathbf{x}_i \mathbf{x}_i^{\mathbf{T}} \tag{2.1}$$

$$\mathbf{A} = \mathbf{a_0}, \mathbf{a_1}, \mathbf{a_2}, \cdots \mathbf{a_{m-1}} \tag{2.2}$$

The eigenvectors of $\mathbf{A}$ are sorted in descending order, along with the eigen values $\Lambda = [\lambda_0, \lambda_1, \lambda_1, \cdots, \lambda_{m-1}]$. The first eigenvector will then represent the basis of highest variance in the sample data $\mathbf{X}$. $\mathbf{X}$ can now be multiplied with the transformation matrix $\mathbf{A}^{\mathbf{T}}$ to yield out transformed data samples, where largest variance across dimensions are bound in the first column vector $\mathbf{y}_0$.

$$\mathbf{Y} = \mathbf{X}\mathbf{A}^{\mathbf{T}} \tag{2.3}$$

For further use of the transformed data, dimensions with low or zero variance can be neglected. When performing PCA on raw data, which are to be used for classification, it is important to note that dimensionality reduction, where dimensions of low variance are neglected can greatly effect classification performance. This can occur in datasets where the variance within observations are high, but variance between classes are are very low. An example of this is given in figure 2.3

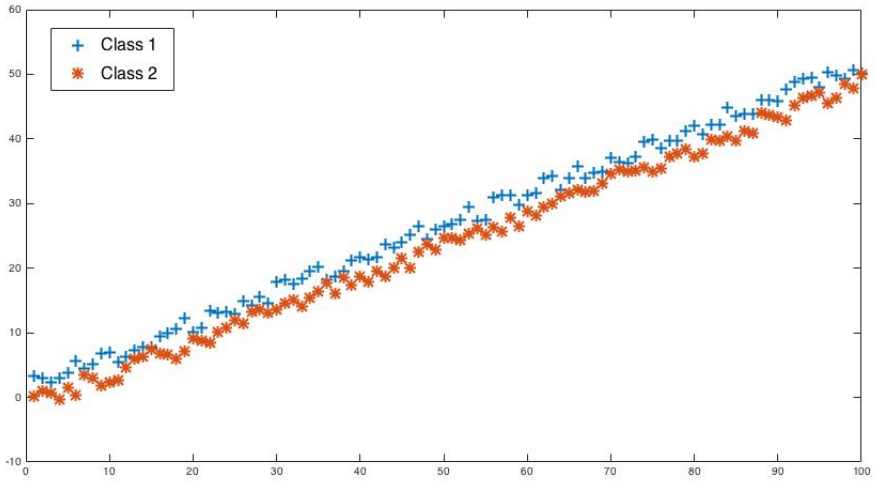


Figure 2.3: Objects of class 1 plotted in blue, and class 2 plotted in orange. Classes are indistinguishable along the vector of greatest variance.

In figure 2.3, dimensionality reduction of raw data by PCA would be catastrophic for further classification, as all the data would be projected on to the line $y = 0.5x$. This would remove the information which distinguish the two classes from each other.

2.5 Correlation Coefficient

The correlation coefficient ρ is a measure of the linear dependency between two random variables A and B of equal length N, and is given by

$$\rho(A,B) = \frac{1}{N-1}\sum_{i=1}^{N}\left(\frac{A_i - \mu_A}{\sigma_A}\right)\left(\frac{B_i - \mu_B}{\sigma_B}\right) \tag{2.4}$$

where μ_A and σ_A are the mean and standard deviation of A, respectively. The correlation coefficient have values ranging from -1 to 1, where $\rho = -1$ implies a perfect negative correlation, a value of $\rho = 0$ implies no correlation, while a value of $\rho = 1$ states maximum positive correlation.

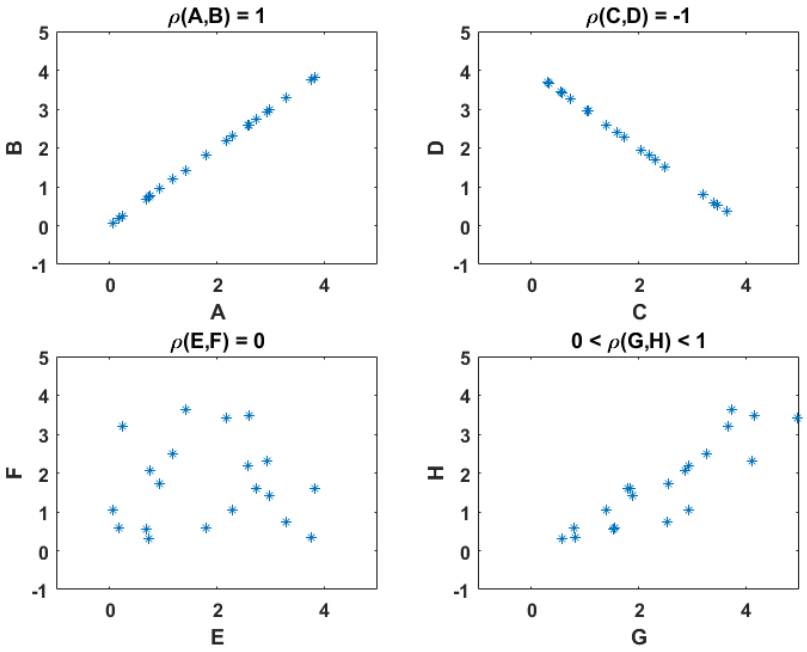


Figure 2.4: Examples of correlation coefficients $\rho(A, B) = 1, \rho(C, D) = -1, \rho(E, F) = 0$ and $0 < \rho(G, H) < 1$.
Source: peronal collection (produced in Matlab)

[Box et al., 2007] In figure 2.4 we are presented with four plots where in each plot two random variables are plotted against each other. In the upper left plot the two variables A and B have a maximal correlation, hence $\rho = 1$. In the upper right plot, variables C and D have a maximum negative correlation, hence $\rho = -1$. For the lower left plot, there seem to be no correlation between E and F, and their correlation coefficient is zero, while in the lower right plot, a positive correlation is evident, and the correlation coefficient will therefor be between zero and 1.

2.6 Cross Correlation

The cross correlation function for two real discrete time signals X and Y is given by

$$\rho_{xy}(\tau) \equiv \sum_{n=-\infty}^{\infty} \frac{(x_n - \mu_x)(y_{n-\tau} - \mu_y)}{\sigma_x \sigma_y} \tag{2.5}$$

where the time shift τ is also referred to as the lag. [Box et al., 2007] The cross correlation is a measure of similarity between two signals where output can also indicate if one signal is lagging the other. The signal data in this thesis vary in space and time and hence cross correlation is a first basic technique that can be employed for finding the level of similarity between two given space time signals.

2.7 Auto Correlation

Auto correlation calculates the cross correlation of one signal $\mathbf{X}$ to itself. If $\mathbf{X}$ is a discrete signal consisting of L samples, the result from an auto correlation is a new function $\rho_{xx}(\tau)$, where $\tau = [-L+1, L+2, \ldots, -1, 0, 1, \ldots, L-2, L-1]$, given by

$$\rho_{xx}(\tau) \equiv \sum_{n=0}^{L-1} \frac{(x_n - \mu_x)(x_{n-\tau} - \mu_x)}{\sigma_x^2} \tag{2.6}$$

As seen from function 2.7 the correlation coefficient is calculated between the signal and a version of the signal which is shifted by tau. For $\tau = 0$, $\rho_{xx}(0) = \sigma_x \sigma_x / \sigma_x^2 = 1$. While the autocorrelation function is often used to search for periodicity within a signal, we will in this thesis merely use it as a performance reference for cross correlation.

2.8 Singular Value Decomposition

The singular value decomposition of the $n \times m$ matrix $\mathbf{A}$ is given through the eigenvalue decomposition of the symmetric matrix AA^T, such that,

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\mathbf{T} = \textbf{(orthonormal)(diagonal)(orthonormal)} \tag{2.7}$$

$$\mathbf{A}\mathbf{A}^\mathbf{T} = (\mathbf{U}\Sigma\mathbf{V}^\mathbf{T})(\mathbf{V}\Sigma^\mathbf{T}\mathbf{U}^\mathbf{T}) = \mathbf{U}\Sigma\Sigma^\mathbf{T}\mathbf{U} = \mathbf{U}\Lambda\mathbf{U}^\mathbf{T} \Leftrightarrow \mathbf{A}\mathbf{U} = \Lambda\mathbf{U} \tag{2.8}$$

, and

$$\mathbf{A}^\mathbf{T}\mathbf{A} = (\mathbf{V}\Sigma^\mathbf{T}\mathbf{U}^\mathbf{T})(\mathbf{U}\Sigma\mathbf{V}^\mathbf{T}) = \mathbf{V}\Sigma^\mathbf{T}\Sigma\mathbf{V} = \mathbf{V}\Lambda\mathbf{V}^\mathbf{T} \Leftrightarrow \mathbf{A}\mathbf{V} = \Lambda\mathbf{V} \tag{2.9}$$

, where the $m \times m$ matrix $\mathbf{U}$ contains the eigenvectors of $\mathbf{A}\mathbf{A}^\mathbf{T}$, the $n \times n$ matrix $\mathbf{V}$ are the eigenvectors of $\mathbf{A}^\mathbf{T}\mathbf{A}$. Σ $(n \times m)$ holds the the $r = rank\mathbf{(A)}$ singular values of A, which consists of the square root of the positive eigenvalues of Λ, sorted in a decreasing order. [Str, 2006]. We can use the SVD to determine the degree of linear dependency of n row vectors of length m by looking at the singular values of the $(n \times m)$ matrix $(m > n)$they make up. If all row vectors of A are *linearly dependent*, then Σ will be a diagonal matrix where all entries

are zero with the exception of σ_{11}, which holds the square root of the one and only non-zero eigenvalue $\lambda_1 1$, originating from $\mathbf{AA^T}$. For an $(n \times m)$ $(n < m)$, where all entries are random variables with zero mean, the n singular values σ_{ii} of $\mathbf{A}$ will have similar values as m grows large.

2.9 Dynamic Time Warping(DTW)

DTW is an algorithm often used when performing similarity search between two temporal sequences of different speed. This is useful in many fields of time series analysis, and was originally developed for speech recognition [H. Sakoe and S. Chiba, 1978]. DTW is a branch of dynamic programming where temporal time series can compared to each other despite deformation and warping in time. This is achieved by first creating a cost matrix also known as a distance matrix between the two temporal time series which are to be compared. Before distance matrix can be computed the two time series, $X = [x_1, x_2, \cdots, x_N]$ and $Y = [y_1, y_2, \cdots, y_M]$, where $N \in \mathbb{N}$ and $M \in \mathbb{N}$, must be Z-normalized.

$$\begin{aligned} X_z &= \frac{X - \mu}{\sigma(X)} \\ Y_z &= \frac{Y - \mu}{\sigma(Y)} \end{aligned} \tag{2.10}$$

, where

$$\mu_x = \frac{1}{N} \sum_{i=1}^{N} X_i \tag{2.11}$$

, and

$$\sigma(X) = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} |X_i - \mu_x|^2} \tag{2.12}$$

Next step is to calculate the $[N \times M]$ cumulative distance matrix $\mathbf{D}$, which is

given by

$$\begin{aligned} D(i,j) = |x_i - y_j| \quad & +min[D(i+1,j) \\ & \quad D(i,j+1) \\ & \quad D(i+1,j+1)] \quad \text{for}(i \geq 1 \leq j) \\ D(1,j) = |x_1 - y_j| \quad & +D(1,j-1) \quad \text{for}(i = 1, j > 1) \\ D(i,1) = |x_i - y_1| \quad & +D(i-1,1) \quad \text{for}(j = 1, i > 1) \\ D(1,1) = |x_1 - y_1| & \end{aligned} \tag{2.13}$$

$for i = 1, .., N$ and $j = 1, ..., N$. **D** hold the cumulative distance between every two data points in **X** and **Y**. The next step of the DTW algorithm is to find the warping path of lowest cost. The warping path $p = [p_1, \cdots, p_L]$, where $p_l = (n_l, m_l) \in [1:N] \times [1:M]$ for $l \in [1:L]$ is a sequence which satisfies the following conditions [Müller, Meinard, 2007]:

- Boundary conditions are: $p_1 = D(1,1)$ and $p_L = (D(N,M))$
- Monotonically increasing: $n_1 \leq n_2 \leq \cdots \leq n_L and m_1 \leq m_2 \leq \cdots \leq m_L$
- Step size condition: $p_{l+1} - p_l \in \{(1,0),(0,1),(1,1)\} for l \in [1:L-1]$

To find the optimal warping path, the matrix D is traversed from $D(N,M)$ to $D(1,1)$, where p_{l-1} is chosen as the minimum value in **D**, which satisfies the three conditions listed above. For $p_L = D(N,M)$ and $p_l = D(i,j)$, we find p_{l-1}, by:

$$\begin{aligned} p_{l-1} = min[D(i-1,j) \\ D(i,j-1) \\ D(i-1,j-1)] \end{aligned} \tag{2.14}$$

The classical DTW algorithm has no restrictions on the warping path other than the three mentioned above. However, for a less time consuming algorithm, we introduce a warping constraint, which restricts the path of p. An example of such a constraint is the Sakoe-Chiba Band, which defines the maximum tolerated relative deviation w of the warping path from the diagonal of the matrix **D**.

$$w \leq \frac{r}{N} \tag{2.15}$$

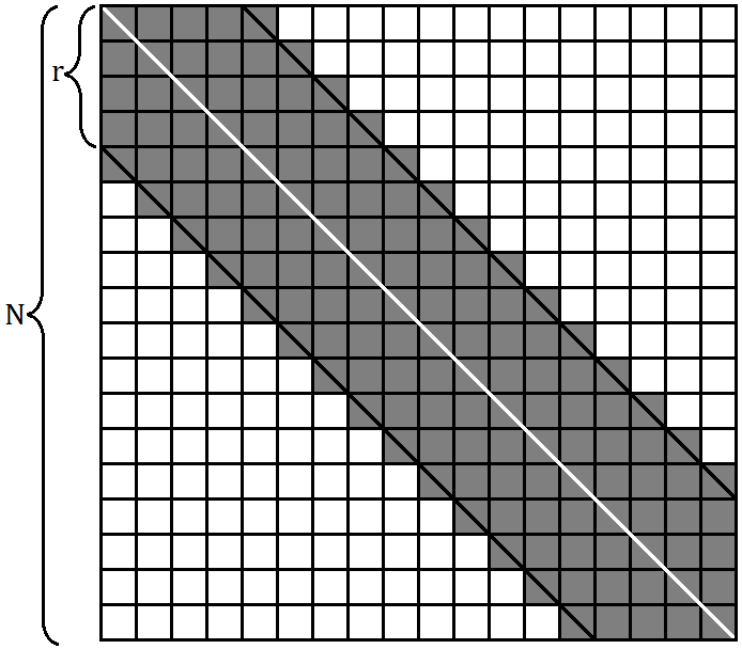


Figure 2.5: Illustration of a Sakoe-Chiba band, where r is the vertical distance from the diagonal and n is length of both signals.

The Sakoe-Chiba band is a very simple way of restricting the warping path, and can be modified to non-square distance matrices as well, by $w \leq floor(|n-m|)$, where $D_{ij} = D(n,m)$. In figure 2.5 the gray diagonal area marks the values calculated in the cumulative $[20 \times 20]$ cost matrix, with a $w = 5/20 = 0.25$ or 25%.

2.10 Pattern Recognition

Pattern recognition is a field of science, where the goal is to recognize or classify specific patterns or features in data. This process can be performed by techniques though supervised classification, where the classifier is trained on preclassified data, or by unsupervised classification, which just considers the data at hand without any prehand knowledge. The two approaches are further discussed in the two following sections.

2.10.1 Unsupervised and Supervised learning

Unsupervised learning is an sub category of machine where the goal is to unravel similarities of a given set of feature vectors $\mathbf{X}$, with the goal of grouping together vectors that are similar to each other. In unsupervised learning also known as unsupervised pattern recognition or clustering, there is no a priori knowledge of classes or labels of the feature vectors. Still, unsupervised learning tasks appear in many fields of social sciences and engineering as a way of clustering, or anomaly detection in large datasets. Supervised pattern recognition uses a priori knowledge to classify new feature vectors into already established classes. One common method for supervised classification is to use a training set where feature vectors are already assigned a class, such that the classifier can learn which features the objects of common class members share and what best separates them from the members of other classes. In this type of machine learning the classifier "practices" on the training set, and then applied its knowledge on new unclassified feature vectors. Another version of supervised learning, uses the training data in the classification task. An example of the latter method is the k-Nearest Neighbor classifier

A supervised classifier assigns a new unclassified feature vector to a class based on a priori knowledge gathered from a training set where the true class membership is known. The training set can either be used to train the classifier, or directly in the classification process, as is the case in the Nearest Neighbor (k-NN) classifier. Here, a distance measure is used to determine a unclassified feature vector's distance to every feature vector in the training set, no matter class. The new vector vector is then assigned to the class which hold the majority of the new features k nearest neighbors, where k is a positive integer. The k-NN classifier is further elaborated in section 2.11.

2.11 Supervised K-Nearest Neighbors(k-NN) Classification

The supervised version of the k-NN classification algorithm is a suboptimal but popular nonlinear classifier Consider the unknown feature vector **x**, which are to be assign one class ω_1. Given the N training vectors, we identify the k nearest neighbors regardless of their class label. For the k samples, we identify the number of vectors, k_i , that belong to the class ω_i, i =1,2,3,...,M. It follows from this, that $\sum ik_i = k$. The unknown feature vector are now assigned to the class ω_i with the maximum numbers of k_i samples [Theodoridis and Koutroumbas, 2008]. A rule of thumb is to choose k as an odd number for two-class problems, and in general not to be a multiple of M. This assures that we avoid ending up with a draw when counting the classes of the k nearest neighbors. For $k = 1$ we have a very simple classifier, as the unknown feature vector x is simply assigned the class of its single nearest neighbor. Although primitive, the case of $k = 1$ is admissible in some classification problems, meaning that it yields the lowest risk of miss-classification for all $1 \leq k \leq \infty$ [Cover, T., and Hart, P., 1967]. The k-NN classifier can be modified by the choice of distance measure techniques, where two popular once are Euclidean and Cityblock distance.

3 Methodology

This chapter will discuss the different approaches and techniques used to collect and process the data, as well as the method of classification

3.1 Diversity in Handwriting

Handwriting is a very complex activity to generalize, as each subject has his or her distinct way of drawing each letter. All though two written letters appear identical and in fact are identical in shape and size, the process of creating the two respective letters need not be the same. Take the letter capital E as an example. Capital *E* is made up of 4 straight lines; three horizontal and one vertical. When drawing this letter, is it not given which line is drawn first, and it is not given how many times the pen or pencil is lifted from the paper from start to finish. In fact, it is easy to quickly come up with at least 10 different writing methods for writing the letter capital *E*. Another factor, which adds to the complexity of the problem, is the variation in writing techniques or hand posture. Some people move their entire arm, from shoulder down, while writing, while others limit their arm movement, and use mainly their wrist and finger joints to move the pen. This means that classification of all letters spanned by all writing techniques will require a vast set of training data. We will therefor in this project mainly focus on letter recognition where the same person provides both the query data and training data.

3.2 Comments on the Choice of Target Letters.

In the process of choosing the target capital letters *E*, *L*, *O* and *R*, the diversity in letter shape, and range of writing methods were considered. The four letters are chosen based on the geometric shapes that each letter consists of, and the arguments for choosing the letters *E*, *L*, *O* and *R* are as follows:

- **E** - The letter consists of 4 straight lines, where each line is parallel to one out of two perpendicular axis. The letter can be written in a vast number of different ways when accounting the permutations of line orders, and draw direction. The letter is also the only chosen letter where the writer is forced to lift the pen at least once during the drawing, hence the letter *E* is expected to be the most difficult letter to classify across different subjects. One drawing configuration of *E*, where the two first lines forms the letter *L*, later followed by the two last horizontal lines, could subject letter *E* and *L* to be miss-classified.

- **L** - Letter consists of two perpendicular lines, which is in almost every case written the same way across subjects. That is, starting from top to bottom, then going left to right, all in one motion, without lifting the pen. If the letter *L* turn out to be impossible to classify across subjects then this would indicate that the cross-subject letter classification is not feasible for other other letters of the alphabet.

- **O** - Letter *O* was chosen due to its circular shape, which is interesting as the motion which creates a circle has no abrupt changes in acceleration, as oppose to the right angles in *L* and possibly in *E*. The potential case, where the pen displacement is recorded as an identical but scaled down displacement of the Myo's IMU, would implicate that the accelerometer data from the Myo armband corresponding to the letter *O* would plot as a one-period sinusoid when plotted as a function of time, in both axis that are parallel to the table.

- **R** - Was chosen as it consists of one vertical line, one arc and one diagonal line. The letter can be drawn in one line, without lifting the pen, but also in two lines, and a third option introduces a drawing path where one line is traversed twice, namely the vertical.

The reason for collecting only capital letters was that we expect them to be more standardized and less influenced by personalization from the subjects.

3.3 Video Recording of experiment

For the recording of each individual experiment, the built in camera of a laptop was used. This camera has a 640x480 pixel resolution and a frame rate of 30 fps. A Microsoft Studio LifeCam HD 1080p with 30 fps was provided by University of Tromsø, and originally the primary choice for the video recording. This camera was rejected after several recordings gave a frame rate of less than 15fps. When conducting the experiments, recording of the video is started a few seconds before the capture of Myo Armband data, as they are controlled through two independent softwares. After recording, the video is stored in a folder along with corresponding data output from Myo. The code for Myo-Data-Capture is compiled in C++ but is activated in Matlab through the use of a built-in function "system()", which sends command lines to the terminal of the computer. The webcam is operated from within Matlab, using the built-in functions "videoWriter()".

3.4 Data Collection

In the experiment setup, each subject writes ten consecutive repetitions of the capital letters E, L, O, and R on a piece of A4 paper, twice. This results in a total of 80 letters collected form each subject. Letters were written from left to right, from short side to short side on an A4 paper. first row of letters consists of ten repetitions for E, second row ten times L, and so on. The letter size is in the range 1-3cm, depending on the subject.

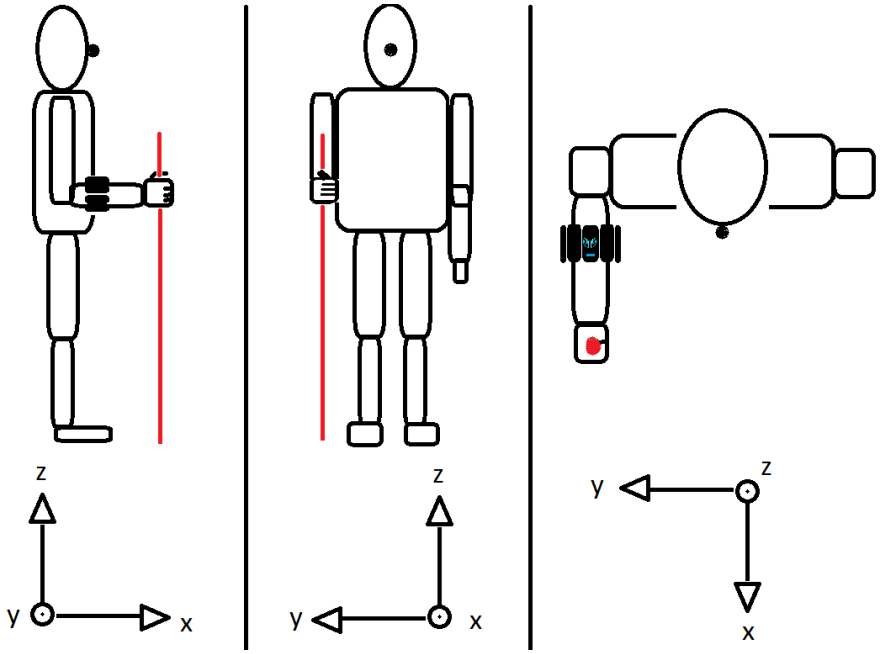


Figure 3.1: Myo's position on the arm, seen from three different angles.

The Myo armband was worn in the writing arm of the subject, such that when the under arm is pointing forwards, elbow is angled at 90 degrees and touching the side of your body, and the fist is clinched as around a pole, the logo on the Myo armband is pointing straight upwards and the Myo is 1 cm. from the subjects biceps. The recording camera is placed on the side opposite to the writing arm, at a 48 degree angle relative to the table surface. The data analyzed in this thesis all originates from subjects that are right handed.

3.5 Myo Data Capture

3.5.1 Data Preprocessing

The built-in Matlab function "csvread" is used to extract and transform the time stamp vector and the recorded data vectors from each of the 5 csv-files into 1-D and 2-D arrays, respectively. The time stamp vectors from IMU sensors are all identical, thus only the time stamp vector from the accelerometer data and EMG data are saved.

The recorded video of each experiment is reviewed in Matlab using the built-in function "implay()". The reason for this is that we want to identify sequences

which corresponds exclusively to data recorded during the writing of a given letter. We can extract these sequences my means of video review, but for this, we need a way of knowing which video frames that corresponds to which recorded sensor data in respect to time. We need an equivalent to a "Hollywood movie clapper". The movie clapper is used to synchronize video and audio in the movie business, but in our situation we need to synchronize sensor data and video. We solve this by giving the main pod of the Myo armband a firm slap while video and sensor data are being recorded. This slap which we will from now refer to as a *jolt*, is visible in both the video and the accelerometer data. And since we have time stamps corresponding to each video frame and sensor data point, we can align everything in respect to time.

When reviewing the recorded video, we have to manually write down the frame number of the jolt, see figurefigure 3.2, along with the frames that corresponds to the beginning and the end of each letter (figure 3.3). The frame number is visible in the lower right corner of the Matlab “Movie Player” as shown in figure 3.2.

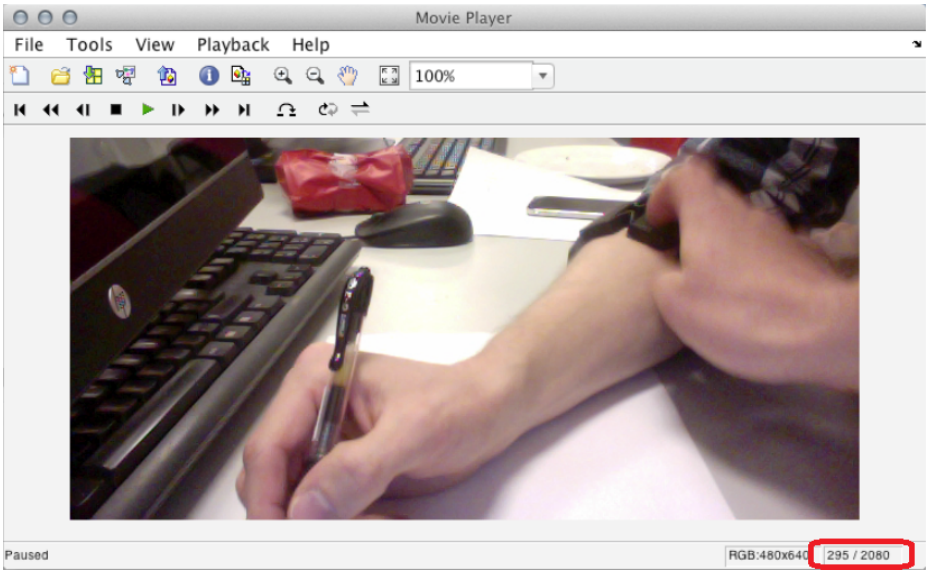


Figure 3.2: Video frame of the jolt, where a firm blow is delivered to pod 4 of the Myo Armband. Frame number is visible and outlined in red, along with total number of frames for the video file.

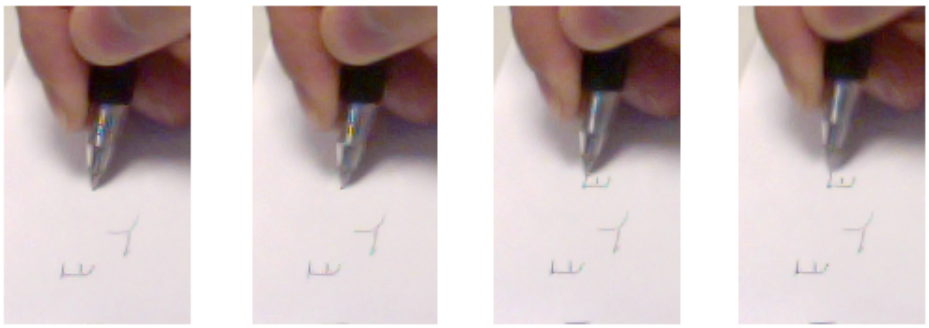


Figure 3.3: Picture on far left shows no pen mark, while a small pen mark is visible in the mid-left image as the pen moves towards right in the frame. The far right picture is the first frame where the pen is disconnected from the paper, while in the mid right, the pen is still connected to the paper.

The frame number from the jolt is stored as a value in an info file, along with frame numbers which corresponds to the last frame before a pen mark is visible, and the first frame of pen disconnected from paper. Examples of these frames are the far left and far right pictures in figure 3.3 respectively. We will from now refer to these frames as start-frame and end-frame. Each of the start- and end-frames are stored consecutively in four arrays, one for each type of letter. Hence for ten repetitions of the letter E, the array that holds it's start and end-frames has 20 elements.

3.5.2 Frame Rate of Recorded Video

The frame rate of the recorded video can be found in the Matlab Movie Player window, in Tools->Video Information. This frame rate can not be used as it is only an estimate of the real frame rate. The real frame rate is not constant during a video recording, but fluctuates. This means that the frame rate in the beginning of a recording is not necessarily the same as in the end of that same recording. For each start-frame and end-frame to accurately predict the corresponding time stamps in the Myo data, a new time vector $\mathbf{FR_r}$has to be constructed from the time stamps of each video frame. The vector $\mathbf{FR_r}$ is a time vector, with unevenly spaces elements, where

$$\begin{aligned} FR_r(1) &= 0 \\ FR_r(2) &= \textbf{Seconds elapsed between the capture of frame 1 and 2.} \\ FR_r(2) &= \textbf{Seconds elapsed between the capture of frame 1 and 3.} \\ &\vdots \\ FR_r(N) &= \textbf{Seconds elapsed between the capture of frame 1 and N.} \end{aligned} \tag{3.1}$$

, where N is the last frame of the video file. The time stamp of a video frame is given in seconds elapsed since capture of the first frame in the video file. By not considering this variation in frame rate the data extracted from the Myo recordings will be shifted, thus including data which is not representing writing, and leaving out data that is.

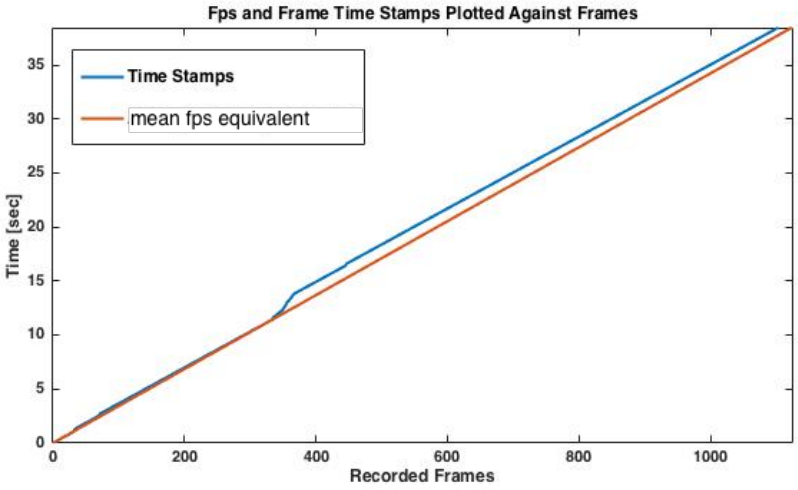


Figure 3.4: Elapsed time in seconds plotted against elapsed frames, where the orange line corresponds to a constant frame rate while the blue is the true frame rate.

As seen in figure 3.4 the true fps deviates from the mean fps value given in Matlab's *Video Information*, where the maximum deviation occurs at frame 447, where the difference is at 1,286 seconds. This deviation can also be seen in figure 3.5, where the two different fps approaches are plotted against elapsed frames.

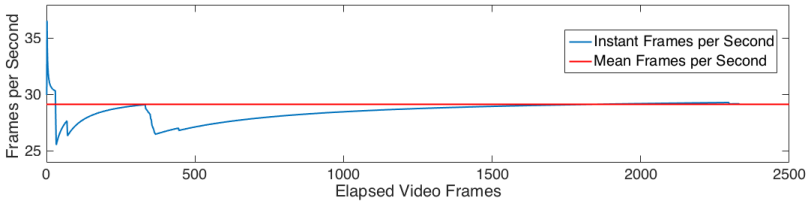


Figure 3.5: Frames per second plotted against frames, for true fps in blue, and mean fps in red.

Since the mean time spent on writing a single letter in the data set corresponding to figure 3.4 was 0.647 seconds, using the mean fps from the video, would lead to the wrong data being extracted from data recorded by Myo.

A different method for sectioning out individual letters from the recorded myo data was attempted, and is further discussed in Chapter 4.

3.5.3 Extracting Myo Data for Classification Training

To extract a correct Myo data sequence, which are to represent the sensor data captured during the drawing of an individual letter, recorded video has to be aligned with the Myo sensor data in respect to time. The jolt in the initializing phase of each recording is our tool for alignment. The *jolt data point* is distinct and easy to locate as it causes a spike in the accelerometer data.

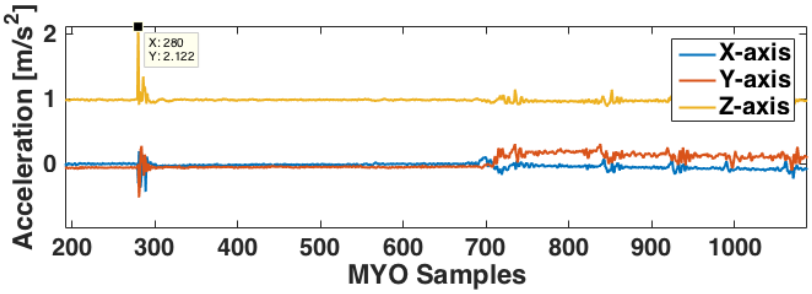


Figure 3.6: Plot of data from accelerometer, where the jolt data point is easily distinguishable from the the rest of the data.

To get sensor data points corresponding to all star and end-frames, time stamp of the *jolt data point* is subtracted form the *sensor* time stamp vector to form vector **X**, and the video time stamp corresponding to the *jolt frame* is subtracted from $\mathbf{FR_r}$ in equation 3.1 to form vector **Y**. **X** hold the same amount of elements as the original sensor data time stamp vector, where the time stamp corresponding to the jolt data point is now zero. **Y** holds only 20 elements, one for each start and stop-frame. The elements in **Y** holds values representing time since jolt, just as in **X**. If for simplicity we assume that **X** holds 1000 elements and J_{indx} is the index of the jolt data point, then the twenty start and end-*data points* $\mathrm{Z_i}$, i = 1,2,3,...,20, are extracted from **X** by,

$$Z_i = J_{indx} + j \quad \text{for} \quad min[Y_i - X_j] \tag{3.2}$$

Z is now a vector which holds the twenty indexes corresponding to the start and end of each written letter of a given type.

3.5.4 Validation of Letter Extraction Method

Although the method in section 3.5.3 is extremely time consuming and awkward, the results are fairly accurate. A test was performed, where 16 synchronization jolts were delivered to the Myo while recording both video and Myo data. The first jolt was used as a synchronization jolt, while the other jolts were predicted by the method described in section 3.5.3, where start and end data points in Myo data are predicted by the synchronization jolt along with the start- and end-frames from the corresponding video recording. The predicted jolts were compared to the highest positive value of each spike visible in Myo accelerometer data for Z-direction.

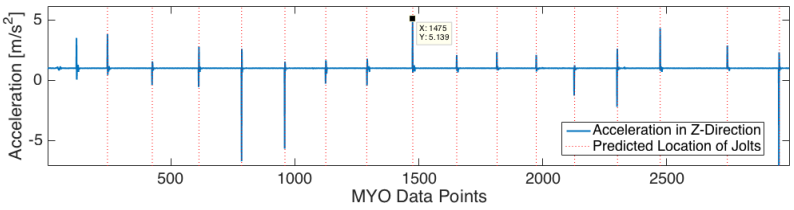


Figure 3.7: Plot of test for synchronization jolt accuracy. Each jolt prediction is plotted as a red dotted line on top of the z-accelerometer data.

The result of this test showed that the offset E_i for the fifteen predictions P_i, compared to the 15 positive spikes S_i in Myo z-acceleration (see figurefigure 3.7) was:

$$\overline{E} = \frac{1}{15}\sum_{i=1}^{15} E_i = \frac{1}{15}\sum_{i=1}^{15}\left(P_i - S_i\right) = 0 \tag{3.3}$$

, and:

$$\sigma_E = \sqrt{\frac{1}{15}\sum_{i=1}^{15}(P_i - S_i)^2} = 0.89 \tag{3.4}$$

As these predictions are of indexes of data points, where sample frequency is 50Hz, the standard deviation $std(E) = 0.89$ corresponds to a $\sigma_{E_{Time}}$ of

$$\sigma_{E_{Time}} = 0.89/50Hz = 18ms \tag{3.5}$$

All the Myo data from one recording is sectioned into 40 sequences, by extracting Myo sensor data between indexes Z(k) to Z(k+1), where k is odd. When the raw Myo accelerometer data in x-direction is plotted with vertical lines at

each start and end data point, see figure 3.8 we can clearly see that the vertical lines comes prior and following a a period of larger amplitudes. This is caused by the moving of the arm, in between each letter.

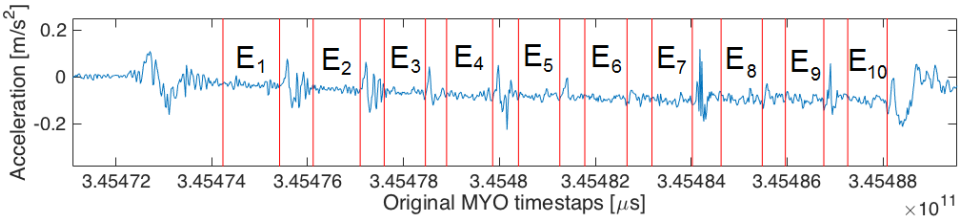


Figure 3.8: Plot of data from accelerometer in x-direction, where the vertical lines represent beginning and end of the new sequences, here emphasized by E_1, E_2,...,E_{10}

Sequences as shown in figure 3.8 are made for every sensor and every recorded letter (E,L,O and R), and stored according to which classification method to use.

3.5.5 Comment on Recorded sEMG Time Stamps.

While the sampling frequency of the Myo armband is 200Hz for EMG data, the Bluetooth is not able so send this information at 200Hz. Data originating from two consecutive time stamps are therefor sent in one package via the bluetooth, hence they are both provided with identical timestamps. The time stamp vectors is therefor re-sampled such that corresponding data is evenly spaced when plotted against the re-sampled time vector.

3.5.6 Moving Average

The 50Hz and 200Hz recording frequency of the Myo armband [Bernhardt, Paul, 2015b] introduces fair amount of noise in the data. To make the data better suited for classification by machine learning methods, a moving average filter is applied to smooth out the unwanted noise . This noise is also visible when the Myo armband is stationary in space, thus giving a strong indication that the noise originates from the Myo itself. The moving average filter was chosen over a filter in the frequency domain as it can be adapted to live data for future work without the need to perform repeated Fourier transforms as data points iterates through the test sequence. A test conducted to find the maximum frequency of a subject actively shaking his hand, found the highest frequency achieved to be 8,6HZ. As subjects are told to write in their normal

speed, we assume that the moving average of each sequence does not remove critical information from the data. If we define the test sequence to be of a finite length N_S, and the most resent arrived data point always to occupy the first index i = 1 in our test sequence $S(i)$ for $i = 1, \cdots, N$ then, with only the delay of $\Delta t = (X/2) - 1$ sample periods, a moving average filter of odd length X, can calculate the moving average filtered sequence for every new data point, where the test sequence occupies indexes $i = 2, \cdots, N+1$. This simple moving average filter for X = 3 is given by:

$$S_{M3}(i) = \frac{1}{3}\sum(S(i-1) + S(i) + S(i+1)) \tag{3.6}$$

3.5.7 Sequence Length Normalization

For the averaged raw data to be used in machine learning algorithms, the sequences need to be normalized in length. This is achieved by re-sampling each signal to the length of the longest recorded sequence across all recorded letters. Since it is unknown if it is possible to classify individual letters from each other based on the Myo data, we are reluctant to perform dimensionality reduction in the initial stages of classification. Therefore, the choice was made to include all data in the classification approach. This was done by first stretching the length of each sequence to a length which is greater than the longest recorded sequence. The reason for this is that we can then re-sample all signals without loosing information. The resampling are done in an inelegant and effective way in Matlab. Given a signal **B** of length 85, which are to be stretched to the desired length 100, the new, stretched signal $\mathbf{B_{new}}$ is given by:

```
4
5       B_new = B(1:0.85:end);
6
```

[Eamonn J. and Mueen, 2016]The implication of the code above is that some data points, in this case 15, are copied to appear twice right after each other.

The standard length for each letter section for IMU data in this thesis is 100 data point, as it is the largest IMU data sequence recorded of a single letter in across all subjects. Standard length for sEMG is 400 as the sampling frequency i four times as high compared to IMU sampling frequency. This means that for future work we are able to test a classifier, specifically trained on one subject, on data from other subjects.

3.5.8 Z-Normalization of Sequences

The letter sequences extracted in 3.5.3 have to be z-normalized to perform well in dynamic time warping [Eamonn J. and Mueen, 2016], and since both the gyroscope and orientation data are non-stationary, we z-normalize all IMU data. For the letter sequence $\mathbf{S}$, the z-normalized letter sequence $\mathbf{S_z}$ is given by:

$$\mathbf{S_z} = \frac{(\mathbf{S} - \mu_S)}{\sigma_S} \tag{3.7}$$

, where μ_S is the mean of the letter sequence S and σ_S is the standard deviation of S.

3.6 Discrimination in Signal Characteristics

3.6.1 Omitting Unpromising Sensor for Further Study

To avoid processing unnecessary amounts of data in a classification scenario, it would be advantageous if some of the sensors turned out to be redundant. This could be the case if some of the eight sEMG-sensors were located at areas above muscles in the forearm which are inactive during the writing procedure. Another occurrence of redundant sensors would be if one of the accelerometer or gyroscope sensor where to only output constant values. A third possibility is that some of the sensor data has low signal-to-noise ratio, thus leaving the data useless for the study at hand.

3.6.2 Additional Signal Realignment

The cross correlation function is an effective tool for measuring similarity of signals which may be shifted in comparison to each other. In this section the cross correlation function is applied to a set of ten signals, each signal originating from the same subject and the same sensor. The the cross correlation is applied in a one-to-all fashion, where the first signal is cross correlated against the nine remaining signals in the set, in addition to an auto correlation which is effectively a cross correlation with itself. The cross correlation results are then normalized with respect to the max value from the auto correlation. The positions of the max value of each cross correlation is then subtracted from the position of the max value from the auto correlation. The result from this operation is then used to adjust the start and stop time stamps described in

the letter extraction section 3.5.3.

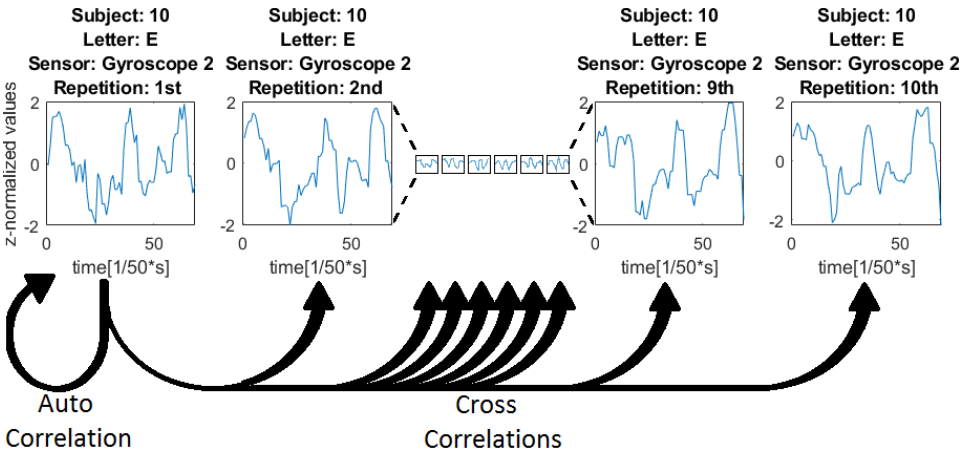


Figure 3.9: The first of the ten signals are auto correlated, then cross correlated against the remaining nine signals.
Source: Personal collection (produced in Matlab).

figure 3.9 illustrates the operation where the first of ten signals, originating from the same subject, sensor and letter, are cross correlated with first itself, then the remaining 9 signals.

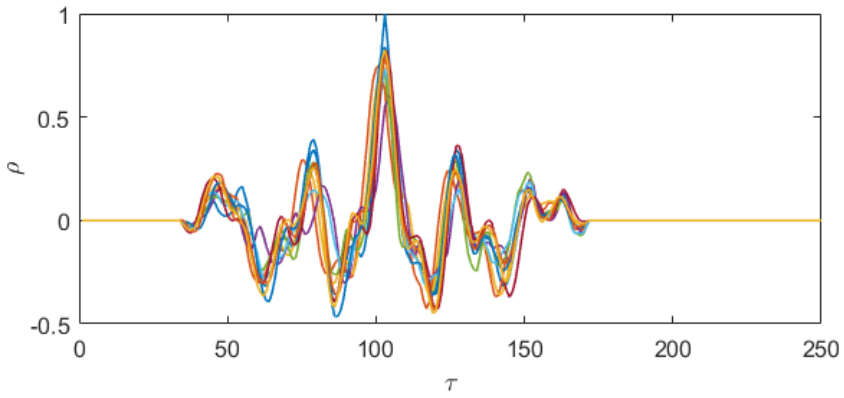


Figure 3.10: Auto correlation and cross correlations plotted against each other before additional realignment.
Source: Personal collection (produced in Matlab).

From figure 3.10 we see the 10 cross correlation results plotted on top of each other. As evident from the respective figure, there are some misalignment of the maximum values in respect to τ. For this particular case, the second and fourth signal has peak values at $\tau_2 - \tau_1 = -2$ and $\tau_4 - \tau_1 = 2$. These values are

used to correct the extraction window for letter repetition two and four, such that when cross correlation operation described in figure 3.9 are performed on the adjusted signals, the maximum peak of each cross correlation is aligned to each other. This second iteration of cross correlation is plotted in figure 3.11 below.

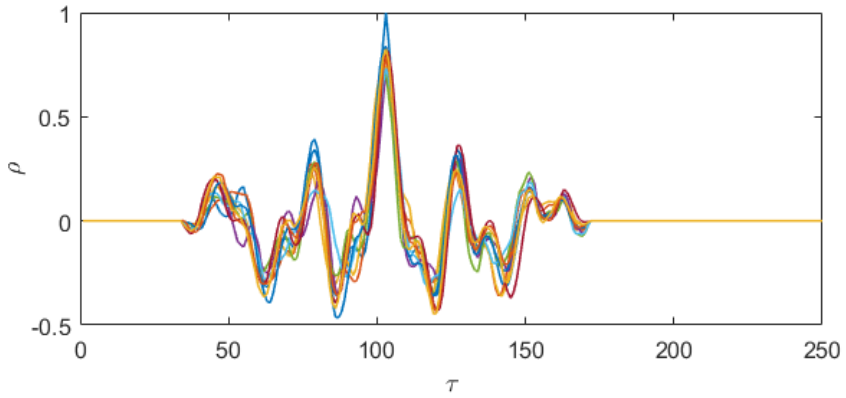


Figure 3.11: Auto correlation and cross correlations plotted against each other, after additional realignment.
Source: Personal collection (produced in Matlab).

The results from the additional realignment described in this section will act as input to the remaining work of this thesis.

4

Results

4.1 Correlation

4.1.1 Method

For simplicity, we rename the 18 sensor data dimensions from accelerometer, gyroscope, orientation and sEMG to:

Table 4.1: Table lists the name conversions for each sensor type and dimension.

Sensor Notation

Senor Type	Dimension/sEMG number	Sensor Number
Accelerometer	X	1
Accelerometer	Y	2
Accelerometer	Z	3
Gyroscope	X	4
Gyroscope	Y	5
Gyroscope	Z	6
Orientation	X	7
Orientation	Y	8
Orientation	Z	9
Orientation	ϕ	10
sEMG	1	11
sEMG	2	12
⋮	⋮	⋮
sEMG	8	18

The results plotted in figures figure 4.1, figure 4.2, figure 4.3 and figure 4.4 originates from the following procedure:

- Consider one sensor dimension.
- The mean sequence within a set of 10 letters originating from one subject are calculated.
- The correlation coefficient is then calculated between the mean sequence and the 10 letter sequences within the set.
- The mean of the ten coefficients is calculated next.
- This process is repeated for all ten sets of that letter type, which originates from five different subjects.
- The mean and standard deviation is then calculated across these ten mean correlation values.
- Repeated for remaining 17 sensor dimensions,

4.1.2 Results

The result of this calculations indicates which sensor dimension that has the highest correlation within subjects for that letter.

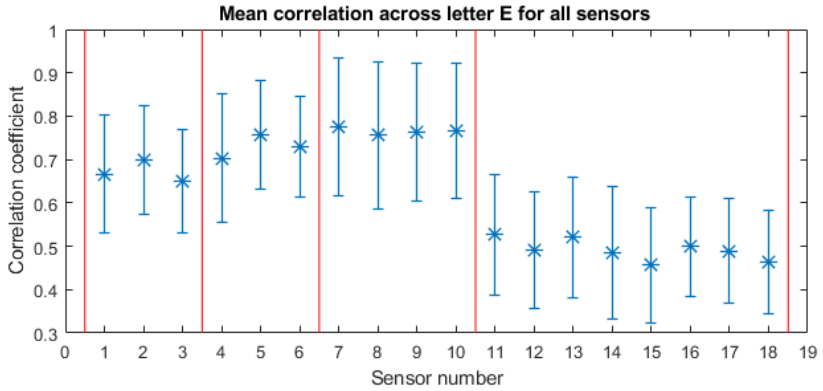


Figure 4.1: Mean and standard deviation of correlation coefficients across all subjects for letter E, for each sensor number.
Source: Personal collection (produced in Matlab).

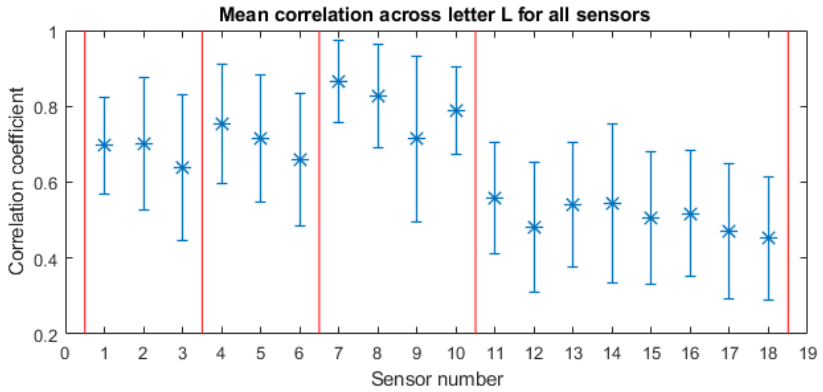


Figure 4.2: Mean and standard deviation of correlation coefficients across all subjects for letter L, for each sensor number.
Source: Personal collection (produced in Matlab).

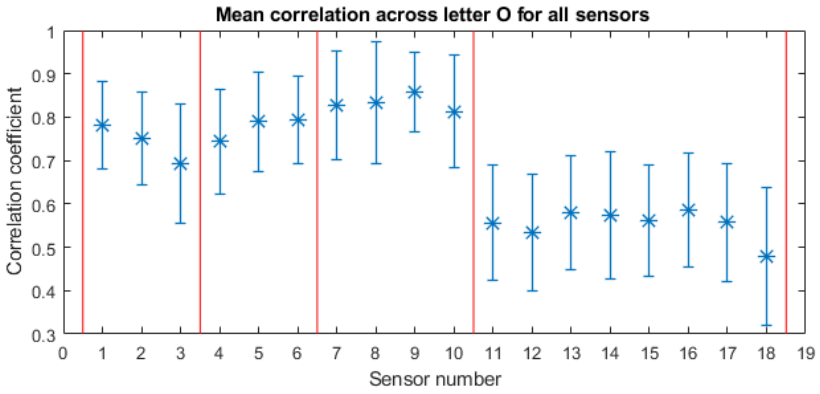


Figure 4.3: Mean and standard deviation of correlation coefficients across all subjects for letter O, for each sensor number.
Source: Personal collection (produced in Matlab).

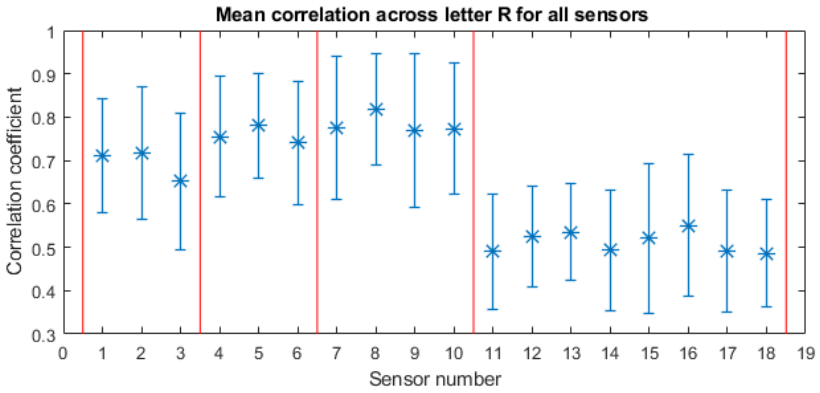


Figure 4.4: Mean and standard deviation of correlation coefficients across all subjects for letter R, for each sensor number.
Source: Personal collection (produced in Matlab).

As seen from the plotted results in figures figure 4.1, figure 4.2, figure 4.3 and figure 4.4, sensor dimensions 7-10 yields the highest mean correlation values. These sensor numbers corresponds to the four orientation dimensions. The mean correlation coefficient for the four orientation dimensions is:

- E = 0.81
- L = 0.81
- O = 0.87
- R = 0.80

, while for the sEMG data, which yields the lowest correlation values in figures 4.1, figure 4.2, figure 4.3 and figure4.4, we have a mean correlation coefficient for all sEMG dimensions of:

- E = 0.52
- L = 0.51
- O = 0.60
- R = 0.54

4.2 Singular Value Decomposition

4.2.1 Method I

In this section we will use SVD to further examine the statistical data similarities for letters written by a given subject. We will also calculate the singular values for the given letters across users to see if there is any similarity in the letter data across subjects.

For one letter sample, each vector representing a sensor number is concatenated such that we end up with vectors of length L, where

$$L = (3 \times 100) + (3 \times 100) + (4 \times 100) + (8 \times 400) = 4200 \tag{4.1}$$

Note that the lengths of all letter sequences are normalized to 100 for IMU-dimensions, and 400 for sEMG-dimensions.

We first calculate the singular values within each letter set. There are 40 sets, 10 sets for each letter, which all consists of data from ten letter repetitions. We calculate the SVD for all sensor data combined as discussed in 4.2.1, and for each of the sensor types, accelerometer, gyroscope, orientation and sEMG, separate.

4.2.2 Results I

Resulting singular values on the diagonal of Σ is normalized by the sum of the diagonal of Σ and plotted in figure 4.5

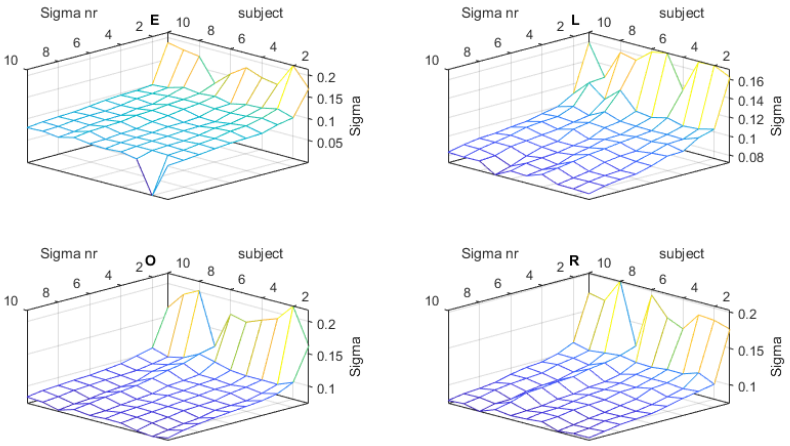


Figure 4.5: [Singular values of subjects across letters.
Source: Personal collection (produced in Matlab).

From figure 4.5 we get a visual representation of the singular value from each subjects letter sets, based on the all the sensor dimensions. The singular values which are calculated based on individual sensor types, e.g. accelerometer data, are listed in table 4.2.

Table 4.2: The table lists mean and standard deviation of singular values within subjects.

Letter	Mean(σ_{11})	Standard deviation(σ_{11})
	All Sensors	
E	0.1705	0.0233
L	0.1579	0.0149
O	0.1783	0.0258
R	0.1731	0.0229
	Accelerometer	
E	0.233	0.0368
L	0.215	0.0380
O	0.258	0.0377
R	0.247	0.0341
	Gyroscope	
E	0.280	0.0488
L	0.244	0.0330
O	0.296	0.0424
R	0.292	0.039
	Orientation	
E	0.337	0.044
L	0.332	0.059
O	0.419	0.093
R	0.330	0.046
	sEMG	
E	0.16	0.017
L	0.151	0.018
O	0.167	0.022
R	0.169	0.016

4.2.3 Method II

To determine the statistical similarities of written letters *across* subjects, a *super matrix* is created containing all the collected data from one letter across all users. When including all sensor numbers in the super matrix for a letter X, the size is 100×4200, where rows 1 to 20 are the twenty letter repetitions from subject 1, rows 21 to 40 are the letter repetitions from subject 2, and so on. There are in total four super matrices, one for each letter. The purpose of these matrices is to explore if their corresponding singular values indicates a

similarity in letters, across subjects.

4.2.4 Results

Table 4.3: Largest singular value along with a summation of the ten largest singular values for each letter. SVD for each letter is calculated from all sensor numbers, across all users.

Singular Values for Super-Matrix		
Letter	σ_1	$\Sigma_{i=1}^{10}\sigma_i$
E	0.033	0.206
L	0.032	0.208
O	0.044	0.217
R	0.035	0.210

We will after calculating the SVD for the super matrices construct four *sensor matrices* which contains individual sensor types, rather than data from all sensors. These are named:

- Super Accelerometer
- Super Gyroscope
- Super Orientation
- Super sEMG

Table 4.4: Largest singular value along with a summation of the ten largest singular values for each letter. SVD for each letter is calculated exclusively from orientation data across all users.

Singular Values for Super-Acceleration Matrix

Letter	σ_1	$\Sigma_{i=1}^{10}\sigma_i$
E	0.047	0.306
L	0.065	0.343
O	0.079	0.340
R	0.057	0.327

Singular Values for Super-Gyroscope Matrix

Letter	σ_1	$\Sigma_{i=1}^{10}\sigma_i$
E	0.060	0.380
L	0.070	0.398
O	0.083	0.423
R	0.075	0.387

Singular Values for Super-Orientation Matrix

Letter	σ_1	$\Sigma_{i=1}^{10}\sigma_i$
E	0.085	0.470
L	0.100	0.515
O	0.145	0.591
R	0.101	0.470

Singular Values for Super-sEMG Matrix

Letter	σ_1	$\Sigma_{i=1}^{10}\sigma_i$
E	0.033	0.206
L	0.032	0.209
O	0.044	0.217
R	0.035	0.210

Table 4.5: Table lists the sum of the 10 first normalized singular values along with the highest singular value.

4.3 K-Nearest Neighbor classification

4.3.1 Method

Sensor data is grouped in four categories based on the sensor type, and data from one letter is concatenated into one vector, such that accelerometer and gyroscope vectors consists of 300 features each, orientation vector consists of 400 features and sEMG vectors holds 3200 features.

For the training of the k-NN classifier, we have, for each subject, two sets of four letter classes with 10 observations in each class. Since the number of observations in our data sets are very low, a cross-validation scheme is implemented, which parts the data into five folds of equal length, where observations are assigned randomly. The classifier uses 4 out of the 5 folds as training data, while the last fold is used as test data. This is repeated 5 times, where a different fold is left out each time. The number of nearest neighbors k was set to 1, 3, and 5, while the distance measure was set to euclidean with equal weighting. The two sets of sampled data from each subject acts as training data and test data, one time each. We will from now assign a number to each of the ten sets such that:

1st subject's 1st set = Set1
1st subject's 2nd set = Set2
2nd subject's 1st set = Set3
⋮
5th subject's 2nd set = Set10

Table 4.6: Notation for Sets

For the notation of the classification results, k-NNX(Y) is the k-nearest neighbor classifier, which is given the task of classifying the data in set Y based on the training data in set X.

The k-NN classifier is first used to classify letters from one subject based on training data form that same object. All the four data types, accelerometer, gyroscope, orientation and sEMG, are used separate to test each type's capability of classification. The full table of these results can be found in Appendix A. In these tables, a classification score of 0 means no correct classifications, while a score of 1 means that all letters of the letter type was classified correctly.

We then go on to splitting all the data in two *bi-sets*. This means that the classifier is fist trained on sets of odd numbers,(see table 4.6), then set to classify sets of even number. This is repeated with opposite training and test

bi-sets.

4.3.2 Results

From the result in Appendix A, its evident that the nearest neighbor rule is capable of classifying data originating from the same subject in most cases. For k-NN1(2) and k-NN2(1), the data originating from subject 1, we have on average a lower classification accuracy, than for the other subjects.

Table 4.7

k-NNX(Y)	1-NN	3-NN	5-NN
Validation Accuracy			
k-NN1	0.925	0.93	0.94
k-NN2(1)	0.935	0.92	0.905
E			
k-NN1(2)	0.58	0.54	0.46
k-NN2(1)	0.6	0.62	0.56
L			
k-NN1(2)	0.58	0.54	0.46
k-NN2(1)	0.6	0.62	0.56
O			
k-NN1(2)	0.78	0.8	0.8
k-NN2(1)	0.84	0.8	0.84
R			
k-NN1(2)	0.78	0.74	0.74
k-NN2(1)	0.78	0.72	0.72
Average			
k-NN1(2)	0.72	0.705	0.69
k-NN2(1)	0.755	0.72	0.705

4.4 Results from Similarity Search with Dynamic Time Warping.

The function output of a DTW between two time series are simply a positive numeric value, where values close to zero indicates high similarity. As knowledge on the position of each individual letter in raw data already is acquired through video analysis, we can test the performance of a sliding DTW function's ability to correctly predict these positions.

4.4.1 Method

In this test we will use DTW to test letter recognition within a subjects data recording. Since we have 18 different sensor dimensions we chose to only use the sensor dimension of greatest correlation within letter sets. This means that for subject 1, where letter E have the highest correlation in sensor 9, we will perform similarity search only between letter sequences and raw data from sensor 9.

The data record from sensor 9 of set 1 from subject 1 is of length 3000. The candidate sequences which are to be tested against the target letter sequence need to be extracted from the recorded data. This is done through the construction of a *candidate matrix* where each column holds a candidate sequence. The candidate sequences overlap each other such that when the mean target letter sequence length is 50, and the recorded signal S is of length 3000, the corresponding candidate matrix has the dimension:

$$(50 \times ((3000/2) - 50 + 1)) = (50 \times 1451) \tag{4.2}$$

This corresponds to the first column of the candidate matrix overlapping the second column with 48 data points. All the columns in the candidate matrix are z-normalized prior to DTW operation. To perform dynamic time warping on the data recording from sensor 9, a mean letter sequence is constructed for the target letter E. This is done by first length normalizing the ten letter sequences. When the ten sequences have the same length, the individual means are subtracted and the sequences are divided by their respective standard deviation. Now the mean of the ten sequences are computed, by averaging.

Similarity search within the data from sensor 9 is done by calculating the DTW distance between the mean letter sequence and every column in the candidate matrix. A warping restriction is set to 10% of the mean letter sequence length.

Since the data recorded from sensor 9 contains ten subsequences corresponding to the writing of letter E, the ten lowest DTW distance values should have positions matching the positions where the letter was written. The success of this approach relies the similarity of letter sequence data, and letter sequence length.

4.4.2 Results

The results for individual cases are plotted in Appendix B. This table consists of both the sensor number that was chosen for each task, and the corresponding success rate of the DTW similarity search. The mean success rate of the DTW classification for each letter is listed below.

- Mean success rate for letter E: 11%
- Mean success rate for letter L: 7%
- Mean success rate for letter O: 11%
- Mean success rate for letter R: 10%

5

Discussion

In this chapter, we will thoroughly discuss the different challenges and aspects of this thesis, along with what should have been done different, and some thoughts on future work.

5.1 Data Collection

5.1.1 Additional Collected Data

Along with the collected data presented in this thesis there are additional data available to the public[1] online. This additional data consists of 5 single sets of 4 (E,L,O,R) by 10 letters , along with complete alphabets written by the respective subject. There is also a gathering of data from 18 subjects, writing the four target letters, one time each. The purpose of this collected data set was to collect data on a broad subject group, The test phase was done in Tromsø Library with permission from the library director. Random persons were asked to participate as anonymous subjects, where data was recorded by the Myo Arband along with a close-up video of the subjects hand movements. Subjects are not identifiable through the video as seen in 5.1

1. From user Brynjulv Tveit at *www.dataverse.no/dataverse/uit*

Figure 5.1: Standardized video angle of subjects hands, in experimental setup type 1.

Each subject were to write the four letters in the following order: E, L, O, and R within the square brackets, as seen in 5.1.The purpose of this experiment was to compare data from single written letters, across subjects. The majority of the time spent on this thesis was used on developing a functioning method for identification of letter sequences in the Myo data, based on recorded video. The rest of the collected data is gathered at University of Tromsø

5.1.2 Omitted Synchonization Method

The first approach letter extraction problem in section 3.5.3, was made while assuming that the orientation data from Myo was calculated from the Myos magnetometer. A conducting coil was coupled with a 9 volt battery together with a visible switch such that the activation of the magnetic field would be visible both in orientation data and in the recorded video. The coil were to produce a magnetic field **B** see figure 5.2, which would be visible in magnetometer data, and thus act like the "synchronization jolt". This would spare the subjects from getting the jolt to their arm.

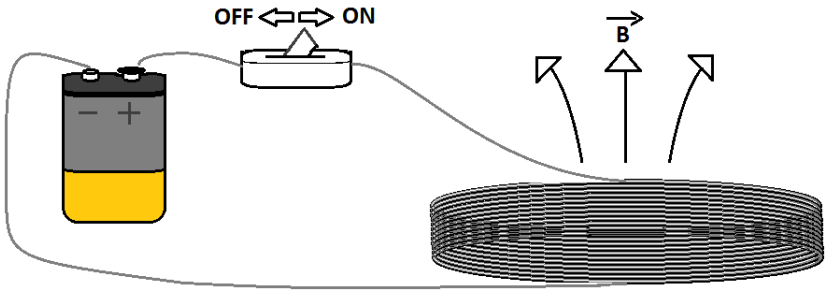


Figure 5.2: An illustration of the device which was initially the primary synchronization tool for alignment of recorded video and Myo data.
Source: Personal collection.

When this method was found unsuccessful, due to the inaccessible magnetometer data, the approach by synchronization jolt was tested and proved sufficiently accurate for the job.

5.1.3 Practical info on Myo Armband

During data collection with Myo in this thesis we experienced some difficulties with delayed start of the data logging. Another problem which caused headache was that the Myo-Data-Capture program fails to record if the battery level of the Myo drops under a certain level. The exact cut-off battery percentage are not known, but one recording session was lost, when the battery level was in the range 35-50 percent.

5.1.4 Alternative to Video Recording

As the method for sequence identification in this thesis is very time consuming, a different approach was attempted, where a microphone recorded the writing of the pen, with a semi-rough and acoustically amplifying surface under the paper. The method showed promising results, but was abandoned when due to the proven accuracy of the already developed video recording method.

5.2 Results

5.2.1 Correlation

The correlation results in this thesis calculated within the individual subjects letters. This was done on an early stage to establish the feasibility of the remaining classification tasks. The orientation data has a mean correlation coefficient of above 0.8 for all letters when calculated within subject sets, while the sEMG have mean correlation coefficients above 0.5, noting that the range of the correlation coefficient is $[-1, 1]$. Although the correlation coefficients within subjects data are high, we need either a strong letter correlation across all users or large amounts of training data to generalize classification such that it will work on test data from *new* subjects. Since we can assume that there are now strong letter correlation from one random subject to another, based on variations in writing techniques and the fact that subjects might use the left hand instead of the right hand, we have to assume that the there is a need for a lot of training data, to achieve letter classification of a new random user.

5.2.2 Singular Value Decomposition

The results from SVD clearly indicates that there exists some similarity between letters across users. The best results were again from the orientation dimensions, where the largest singular values were:

- E = 0.085
- L = 0.100
- O = 0.145
- R = 0.101

This means that the first singular value of letter O along with it's corresponding vectors in matrix U and matrix V, holds 14.5% of the total information regarding letter O, and similar for the letters, E,L and R. Results for all the sensors combined in the *super matrix* yields a lower singular values. The result from SVD in this thesis proves that there is a similarity between the letters across users. The method of using SVD as a proof of statistical similarity between vectors are used with great success in other studies, such as in [Sharma and Alsam, 2014].

5.2.3 Nearest Neighbor Classification

The results from section 4.3, where classifier is trained and tested on data from the same subject, further establishes the similarity of letters with in subjects. The results in table 4.7, where half of all orientation data is used as training data and half as test data, yields promising results, as the k-NN classifier is considered as a simple classifier in general. Since the test data and training data in table4.7 both contains sets from all users, this does not prove the feasibility of classification across users but gives an indication on how much the miss-classification grows when training data from multiple users are included. This is interesting as it is important to know if the recorded sequence of letter X for one subject is similar of identical to the recorded sequence of letter Y for a different subject. Although the classification rate of across all users went down when compared to mean classification within subjects, we still achieve a classification rate significantly higher than the that of 0.25, which would would be the expected success rate of randomly assigning classes to letter sequences.

5.2.4 Dynamic Time Warping

The attempted similarity search using DTW faces one obstacle which the other methods in this thesis is not introduced to. The DTW algorithm in this thesis searches through the *entire* raw file from one sensor dimension, as oppose to the other techniques which are given test queries as input. The difference in these two approaches is that the DTW is is not guarantied to extract the correct query length from the raw data. This will maybe turn out to be the greatest challenge in future studies, as one subject might write twice as fast as another subjects, if not more, if we consider the case were subjects are in the early stage of education.

5.3 Comment on future application methods

5.3.1 Hand Written *Word* Recognition

Consider the scenario where we are not able to classify the written letters from the (English) alphabet into it's 26 distinct classes. Classification of individual words can still be possible, by the means of *disambiguation* [Kreifeldt et al., 1989]. An example of this technique being utilized is in older mobile phones through *text on 9 keys*(T9), or *predictive text*. The same concept could be used in handwriting recognition though Myo, by grouping one letter together with the letters it is most likely to be miss-classified as. Using this technique the

classifier is not dependent of 100% success rate in classification, which in turn increases the chances of a future where handwriting recognition through Myo Armband is reality.

5.3.2 Myo versus Smart Watches

As seen in the results presented in this thesis, the IMU sensors out performs the sEMG sensors on average. Since most smart watches are already embedded with IMU sensors, it is possible that the future of handwriting recognition is in smart watches and not in the Myo Armband. A smart watch is also worn in a consistent position and with a more or less fixed orientation. The watch has also the advantage that the wrist is much harder than the upper forearm, which therefor translates movement to the IMU more accurate. If handwriting recognition though arm mounted IMU sensors exists in the future, it is more applicable in the form of a watch, then for the Myo Armband.

5.4 Conclusion

In this thesis we have explored the Myo Armbands potential as a multi sensor for handwriting recognition. A strong positive correlation between same class letters within subjects has been proven in all of the four sensor types, where the orientation data yields the highest correlation coefficient values, while the sEMG data yields the lowest. Statistical similarity between same class letters has been found through singular value decomposition, where again orientation data yields the highest values, while sEMG scores the lowest of all sensor types. In an attempt to cross subject classification though k-NN, with $k = 1$, $k = 3$, and $k = 5$, the 1-NN classifier yields a minimum success rate of 58% across the four letters. This is considerably better that what we would expect from a random assignment of letter classes. In the last part of the results, a similarity search by DTW is attempted. This yield poor results, with a classification success rate of around 10% on average across letters. Since the data from IMU sensors over all out performs the data from the sEMG, we suggest that future studies on this subject of handwriting recognition by arm mounted multi sensors are performed with IMU sensors mounted on the wrist, instead of on the upper forearm, as this would result in larger arm movements, hence larger amplitudes in the recorded IMU data. As the IMU sensors are embedded in a vast variety of smart watches, this seems like a more realistic approach to the problem, as the data collection can be done more effectively. If future work succeeds in hand writing recognition through wrist mounted IMU sensors, a hypothetical handwriting recognition-app would be accessible to a larger number of users.

APPENDIX A

k-NN Classification Results for sEMG Data

Validation accuracy for training data			
	K=1	K=3	K=5
k-NN1(2)	0.75	0.775	0.825
k-NN2(1)	0.775	0.8	0.8
k-NN3(4)	0.55	0.5	0.45
k-NN4(3)	0.775	0.65	0.625
k-NN5(6)	0.875	0.75	0.85
k-NN6(5)	0.85	0.85	0.85
k-NN7(8)	0.875	0.875	0.825
k-NN8(7)	0.825	0.825	0.825
k-NN9(10)	0.8	0.875	0.8
k-NN10(9)	0.8	0.875	0.925

Classification of letter **E**			
	K=1	K=3	K=5
k-NN1(2)	0.6	0.9	0.9
k-NN2(1)	0.3	0.7	0.8
k-NN3(4)	1.0	1.0	1.0
k-NN4(3)	1.0	1.0	1.0
k-NN5(6)	0.8	0.8	0.8
k-NN6(5)	0.8	0.8	0.9
k-NN7(8)	0.9	0.9	0.9
k-NN8(7)	0.9	0.9	1.0
k-NN9(10)	0.9	1.0	1.0
k-NN10(9)	1.0	1.0	1.0

Classification of letter **L**			
	K=1	K=3	K=5
k-NN1(2)	0.6	1.0	0.9
k-NN2(1)	0.8	0.8	0.6
k-NN3(4)	0.4	0.3	0.4
k-NN4(3)	0.6	0.5	0.1
k-NN5(6)	0.5	0.7	0.7
k-NN6(5)	0.3	0.3	0.2
k-NN7(8)	0.4	0.4	0.4
k-NN8(7)	0.5	0.6	0.5
k-NN9(10)	0.6	0.6	0.6
k-NN10(9)	0.6	0.7	0.6

Table 5.1: Table lists values for validation accuracy for classifiers trained on data from all sEMG dimensions, and the classification accuracy on test sets for the letters E and L, respectively.

Classification of letter **O**			
	K=1	K=3	K=5
k-NN1(2)	0.2	0.1	0
k-NN2(1)	0.2	0	0
k-NN3(4)	0.7	0.7	0.7
k-NN4(3)	0.2	0	0
k-NN5(6)	1.0	1.0	1.0
k-NN6(5)	1.0	1.0	1.0
k-NN7(8)	1.0	1.0	1.0
k-NN8(7)	1.0	1.0	0.9
k-NN9(10)	0.8	0.9	1.0
k-NN10(9)	0.8	1.0	1.0

Classification of letter **R**			
	K=1	K=3	K=5
k-NN1(2)	0.1	0	0
k-NN2(1)	0.2	0.1	0.2
k-NN3(4)	0.9	0.7	0.7
k-NN4(3)	0.9	0.9	0.8
k-NN5(6)	1.0	1.0	1.0
k-NN6(5)	0.9	1.0	1.0
k-NN7(8)	0.9	1.0	0.9
k-NN8(7)	0.9	0.8	0.9
k-NN9(10)	0.8	0.8	0.8
k-NN10(9)	0.9	0.7	0.8

Mean results across all 4 letters.			
	K=1	K=3	K=5
k-NN1(2)	0.375	0.5	0.45
k-NN2(1)	0.375	0.4	0.4
k-NN3(4)	0.75	0.675	0.7
k-NN4(3)	0.675	0.6	0.475
k-NN5(6)	0.825	0.875	0.875
k-NN6(5)	0.75	0.775	0.775
k-NN7(8)	0.8	0.825	0.8
k-NN8(7)	0.825	0.825	0.825
k-NN9(10)	0.775	0.825	0.85
k-NN10(9)	0.825	0.85	0.85

Table 5.2: Table lists the sEMG data classification accuracy on test sets for the letters O and R, respectively, and the mean

k-NN Classification Results for Accelerometer Data.

Validation accuracy for training data			
	K=1	K=3	K=5
k-NN1	0.9	0.925	0.95
k-NN2	0.9	0.9	0.875
k-NN3	1.0	0.975	1.0
k-NN4	1.0	1.0	0.975
k-NN5	0.825	0.9	0.8
k-NN6	0.875	0.825	0.85
k-NN7	0.875	0.85	0.85
k-NN8	0.925	0.825	0.875
k-NN9	0.875	0.95	0.9
k-NN10	0.9	0.875	0.875

Classification of letter **E**			
	K=1	K=3	K=5
k-NN1(2)	0.8	0.8	0.8
k-NN2(1)	0.9	0.9	0.9
k-NN3(4)	1.0	1.0	1.0
k-NN4(3)	1.0	1.0	1.0
k-NN5(6)	1.0	1.0	1.0
k-NN6(5)	0.7	0.5	0.5
k-NN7(8)	0.9	0.9	1.0
k-NN8(7)	0.7	0.9	0.8
k-NN9(10)	1.0	1.0	0.9
k-NN10(9)	1.0	1.0	1.0

Classification of letter **L**			
	K=1	K=3	K=5
k-NN1(2)	0.8	0.7	0.7
k-NN2(1)	0.6	0.6	0.6
k-NN3(4)	1.0	1.0	1.0
k-NN4(3)	1.0	1.0	1.0
k-NN5(6)	0.6	0.5	0.7
k-NN6(5)	0.9	1.0	1.0
k-NN7(8))	0.2	0.3	0.3
k-NN8(7)	0.8	0.4	0.3
k-NN9(10)	0.8	0.6	0.6
k-NN10(9)	1.0	0.8	0.9

Table 5.3: Table lists values for validation accuracy for classifiers trained on data from accelerometer dimensions, and the classification accuracy on test sets for the letters E and L, respectively.

Classification of letter **O**			
	K=1	K=3	K=5
k-NN1(2)	0.2	0.2	0
k-NN2(1)	0.2	0.2	0
k-NN3(4)	1.0	1.0	1.0
k-NN4(3)	1.0	1.0	1.0
k-NN5(6)	0.9	0.9	0.9
k-NN6(5)	0.7	0.8	0.8
k-NN7(8)	1.0	1.0	1.0
k-NN8(7)	1.0	1.0	1.0
k-NN9(10)	0.8	0.8	1.0
k-NN10(9)	1.0	0.8	1.0

Classification of letter **R**			
	K=1	K=3	K=5
k-NN1(2)	0.1	0.1	0.1
k-NN2(1)	0.2	0.1	0.1
k-NN3(4)	01.0	1.0	1.0
k-NN4(3)	1.0	1.0	1.0
k-NN5(6)	0.9	1.0	0.8
k-NN6(5)	1.0	1.0	1.0
k-NN7(8)	1.0	0.8	1.0
k-NN8(7)	0.9	1.0	1.0
k-NN9(10)	1.0	1.0	1.0
k-NN10(9)	0.9	0.9	0.9

Mean results across all 4 letters.			
	K=1	K=3	K=5
k-NN1(2)	0.475	0.45	0.4
k-NN2(1)	0.475	0.45	0.4
k-NN3(4)	1.0	1.0	1.0
k-NN4(3)	1.0	1.0	1.0
k-NN5(6)	0.85	0.85	0.85
k-NN6(5)	0.825	0.825	0.825
k-NN7(8)	0.775	0.75	0.825
k-NN8(7)	0.85	0.825	0.775
k-NN9(10)	0.9	0.85	0.875
k-NN10(9)	0.975	0.875	0.95

Table 5.4: Table lists the accelerometers classification accuracy on test sets for the letters O and R, respectively, and the mean cross all letters.

k-NN Classification Results for Gyroscope Data

Validation accuracy for training data			
	K=1	K=3	K=5
k-NN1	0.925	0.925	0.925
k-NN2	0.875	0.95	0.925
k-NN3	1.0	1.0	0.975
k-NN4	0.975	0.975	1.0
k-NN5	0.9	0.825	0.825
k-NN6	0.95	0.95	0.975
k-NN7	0.95	0.95	0.975
k-NN8	0.95	0.85	0.9
k-NN9	0.975	1.0	0.975
k-NN10	0.975	0.95	0.95

Classification of letter **E**			
	K=1	K=3	K=5
k-NN1(2)	0.6	0.7	0.7
k-NN2(1)	0.6	0.4	0.4
k-NN3(4)	0.8	0.8	0.8
k-NN4(3)	1.0	1.0	1.0
k-NN5(6)	1.0	0.9	0.9
k-NN6(5)	0.8	0.7	0.6
k-NN7(8)	1.0	1.0	1.0
k-NN8(7)	1.0	1.0	1.0
k-NN9(10)	1.0	1.0	1.0
k-NN10(9)	1.0	1.0	1.0

Classification of letter **L**			
	K=1	K=3	K=5
k-NN1(2)	0.6	0.4	0.4
k-NN2(1)	0.7	0.7	0.7
k-NN3(4)	0.9	0.9	0.7
k-NN4(3)	1.0	1.0	1.0
k-NN5(6)	0.8	0.8	0.9
k-NN6(5)	1.0	1.0	1.0
k-NN7(8)	0.8	0.8	0.8
k-NN8(7)	0.9	0.8	0.9
k-NN9(10)	0.6	0.7	0.7
k-NN10(9)	0.9	0.8	0.8

Table 5.5: Table lists values for validation accuracy for classifiers trained on data from gyroscope dimensions, and the classification accuracy on test sets for the letters E and L, respectively.

Classification of letter **O**			
	K=1	K=3	K=5
k-NN1(2)	0.6	0.4	0.3
k-NN2(1)	0.5	0.5	0.4
k-NN3(4)	1.0	1.0	1.0
k-NN4(3)	1.0	1.0	1.0
k-NN5(6)	1.0	1.0	1.0
k-NN6(5)	1.0	1.0	1.0
k-NN7(8)	1.0	1.0	1.0
k-NN8(7)	1.0	1.0	1.0
k-NN9(10)	1.0	1.0	1.0
k-NN10(9)	0.9	1.0	1.0

Classification of letter **R**			
	K=1	K=3	K=5
k-NN1(2)	0.3	0.3	0.4
k-NN2(1)	0	0	0
k-NN3(4)	1.0	1.0	1.0
k-NN4(3)	1.0	1.0	1.0
k-NN5(6)	0.9	0.9	0.7
k-NN6(5)	1.0	1.0	1.0
k-NN7(8)	1.0	1.0	1.0
k-NN8(7)	0.9	0.8	0.8
k-NN9(10)	1.0	1.0	1.0
k-NN10(9)	1.0	1.0	1.0

Mean results across all 4 letters.			
	K=1	K=3	K=5
k-NN1(2)	0.525	0.45	0.45
k-NN2(1)	0.45	0.4	0.375
k-NN3(4)	0.925	0.925	0.875
k-NN4(3)	1.0	1.0	1.0
k-NN5(6)	0.925	0.9	0.875
k-NN6(5)	0.95	0.925	0.9
k-NN7(8)	0.95	0.95	0.95
k-NN8(7)	0.95	0.9	0.925
k-NN9(10)	0.9	0.925	0.925
k-NN10(9)	0.95	0.95	0.95

Table 5.6: Table lists the gyroscope data classification accuracy on test sets for the letters O and R, respectively, and the mean

k-NN clustering using orientation data.

Validation accuracy for training data			
	K=1	K=3	K=5
KNN1(2)	0.9	0.9	0.95
KNN2(1)	0.925	0.925	0.9
KNN3(4)	0.975	0.975	0.975
KNN4(3)	0.975	0.95	0.975
KNN5(6)	0.9	0.9	0.875
KNN6(5)	0.95	0.975	0.925
KNN7(8)	0.925	0.975	0.975
KNN8(7)	0.975	0.925	0.9
KNN9(10)	0.975	0.975	0.975
KNN10(9)	0.875	0.9	0.95

Classification of letter **E**			
	K=1	K=3	K=5
KNN1(2)	0	0	0
KNN2(1)	0.4	0.4	0.3
KNN3(4)	0.2	0.2	0.1
KNN4(3)	0.5	0.4	0.6
KNN5(6)	1.0	0.8	0.9
KNN6(5)	0.7	0.9	0.7
KNN7(8)	0.9	1.0	1.0
KNN8(7)	0.9	0.9	0.9
KNN9(10)	1.0	1.0	1.0
KNN10(9)	1.0	1.0	1.0

Classification of letter **L**			
	K=1	K=3	K=5
KNN1(2)	0	0	0
KNN2(1)	0.1	0.1	0.1
KNN3(4)	1.0	1.0	0.9
KNN4(3)	1.0	1.0	1.0
KNN5(6)	0.8	0.8	0.8
KNN6(5)	0.8	0.9	0.9
KNN7(8)	0.8	0.8	0.8
KNN8(7)	1.0	1.0	1.0
KNN9(10)	0.7	0.8	0.8
KNN10(9)	1.0	1.0	1.0

Table 5.7: Table lists values for validation accuracy for classifiers trained on data from orientation dimensions, and the classification accuracy on test sets for the letters E and L, respectively.

Classification of letter **O**			
	K=1	K=3	K=5
KNN1(2)	0	0	0
KNN2(1)	0.4	0	0.1
KNN3(4)	1.0	1.0	1.0
KNN4(3)	0.9	0.9	1.0
KNN5(6)	0.8	0.8	0.8
KNN6(5)	1.0	1.0	1.0
KNN7(8)	1.0	1.0	1.0
KNN8(7)	1.0	1.0	1.0
KNN9(10)	1.0	1.0	1.0
KNN10(9)	1.0	1.0	1.0

Classification of letter **R**			
	K=1	K=3	K=5
KNN1(2)	0.5	0.5	0.5
KNN2(1)	0.1	0.1	0.1
KNN3(4)	0.8	0.9	0.9
KNN4(3)	1.0	1.0	1.0
KNN5(6)	0.9	0.9	0.8
KNN6(5)	0.9	0.9	0.9
KNN7(8)	1.0	1.0	1.0
KNN8(7)	0.9	1.0	1.0
KNN9(10)	0.9	0.9	0.8
KNN10(9)	1.0	1.0	1.0

Mean results across all 4 letters.			
	K=1	K=3	K=5
KNN1(2)	0.125	0.125	0.125
KNN2(1)	0.25	0.15	0.15
KNN3(4)	0.75	0.775	0.725
KNN4(3)	0.85	0.825	0.9
KNN5(6)	0.875	0.825	0.825
KNN6(5)	0.85	0.925	0.875
KNN7(8)	0.925	0.95	0.95
KNN8(7)	0.95	0.975	0.975
KNN9(10)	0.9	0.925	0.9
KNN10(9)	1.0	1.0	1.0

Table 5.8: Table lists the orientation data classification accuracy on test sets for the letters O and R, respectively, and the mean

APPENDIX B

Set 1

Letter	Sensor of Max(ρ)	Classified
E	9	2
L	7	3
O	1	3
R	4	3

Set 2

Letter	Sensor of Max(ρ)	Classified
E	2	1
L	7	0
O	9	0
R	9	2

Set 3

Letter	Sensor of Max(ρ)	Classified
E	8	1
L	4	0
O	7	0
R	1	1

Set 4

Letter	Sensor of Max(ρ)	Classified
E	7	1
L	4	2
O	7	6
R	5	0

Set 5

Letter	Sensor of Max(ρ)	Classified
E	3	0
L	10	0
O	10	0
R	8	0

Set 6

Letter	Sensor of Max(ρ)	Classified
E	5	1
L	7	0
O	7	0
R	8	3

Set 7

Letter	Sensor of Max(ρ)	Classified
E	10	0
L	10	0
O	7	1
R	10	0

Set 8

Letter	Sensor of Max(ρ)	Classified
E	10	1
L	8	1
O	7	1
R	7	0

Set 9

Letter	Sensor of Max(ρ)	Classified
E	5	3
L	8	0
O	9	0
R	8	0

Set 10

Letter	Sensor of Max(ρ)	Classified
E	4	1
L	8	1
O	9	0
R	5	1

Bibliography

[Str, 2006] (2006). *Linear algebra and its applications*. Thomson, 4th ed. edition.

[Bernhardt, Paul, 2015a] Bernhardt, Paul (2015a). *Myocraft - EMG in the bluetooth protocol*. from `http://developerblog.myo.com/myocraft-emg-in-the-bluetooth-protocol/` .

[Bernhardt, Paul, 2015b] Bernhardt, Paul (2015b). *Myocraft - Logging IMU and Raw EMG Data*. from `http://developerblog.myo.com/myocraft-logging-imu-and-raw-emg-data/` .

[Box et al., 2007] Box, G. E. P., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2007). *Time Series Analysis : Forecasting and Control (4th Edition)*. Somerset, NJ, USA: John Wiley and Sons, Inc, Hoboken, New Jersey, USA.

[Cover, T., and Hart, P., 1967] Cover, T., and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27.

[Eamonn J. and Mueen, 2016] Eamonn J., K. and Mueen, A. (2016). Extracting Optimal Performance from Dynamic Time Warping.

[Fried, Carrie B., 2008] Fried, Carrie B. (2008). In-class laptop use and its effects on student learning. *Computers and Education*, 50(3):906–914.

[H. Sakoe and S. Chiba, 1978] H. Sakoe and S. Chiba (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49.

[Kreifeldt et al., 1989] Kreifeldt, J. G., Levine, S. L., and Iyengar, C. (1989). Reduced keyboard designs using disambiguation. *Proceedings of the Human Factors Society Annual Meeting*, 33(6):441–444.

[Müller, Meinard, 2007] Müller, Meinard (2007). *Information Retrieval for Music and Motion*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Sharma and Alsam, 2014] Sharma, P. and Alsam, A. (2014). A robust metric for the evaluation of visual saliency models. In *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 2, pages 654–661.

[Thalmic Labs, 2018a] Thalmic Labs (2018a). urlhttps://www.myo.com/techspecs.

[Thalmic Labs, 2018b] Thalmic Labs (2018b). urlhttps://support.getmyo.com/hc/en-us/articles/202641403-Battery-life-of-the-Myo-armband.

[Theodoridis and Koutroumbas, 2008] Theodoridis, S. and Koutroumbas, K. (2008). *Pattern Recognition 4th Edition*. Burlington: Elsevier Science.