DEVELOPMENT OF A FAST AND EFFICIENT ALGORITHM FOR P300 EVENT RELATED POTENTIAL DETECTION

A Thesis
Submitted to
the Temple University Graduate Board

In Partial Fulfillment
of the Requirements for the Degree
MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

---

by
Elliot Franz
May 2014

Thesis Approval(s):

Iyad Obeid, PhD., Thesis Advisor, Electrical and Computer Engineering
Joseph Picone, PhD., Electrical and Computer Engineering
Andrew Spence, PhD., Bioengineering

Abstract

Electroencephalography (EEG) is an over one hundred year old technique which refers to the recording of electrical potentials in the brain. The most commonly used type of EEG is surface EEG, where electrical potentials from the brain are recorded from electrodes on the surface of the scalp. Recently, a number of consumer grade wireless EEG headsets have been developed. These headsets allow users to access recorded electrical potentials from the scalp. One application of this technology has been 'brain training,' in which users play video games designed to promote activity in certain regions of the brain in order to improve concentration and problem solving ability. Another application of the wireless headsets is for use in Brain-Computer Interfaces (BCI). The goal of BCI research is to provide an individual (e.g., an individual with paralysis or another impairment which limits functionality) with 'mind-control' of a computer. EEG recorded from the scalp has been an area of interest in the BCI community for some time. One type of BCI which can use surface EEG is the P300 Event Related Potential (ERP) BCI. This BCI employs a 6×6 matrix of alphanumeric characters to spell words based solely on input from the brain. The P300 ERP signal is elicited from the user when rows and columns of the matrix containing the target character are flashed. The ERPs which are elicited to spell words in this type of setup are notoriously difficult to detect due to the large amount of noise in the signal. The goal of this research is to optimize the detection of P300 potentials using the EPOC from Emotiv Systems, a consumer grade wireless EEG headset. This work compares the Emotiv EPOC directly with a high grade EEG collection system (the Neurodata 12 Acquisition System from Grass Technologies) by recording signals from spelling sessions in parallel. This research also presents a novel algorithm for optimizing P300 spelling speed to improve the throughput of a P300 based BCI speller. Increasing speed is an important concern for any future mobile application of the BCI technology (e.g., a tablet), because battery life and processor capabilities are limited. Increasing speed is synonymous with decreasing computational complexity, decreasing processor load, and increasing battery life.

Dedication

This project is dedicated first to the Lord and then to my wife, Tara, for the sunshine.

Acknowledgments

Thanks to Dr. Iyad Obeid and the committee for useful project advice, Andrew Williams for assistance in collection of data, and the Electrical Engineering Department of Temple University for funding.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

There has been a recent surge of interest in low-cost, consumer EEG headsets [1, 2, 3]. One of the more popular headsets is the EPOC from Emotiv Systems (San Francisco, California), a wireless headset [4]. Since the EPOC is wireless, it can be linked with tablets or other mobile devices. Processing brain signals locally on mobile devices could make BCIs (Brain-Computer Interfaces) not only portable, but also more widely available and affordable. One method, which is currently used in research, to control a device using brain activity, is the Oddball Paradigm and the P300 signal [5]. Typically, a user is shown a random item, if that item is a 'target' item to the user, then a small (approximately 5 μV) peak, known as the P300, is visible about 300 ms after the stimulus. Various studies have exploited this phenomenon to spell words, for example, in a P300 speller implementation [6].

A P300 speller consists of a 6×6 character and number array and an EEG acquisition system. The columns and rows of the matrix are flashed at random to the user. If a row or column contains the user's target character, a P300 is triggered [7]. By determining the row and column focused on by the user, it is possible to determine which character was the target. A challenge has been to implement an algorithm which is capable of classifying user inputs at an increased rate without sacrificing accuracy. The throughput of a modern P300 speller is quite low (about 1 character per minute [8]), since many trials must be averaged in order to eliminate noise.

The problem of noise is even worse for consumer headsets like the EPOC, which have poorer signal quality than medical-grade acquisition systems [1]. In addition to the system resources required for averaging, processing power is also needed to run a classification algorithm. The goal of this research is to do the following without sacrificing classification accuracy: decrease the stimulus to classification time of the P300 speller, and decrease the amount of EEG data processing so that it can be carried out on resource-limited mobile platforms.

A solution to this problem is a 'first pass' probabilistic classifier. After a low number of rounds, row and column signals will be averaged and input to the classifier. The classifier will then eliminate the rows and columns which it has determined have a low probability of containing the P300. Those rows and columns will not be presented to the user in future rounds of a trial.

Determining which probabilistic classifier to use is the first challenge. This must be done in tandem with the second challenge, which is to determine how many rounds are needed for the classifier so that it can confidently distinguish low probability rows and columns from high probability rows and columns. Finally, the remaining rows and columns which were not eliminated by the first probabilistic classifier must be classified. This hypothesis will be tested by recording EEG data from the EPOC device using OpenViBE (see Section 3.2.1). Analysis of the data will be done in MATLAB, using the EEGLAB toolbox. For validation purposes, experiments will also be performed on publicly available EEG data from the BCI 2003 Competition.

1.2 Background

This section is intended to give the reader who may be unfamiliar with BCIs, neurophysiology, or biological signals, a framework for understanding this project. The following subsections are not intended to be in-depth reviews of the literature. Please refer to the 'Literature Review,' section of this proposal for a more focused review (Section 2.1).

1.2.1 Brain-Computer Interfaces

Brain-Computer Interfaces (BCIs) are systems which interpret a user's brain activity with the goal of controlling an action or set of actions. More specifically, BCIs give their users communication and control channels which replace or augment the normal control channels of the human body (e.g., nerves and muscles) [9]. On a medical level, BCIs can enable individuals who have been dismembered or paralyzed to once again, or for the first time, gain brain control of their appendages.

On an academic level, BCI research provides insight into the complex workings of the

human brain. The brain contains around 100 billion neurons [10], and there are billions of interconnections between those neurons, making the brain an incredibly complex system. The challenge for BCI research has been to interpret brain activity, which is inherently complex, with limited detector resolution. For example, moving a pinky finger involves millions of neurons firing in a precise order. If a BCI were able to detect the precise order of individual neuron activation which corresponds to the finger movement, then computer controlled mechanical parts could be programmed to duplicate the finger's movement exactly. This 'best case scenario' has not yet been attained for many reasons not limited to: low detector resolution, inter-subject variability, and hardware limitations.

All Brain-Computer Interfaces have the same general architecture. Figure 1 is a block diagram of a standard BCI. There is a user (red block), an acquisition system, a data collector, a processor, a classifier, and an action. The user is able to see, hear, feel, or otherwise observe the action of the computer, thus completing the feedback loop. Every block in Figure 1 which is not the user block can be varied. The acquisition system, for example, does not have to be a surface Electroencephalogram (EEG), as it is in many BCIs. Instead, the acquisition system could be an functional MRI (fMRI) brain scan machine or an implanted EEG electrode array.

BCI implementations can be categorized based on the type of brain signal acquisition system. The acquisition systems can be either invasive or non-invasive. For a BCI to be defined as 'invasive,' it must use an implanted electrode array as its acquisition system. An implanted array offers better signal quality (i.e., less noise) and better spatial signal density, but a risky surgical procedure is needed for the implantation of the array. It is widely believed that invasive BCIs have the highest potential for improvement [11] since small bunches of neurons can be probed with micro-electrodes, thereby greatly increasing the signal resolution. Non-invasive BCIs use surface EEG to acquire brain signals. Surface EEG can provide enough neural information to afford patients simple communication capability, but the bandwidth of such signals is limited. Since this project focuses on a wireless EEG headset, non-invasive systems will be the focus for the remainder of this discussion. (Note:

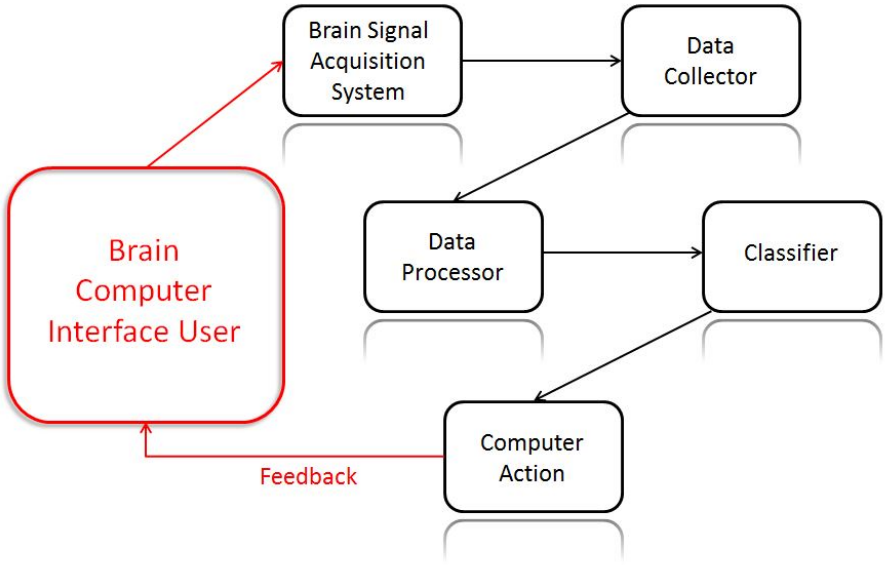

Figure 1: Basic overview of a BCI

Although fMRI and other scanning systems would be considered 'non-invasive,' they are not commonly used due to high cost. For the purposes of this review, the term 'non-invasive' shall refer exclusively to EEG-based BCIs.) [11]

There is wide variety within the subset of non-invasive BCIs. For example, referring to Figure 1, the type of EEG electrode cap ('Data Collector'), the type of processor ('Data Processor'), the classification algorithm(s) ('Classifier'), and the final goal of the BCI ('Computer Action'), all vary from one implementation to another. Sometimes there can be constraints on these blocks (e.g., limited data processing capability). The task of extracting meaningful data from EEG traces is the job of the pre-preprocessing and classification algorithms.

EEG data must be processed before anything meaningful can be extracted from it. Depending on the number of electrodes used, the spatial resolution of EEG is limited, at best. Even if the 'best case scenario' of several hundred electrodes with a very high signal to noise ratio could be realized, the problem of resolution would still persist. This is because EEG signals represent the combined electrical activity of massive neuronal populations

filtered through bone, muscle, and skin [11].

That said, there has been much success in the field of non-invasive BCI when it has been used for simple control and communication tasks [9, 12, 13, 11]. The signals of interest are found in either the time-domain or the frequency-domain. Sometimes a combination of both time-domain and frequency-domain features are used in a BCI. For example, it is known that the ratio of Beta (15-30 Hz) to Theta (4-7 Hz) brainwaves (i.e., the ratio of their respective signal powers) is a measure of attention. Beta/Theta based neurofeedback has been used for over thirty years as a means of treating Attention Deficit Disorder [14]. Tracking eye blinks in addition to the ratio of Beta to Theta frequency content as a measure of drowsiness could improve the quality of the feedback. Examples of useful time-domain signals include the following: visual evoked potentials, slow cortical potentials, or P300 event related potentials [12]. The next section will discuss the advantages and disadvantages of using P300 for a BCI (more specifically, a word spelling BCI) over another type of signal.

1.2.2 P300 Event Related Potential

It has been over forty years since the discovery of the P300 event related potential (ERP) [15]. Its discovery was largely due to advances in signal averaging of brain waves. Oftentimes, the term 'P300,' also known as P3, P3a, P3b, or even LPC (Late Positive Component), is associated with a protocol known as the Oddball Paradigm, previously mentioned in Section 1.1. The Oddball Paradigm, however, is only one technique for eliciting a P300 response. Figure 2 depicts three protocols which can be used to elicit a P300 response from a subject.

The first protocol in Figure 2, known as the 'Single-Stimulus,' is performed by infrequently presenting the subject with a target. When the target is observed, a P300 potential can be detected. The second protocol is the Oddball Paradigm. In the case of this protocol, recordings contain both target and non-target traces, allowing for simple yes/no discrimination. When a stimulus (e.g., a character) is presented to the subject, the BCI does not 'know' if that character is a target or a non-target. This makes the Oddball Paradigm a

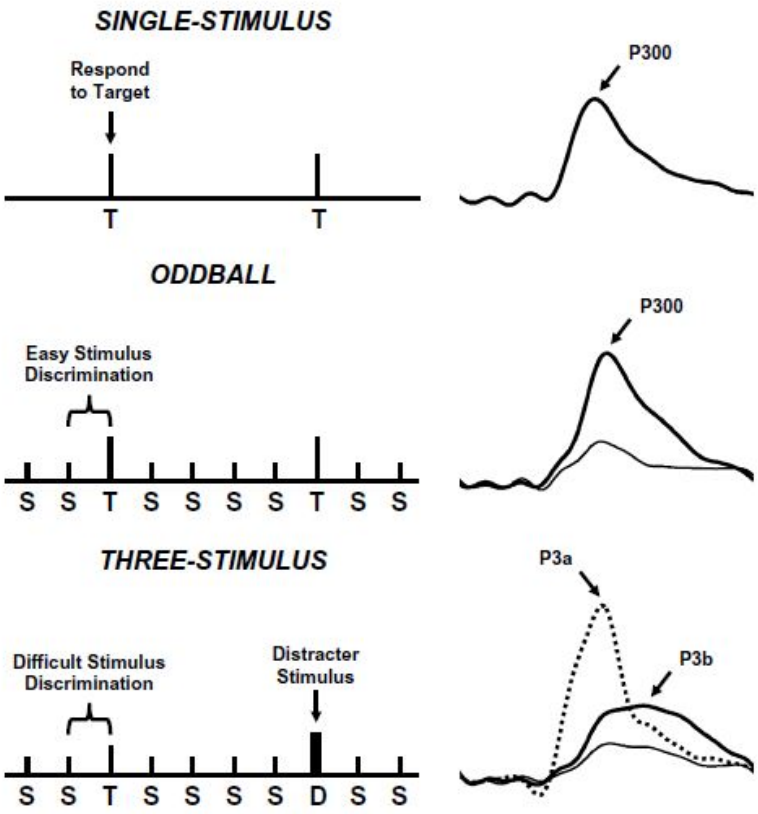

Figure 2: Three protocols which can be used to elicit a P300 response [15]

good choice for a P300 speller implementation. In the more complicated 'Three-Stimulus' protocol, a 'distractor' stimulus is added to the sequence. The distractor corresponds to the P3a curve in Figure 2, and the target elicits the P3b trace [15]. The exact neurophysiological processes underlying the P300 response are unknown, but a number of factors are known to affect it. The remainder of this overview will focus on two items: biological or task related factors which affect the P300, and current theories on its neurophysiological origin.

The P300 is a small electrical potential which ranges in amplitude from 6 to 20 μV. It has a latency (i.e., time from stimulus onset to peak amplitude) of 250 to 400 ms. The signal peak persists for less than 100 ms. The P300 is has a greater amplitude in signals recorded from parietal electrodes compared to frontal electrodes (i.e., magnitude increases from Fz (frontal center) to Cz to Pz (parietal center) [16]. The reason for this increase has

to do with the neurological origin of the P300.

Although the P300 has not yet been explained in its entirety, with the improvement of fMRI imaging technology, the physiological origins of the signal are better understood. Before discussing neurophysiology, however, it is important to understand that 'P300' refers to two distinct signals: the P3a and the P3b. Both signals have similar amplitude and latency. The P3a, however, is associated specifically with a 'distractor' or a 'novel' stimulus. Referring to Figure 2, the distractor is represented by the 'D' under the Three-Stimulus protocol. The P3b is elicited when a target 'T' stimulus is presented. The signals also have different topographic amplitude distributions. P3a signals have a central maximum, whereas P3b signals have a parietal maximum. P3a latency and response time are widely associated over all electrodes, but P3b signals are correlated specifically over parietal electrodes. Furthermore, P3a signals are thought to be associated with visual attention, while P3b signals are thought to be associated with memory when performing a task [15]. For the purposes of this review, we will focus on the P3b signal because it is elicited via the Oddball Paradigm, the engine behind the P300 speller.

The theory that P3b is associated with task memory makes sense in light of recent fMRI data. It is well known that the temporal lobe of the brain deals with memory storage. It is also well known that the parietal lobe contains the homunculus, the area of the brain responsible for motor action. A recent study shows that the P3b is observable after sequential temporal and parietal activation. It is therefore logical to think of the P3b (or 'P300,' as it will be referred to from this point forward) as a sort of 'pre-action signal.' The brain recognizes the stimulus as significant, and a signal (the P3b) is passed to the motor cortex in the parietal lobe in the case that the stimulus should require an action. [15]

Since it has been over forty years since the discovery of the P300, there have been many studies on various ways the amplitude and latency of the signal can be affected. There are two main categories of signal effectors: biological effectors (i.e., varying the exercise state of the subject) and task-related effectors (i.e., changing the probability of target 'T' occurrence). For convenience, these results are listed below in Table 1.

Table 1. Summary of P300 effectors (biological and task related).

Experimental Variable	Effect on P300 Amplitude	Effect on P300 Latency
Biological effectors		
Eating food	Food intake increases the amplitude of the P300	No effect on latency [16]
Body temperature	No effect	Shorter latency with increased temperature [16]
Heart rate	No effect	Shorter latency with increased heart rate [16]
Seasonal variation	In seasons with more light, P300 amplitude is increased (females had higher amplitudes than males)	No effect [16]
Tonic exercise	With exercise, amplitude increases	Latency decreases [16]
Alcohol intake	Decreased amplitude	Increased latency [16]
Task-related effectors		
Memory and perceptual load	Decrease in amplitude	Not applicable [17]
Rarity of target	Same amplitude for 15% and 45% stimulus probability	Not applicable [18]
Inter-stimulus interval (ISI)	Over 60% increase in amplitude from 1 sec to 2 sec ISI	Not applicable [18]

1.2.3 P300 Speller

A P300 speller exploits the P300 signal in order to determine a letter to type. It consists of a 6×6 array of alphanumeric characters, as shown in Figure 3. In Section 1.2.2, the method by which the P300 can be elicited, the 'Oddball Paradigm,' was discussed. To summarize, a series of non-target stimuli are presented to the user with a lower number of target stimuli. When the user notices a target stimulus, the P300 can be observed. [19, 6]

Figure 3: The 6×6 P300 speller matrix [7]

In a P300 speller implementation, the rows and columns of the speller matrix are 'block-randomized.' Each of the six rows and columns flash once per trial in a random order. To maintain attention, the user is instructed to count the number of times the target character was highlighted [20]. The cycle of twelve flashes repeats a number of times in order to obtain enough signals to eliminate noise by averaging. The amount of time that the row or column flashes is called the 'stimulus interval,' and the amount of time the matrix is blank in between flashes is called the 'inter-stimulus interval,' or ISI.

The stimulus interval and the ISI vary from study to study, but there is a publicly available P300 speller dataset which is typically used by researchers in the field for validation

of results. The dataset (Dataset IIb) comes from the 2003 BCI Competition and is available for download on-line [21]. A detailed description of this data set, such as the protocols used for stimulus interval and ISI, the hardware used to record signals, and more, can be found in Section 3.1.

1.2.4 EEG Hardware

Electroencephalogram (EEG) technology has been in use by medical and research professionals for over sixty years. In the most general sense, an EEG is a recording of electrical potentials generated by the brain. EEG can be invasive or non-invasive, as discussed in Section 1.2.1. In this document, only non-invasive surface EEG is discussed. It is impractical and unsafe for an off-the-shelf consumer grade headset, such as the EPOC from Emotiv, to require the user to penetrate their own skin.

There are four main parts to an EEG system which all affect the quality of the signal attained. The first is background noise. The environment in which signals are recorded is critical. Many EEG studies are performed in special settings where signal contaminants, such as wireless signals and even power line interference, have been minimized. The second is the electrodes themselves. Electrodes can be made out of a variety of materials in differing shapes which may either improve or degrade signal quality. They are placed according to the International 10-20 System, which is addressed in the following section. In addition to the electrode itself, the wires which carry the signals from the electrodes to the amplifier are also important. The third critical component is the type of signal amplifier used. EEG signals have small amplitudes, and an amplifier capable of high gains without distorting the signal is preferred. Finally, the fourth main part to the EEG recording system is the analog to digital converter which discretizes the continuous waveform. The sampling rate and the resolution of the analog to digital converter are critical. [22]

The most common shape of surface electrodes is a cup or a disk. The diameter of round electrodes is in the range of 4-10 mm. If electrodes are smaller or larger than this, they are not capable of maintaining good mechanical and electrical contact with the scalp [24]. Most

often, electrodes are coated with a very conductive metal, such as gold, silver chloride, tin or platinum. Silver has the highest conductivity of all common electrode metals, and it is generally regarded as the best choice for EEG. A medical glue is used to affix the metal cup electrodes to the scalp. After the glue dries, the cup electrodes are filled with a conductive jelly. Saline electrodes are also capable of conducting electrical signals, but they have poor conductivity compared to metal electrodes. The benefits of using saline electrodes are setup time and cost. Glue and conductive jelly take time to carefully apply to each electrode, they can be expensive, and the electrodes are difficult to remove once they have been installed. The wire used to carry the signals from the electrodes to the amplifier is always insulated, but it can also be shielded for added protection against noise. Shielded wires are more expensive, but they are able to better maintain the quality of the signal from the electrode site to the amplifier [23]. [24]

The amplifier should be able to provide a gain of 100-100,000 without distorting the signal. The highest gain is not always the best, as there are other factors which contribute to good digital signal resolution. The input impedance of the amplifier should be at least 100 MΩ to mitigate noise. It is recommended to use an analog to digital converter which is capable of resolving at least 0.5 μV amplitude. To do this typically requires twelve bits of information. Once the signals have been digitized and input to a computer, they can be processed by filters to isolate desired frequency bands. Artifact removal (e.g., EKG, EoG, or EMG signal contaminants) can also be performed with computer algorithms. [22]

1.2.5 Emotiv EEG Headset

The EPOC is a 14-channel wireless EEG headset. The electrodes are positioned at AF_3, F_7, F_3, FC_5, T_7, P_7, O_1, O_2, P_8, T_8, FC_6, F_4, F_8, and AF_4 according to the International 10-20 Location System, which is a standard way of denoting electrode positions on the scalp (see Figure 4). There is also a reference electrode below the ear on either side of the head. The data are transmitted to a compatible computer or mobile device via a 2.4 GHz wireless USB dongle. The device samples sequentially at 128 Hz with 0.51 μV resolution. There is a

single Analog to Digital converter which is used to convert the continuous EEG readings to discrete data. The EPOC has a digital 5th order Sinc filter which restricts the bandwidth of recorded data between 0.2 and 45 Hz. The device also has two notch filters for power line noise at 50 and 60 Hz. [4]

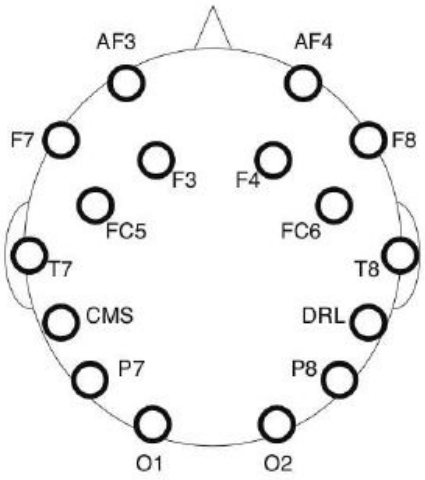

Figure 4: EPOC electrode locations [1]

Although the Emotiv EPOC is inferior to medical grade systems, a recent study has shown that it does capture the P300 signal. Work by Duvinage, *et al.* [1], compares the EPOC to the ANT, a 128 channel cap with Silver Chloride electrodes and a sampling rate of 2048 Hz. There were 7 participants in the study. A 2×2 matrix was used to elicit the P300 in the study, rather than a full 6×6 matrix. Note that the odds of randomly selecting the correct item from a 2×2 matrix are $\frac{1}{4}$ compared to $\frac{1}{36}$ for a 6×6 matrix. However, the signal was classified correctly much more than 25 percent of the time, indicating that the P300 was successfully isolated. Results of the study show that the ANT performed 8% better than the EPOC in correct detection on average, and both the ANT and the EPOC exhibited correct detection rates above 75%. Another research group has had success using the EPOC to detect another type of EEG signal, Steady-State Visually Evoked Potentials (SSVEPs) [25].

In academic circles, the EPOC is gaining popularity. Its attractiveness is not due to

superior signal quality. Instead, the headset represents a new wave of research. The EPOC and devices like it open up the potential for mobile BCI to become a reality. As a result of this new possibility, research groups are beginning to take notice [1, 2, 3]. If such devices are available at reasonable prices, perhaps lower quality signals can be acceptable. With the growing popularity of machine learning data processing techniques, perhaps many of these noise issues can be mitigated. The search is on for a computationally inexpensive classifier for use in mobile BCI. The remaining sections of this document will present a literature review of current P300 detection techniques and outline a proposal to do research in this growing field.

2 Research Objectives

2.1 Literature Review

The first section of this chapter, 'Research Objectives,' will give the reader a detailed overview of relevant P300 research. There is wide variability in this field, and no two approaches to the P300 problem are the same. Subsection 2.1.1 will cover the relative pros and cons of common EEG preprocessing techniques and 2.1.3 will discuss applying spatial filtering algorithms to EEG data. Finally, 2.1.4 will summarize some common machine learning techniques used to classify a given signal as a target or a non-target. Subsection 2.1.2 presents a valuable, less common, approach to the P300 detection problem.

2.1.1 EEG Preprocessing

Once raw brain signals have been digitized and sent to a computer, they must be pre-processed before they can be submitted to the classification algorithm(s). Pre-processing can refer to simple frequency filtering, downsampling, and time windowing of data, or it can refer to more complex spatial filtering algorithms (e.g., ICA). In this section, the former is discussed, while the latter spatial filtering algorithms are discussed in 2.1.3. There are two main questions which must be answered in regard to EEG pre-processing. The first is a question of frequency bandwidth. EEG recordings contain signals with many different frequencies. The target signal, in this case the P300, resides in a certain bandwith f_1 to f_2. From this, the question arises: what is the optimum f_1 and f_2 for detecting the P300? Similarly, if an EEG time window, called an 'epoch,' containing a P300 potential is defined by the interval t_1 to t_2, what are the optimal t_1 and t_2 for accurate detection of that signal? Each research lab has its own answer to these two questions, and the motivation behind their choices is almost never mentioned. Choice of the time and frequency windows is apparently arbitrary. However, there are a few studies which have critically investigated the pre-processing problem.

Manyakov and Chumerin (2010) addressed the questions of optimal frequency and time window. Five male and two female subjects were fitted with an eight channel wireless EEG

headset. The subjects were given the task of spelling given a 6×6 matrix of characters and numbers. Each row and column was highlighted for 100 ms with a 100 ms ISI. Beginning from stimulus onset, each epoch was 1000 ms in length. For every symbol spelled, each row and column of the matrix was flashed ten times. The signals were then averaged to reduce noise. The data were filtered with a zero-phase, fourth order digital Butterworth filter and down-sampled to $2(f_2) + 1$ samples, per the Nyquist-Shannon sampling theorem. The Feature Extraction algorithm used to classify the data was held constant for all trials. The frequency bandwidth of the filter and the time interval of the data sent to the classification algorithm were the only experimental variables. [26]

Since the experiment investigated two parameters, the time window and the frequency window, one was held constant to isolate the behavior of the other. In this study, the authors fixed the time window at 1000 ms in order to find the ideal frequency bandwidth. After finding the ideal f_1 and f_2, the data were filtered in that frequency range to observe the effect of time window on classification accuracy. The results show that the best frequency range and time window with which to pre-process raw data are 0.1-10 Hz and 200-350 ms, respectively. [26]

A more recent (2012) study with 23 participants and similar experimental parameters validates the frequency bandwidth findings of Manyakov and Chumerin. Using a linear Support Vector Machine (SVM) classfier, the ideal preprocessing frequency range was found to be 0.1-15 Hz. [27]

In addition to frequency bandwidth and time window considerations, size of data and noise must also be considered. A common way of dealing with noise is averaging each channel over multiple trials. This is not a spatial filtering technique since the channel is averaged with itself, not other channels (the latter method is called Grand Averaging and is discussed in 2.1.3). The accuracy of the speller increases as the number of averaged trials increases. [28]

Another technique to mitigate noise is called Windsorizing. It is well known that EEG contains artifacts from other biological signals. To deal with these artifacts, one can parti-

tion data from a given epoch into the lowest 10% values and the highest 10% values. Values lower than the tenth percentile and higher than the ninetieth are set to the values for tenth and ninetieth percentiles, respectively. Following Windsorizing, it is common also to scale all recordings (e.g., restrict data to the interval $[-1, 1]$). [29]

Below is a list detailing (in order of implementation) some common EEG pre-processing techniques specific to P300 ERP detection:

1. Record n channels of data and at least one reference channel
2. Filter data
3. Downsample data
4. Select time window
5. Windsorize data
6. Scale data

Preprocessing is one common component of a P300 study, the other being the classification algorithm(s), which are discussed in 2.1.4. The next section, however, addresses describes a novel approach to improve P300 detection accuracy and speed.

2.1.2 Electrode Selection

There are a wide variety of electrode configurations used in P300 research. Most studies follow a similar methodology: determine a set of possible electrode configurations, acquire the data, and evaluate the performance of the algorithm given a certain configuration [29]. Selection of the electrodes is usually based on some *a priori* knowledge of sensor significance [30]. However, from one subject to another, some sensors may contain more useful information than others. If the signal collection framework is fixed, variations in the individual subject cannot be taken into account. A recent study by Cecotti *et al.*, addressed this issue [31]. P300 data were acquired via the 6×6 speller matrix. Data were recorded from 32 electrodes and analyzed off-line. Given an *a posteriori* knowledge of the class of a given epoch (target or non-target), a backwards elimination protocol for sensor rejection was followed:

1. Use BLDA (Bayesian Linear Discriminant Analysis) to classify epochs as either 'target' and 'non-target' after spatial filtering using the xDAWN algorithm
2. Determine which two electrodes least affect the given parameter (i.e., classifier accuracy)
3. Eliminate the electrodes from Step 2
4. Repeat Steps 2 and 3 fifteen times until all electrodes have been eliminated
5. Assign a rank to each electrode based on when it was eliminated (electrodes eliminated in later rounds are more important and have higher rank)

In some subjects, certain electrodes have higher rank than in other subjects. However, even though there is wide variability between subjects, when spatial filtering is used and signal to signal plus noise ratio (SSNR) is minimized, there is a 'common factor' set of electrodes which are shared in half of the subjects. Those sensors are: P_z, P_8, O_z, P_3, P_7. The sensors are circled in red below in Figure 5.

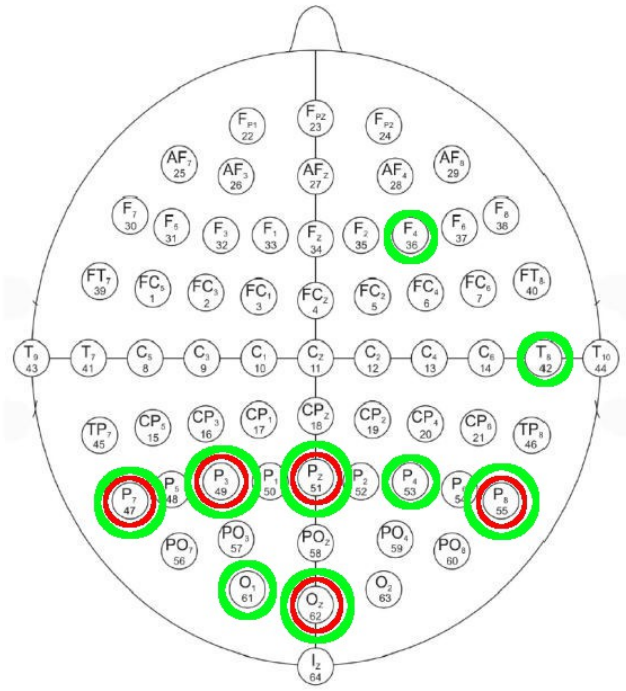

Figure 5: P_z, P_8, O_z, P_3, and P_7 electrodes circled in red. The red circles denote 'important' electrodes common to subjects when spatial filtering is used and SSNR is minimized. The green circles denote 'important' electrodes common to subjects when spatial filtering is used and classification error is minimized.

In addition to this finding, in all cases when spatial filtering was used in conjunction with the BLDA classifier, correct detection rates significantly improved. Some common spatial filtering techniques are discussed in 2.1.3. The most important electrodes in terms of both speller and P300 classification accuracy are circled in green in Figure 5. In summary, the red circles in Figure 5 represent top electrodes (top 8 highest ranks) common to half of the subjects in the study when spatial filtering is used and SSNR is minimized. The green circles represent sensors with the top 8 highest ranks and occurances across subjects when spatial filtering is used and classification error is minimized.

2.1.3 Spatial Filtering Algorithms

The simplest method of filtering EEG signals is to average all of the electrodes of a given epoch, point for point. Such an approach is referred to as a fixed spatial filter. The weights for each sensor can be uniform or tailored based on the problem (i.e., if *a priori* knowledge about a given electrode's significance exists, it can be weighted accordingly). Although such fixed spatial filters are computationally cheap, they have obvious limitations. For example, no two EEG patterns are the same, so a fixed-weight system would not be able to adapt to the variability between individuals. Adaptive spatial filters (e.g., Independent Component Analysis (ICA), Principle Component Analysis (PCA), and Common Spatial Patterns (CSP)) offer a solution to this problem. ICA, for examle, assumes no *a priori* knowledge about electrode significance. Instead, the algorithm minimizes the amount of mutual information between the electrodes, thus determining the 'independent' components [32]. Another way of explaining ICA is the classic 'cocktail party' analogy. In this analogy, brain signals can be thought of as 'voices' in a crowded room. Electrodes can be thought of as 'microphones' recording the voices. As long as the number of microphones is equal to or greater than the number of voices in the room, it is possible to decipher what an individual voice is saying using ICA. [31]

There are three assumptions which must be made to use ICA: the sources (e.g., electrodes) are independent, their mixing is linear, and the mixing model is time-invariant [33].

In ICA, we assume that n independent signals, $s(t) = s_1(t), ..., s_n(t)$, exist. These signals can be thought of as the 'pure' EEG (i.e., without noise). The recording hardware is not capable of detecting $s(t)$. Instead, it records p linear mixtures of the sources, $x_1(t), ..., x_p(t)$, for times $t = t_0, ..., t_{final}$. So each $x_p(t)$ is a linear combination of the source signals $s_n(t)$. The following Equation 1 represents this mixing of sources in matrix form.

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) \; \forall \; t \tag{1}$$

In Equation 1, $\mathbf{s}(t)$ is the column vector of source signals, $\mathbf{x}(t)$ is the column vector of observed signals, and $\mathbf{A}$ is the mixing matrix. The task of ICA is to find a 'de-mixing' matrix, commonly referred to as a separating matrix, $\mathbf{W}$, such that observed signals can be translated into the original source signals. The matrix $\mathbf{W}$ can be multiplied by the observed signal vector $\mathbf{x}(t)$ to yield the independent components, $\mathbf{u}(t)$, as in Equation 2. [32]

$$\mathbf{u}(t) = \mathbf{W}\mathbf{x}(t) \; \forall \; t \tag{2}$$

There are several ways of determining the matrix, $\mathbf{W}$. One common approach is to minimize the mutual information, $\mathbf{I}$, between components. Another method is to maximize the non-Gaussian attributes of the sources. The following equation defines mutual information for a random vector $\mathbf{X}$, where $p(x)$ is the probability density function (pdf) of x:

$$\mathbf{I}(\mathbf{X}) = \int p(x) \log \frac{p(x)}{\prod_{i=1}^{n} p_i(x)} dx \tag{3}$$

When the sources are independent, $p(s) = \prod_{i=1}^{n} p_i(s_i(t))$, then Equation 3 is equal to zero because the log of one is zero. The goal of many ICA algorithms is to obtain components $\mathbf{u}_i(t)$ which are independent (i.e., share little mutual information). In other words, the goal of ICA is to minimize the value of Equation 3. Obtaining independent components can also be thought of in terms of maximizing entropy, $\mathbf{H}$, between components. The information between two signals $\mathbf{X}$ and $\mathbf{Y}$ can be expressed in terms of the entropy of the signals, where

entropy of $\mathbf{X}$ is defined as $\mathbf{H}(\mathbf{X}) = \mathbf{H}(p(x)) = -\int p(x) \log p(x) dx$. The following equation relates entropy to mutual information:

$$\mathbf{I}(\mathbf{X}, \mathbf{Y}) = \mathbf{H}(\mathbf{X}) - \mathbf{H}(\mathbf{X}|\mathbf{Y}) \tag{4}$$

It can be seen from Equation 4 that maximizing $\mathbf{H}(\mathbf{X}|\mathbf{Y})$ (which measures the uncertainty in $\mathbf{X}$ given a knowledge of $\mathbf{Y}$) is equivalent to minimizing $\mathbf{I}(\mathbf{X}, \mathbf{Y})$. If the variables follow a Gaussian distribution, however, we cannot separate them with ICA. Therefore, most ICA algorithms attempt to maximize non-Gaussianity. [32] This is the basic mathematical framework for ICA. There are variations in how the matrix $\mathbf{W}$ is calculated, as well as tweaks to speed up computation (such as assuming an *a priori* knowledge of P300 shape [34]), but the basic concepts stay the same: minimize mutual information and maximize non-Gaussianity between components.

Principle Component Analysis, or PCA, is closely related to ICA, but there are a few key differences. PCA algorithms still attempt to find a set of components which explain most of the data, which is why PCA (like ICA) is often used to reduce the dimensionality of a problem. PCA algorithms find an orthogonal frame of reference directions whose axes are determined by second order statistics. In ICA, the frame of reference is not necessarily orthogonal, so ICA is not as limited. However, the higher order statistics used by ICA require more computational power than PCA, which at its most basic level solves for the eigenvalues and eigenvectors of the covariance matrix. Another difference is that in ICA, the number of independent sources obtained depends on the number of sources chosen to compute, which is not the case in PCA. [32]

One way of implementing PCA is called Singular Value Decomposition, or SVD. Vectors containing the P300 signal are used as inputs to the SVD algorithm. SVD decomposes the matrix containing P300 vectors into two orthogonal matrices, $\mathbf{U}$ and $\mathbf{V}$, and a diagonal matrix, Σ. If $\mathbf{X} = [x_1, ..., x_n]$, where x_n are vectors containing the P300, the following equation represents the SVD of $\mathbf{X}$:

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T \tag{5}$$

The columns in $\mathbf{U}$ and $\mathbf{V}$ from Equation 5 are the left and right singular vectors. The diagonal values in Σ are the singular values. Once the decomposition of the 'training set' $\mathbf{X}$ into these three matrices has been made, it is now possible to form a feature matrix for targets, $\mathbf{F}_X$. This can be accomplished by taking the vectors in $\mathbf{U}$ corresponding to the l largest singular values in Σ and combining them into a matrix $\hat{\mathbf{U}} = [u_1, ..., u_l]$. The feature matrix $\mathbf{F}_X$ is formed via the following equation:

$$\mathbf{F}_X = \hat{\mathbf{U}}^T\mathbf{X} \tag{6}$$

Similar to Equation 6, multiplying the non-targets $\mathbf{Y}$ by the matrix $\hat{\mathbf{U}}^T$ forms the feature space $\mathbf{F}_Y$. Once the means of the feature matrices have been calculated, classification of a test epoch can be carried out with an algorithm such as LDA (Linear Discriminant Analysis). [35]

The final spatial filtering algorithm to be discussed in this section is known as xDAWN. xDAWN is a recent spatial filtering algorithm based loosely on SVD and PCA, but with added layers of intricacy. The algorithm attempts to define an estimation of the subspace which contains P300 signals. Additionally, the algorithm attempts to maximize the Signal to Signal plus Noise Ratio (SSNR) to enhance the quality of the estimated signals. [36] The effect of applying the xDAWN algorithm to P300 containing EEG data on Classification Accuracy is reflected below in Figure 6. xDAWN performs the best out of all algorithms, although it is worth noting that no spatial filtering algorithm (Reference data series) outperformed PCA. PCA is limited in the sense that the calculated components only explain data with orthogonal axes, which could explain its poor performance.

In Equation 7, the matrix $\mathbf{X}$ represents recorded EEG signals, the matrix $\mathbf{N}$ is the noise term, the matrix $\mathbf{A}$ is the matrix of P300 signals, and the matrix $\mathbf{D}$ is the Toeplitz matrix with $\mathbf{D}_{\tau_k,1} = 1$ where τ_k is the stimulus onset of the kth target stimulus. All other elements

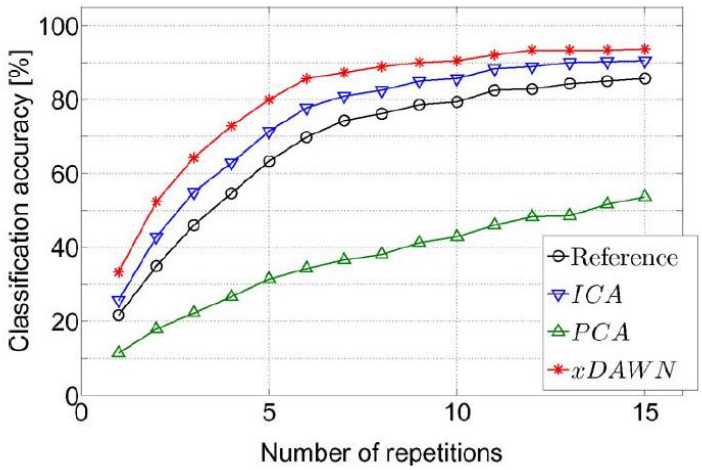

Figure 6: A BLDA classifier was used for all spatial filtered data. Here xDAWN is compared to a Reference set of data (no spatial filtering), PCA filtered data, and ICA filtered data. As the number of averaged epochs increases, the classification accuracy increases. [36]

in $\mathbf{D}$ are zero. [36]

$$\mathbf{X} = \mathbf{DA} + \mathbf{N} \tag{7}$$

From Equation 7, the solution to the least-squares estimation of $\mathbf{A}$ is: $\hat{\mathbf{A}} = (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{X}$. If we then take the SVD of the matrix $\hat{\mathbf{A}}$, we can obtain the largest singular values of the P300 matrix, which can be used to construct a subspace onto which new signals can be projected. The SVD of $\hat{\mathbf{A}}$ is as follows:

$$\hat{\mathbf{A}} = \Sigma\Delta\Pi^T \tag{8}$$

From Equation 8, Σ, Δ, and Π can be separated into their signal and noise subspaces: Σ_s and Σ_n, Δ_s and Δ_n, and Π_s and Π_n. With this knowledge, $\hat{\mathbf{A}}$ can be re-written as:

$$\hat{\mathbf{A}} = \Sigma_s\Delta_s\Pi_s^T \ + \ \Sigma_n\Delta_n\Pi_n^T \tag{9}$$

The xDAWN model then becomes:

$$\mathbf{X} = \mathbf{D}\mathbf{A}'_{PCA}\mathbf{W}^T_{PCA} + \mathbf{N}' \tag{10}$$

Where $\mathbf{A}'_{PCA} = \Sigma_s\Delta_s$, $\mathbf{W}_{PCA} = \Pi_s$, and $\mathbf{N}' = \mathbf{N} + \mathbf{D}\Sigma_n\Delta_n\Pi_n^T$. The problem with a model like this is that the noise was not taken into account when the spatial filters were being estimated. To mitigate this problem and improve the model, Rivet *et al.*, chose to incorporate maximization of the SSNR into their algorithm.

The matrix $\mathbf{U}$ is a spatial filter matrix with row number equal to the number of samples and column number equal to the number of features. To maximize the SSNR, the following spatial filters $\hat{\mathbf{U}}$ are constructed:

$$\hat{\mathbf{U}} = \arg\max \frac{\mathrm{Tr}(\mathbf{U}^T\hat{\mathbf{A}}^T\mathbf{D}^T\mathbf{D}\hat{\mathbf{A}}\mathbf{U})}{\mathrm{Tr}(\mathbf{U}^T\mathbf{X}^T\mathbf{X}\mathbf{U})} \tag{11}$$

By substituting a QR factorization of $\mathbf{X}$ and $\mathbf{D}$ into Equation 11 and solving for the spatial filters $\hat{\mathbf{U}}$, it becomes possible to write the equation for the computation of the enhanced signals, denoted by $\hat{\mathbf{S}}$: $\hat{\mathbf{S}} = \mathbf{X}\hat{\mathbf{U}}$. [36]

A summary of the algorithms covered can be found in Table 2 (see Section 2.1.5).

2.1.4 Classification Algorithms

In this section, we will review classification algorithms which are frequently used in P300 detection applications. Before beginning, it is important to consider the topology of the data. The input data for a given classification algorithm may differ due to the type of preprocessing or spatial filtering used in an earlier step. For example, if the data were segmented into 1000 ms epochs and all electrodes were spatial filtered with ICA, the first independent component is only one vector of data. Recall from 2.1.3 that PCA and ICA are effective ways of reducing the dimensionality of a problem (i.e., expressing the main characteristics of data from 14 electrodes in one feature vector). By contrast, suppose no dimensionality reduction method has been employed. Instead of one feature vector, we have n vectors, where n is the number of electrodes. In this case, the input to the classifier is

actually a feature vector of n concatenated vectors.

To begin the discussion, suppose this latter case of n vectors concatenated in a long vector is the input. In this case, it is likely that the data from each electrode have been averaged with that same channel for a number of epochs to improve the signal to noise ratio (SNR). For example, there are n recordings of channel P_z for row three of the P300 speller matrix. Instead of grand averaging P_z with other channels, it is only averaged with itself over many trials. Each channel in the feature vector is an averaged channel, and the size of the vector is $Time_s$ multiplied by the total channel number, $Channel_n$.

One of the simplest and most commonly used ways of determining correlations between data series is Pearson's Correlation Coefficient [30]. Once a target (P300 containing) epoch has been obtained for each channel, it is averaged over a number of trials from the same channel which also contain the P300. Next, the correlation of each channel to its respective target signal can be evaluated for subsequent epochs. The more correlation a given channel has with its target data, the higher the weight for that channel. Equation 12 describes how to calculate this weight, r, for each electrode [37]. Since r varies between -1 and 1, the absolute value of r determines the weight [30]. It is easy to see how the summation of all the calculated r values for an epoch in question are higher if that epoch contains the P300 and lower if it does not.

$$r = \frac{\Sigma(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\Sigma(x_i - \bar{x})^2 \cdot \Sigma(y_i - \bar{y})^2}} \tag{12}$$

Where x is the feature in question and y is the target feature obtained from training. The values $\bar{x}$ and $\bar{y}$ represent the mean of the epoch in question for a given channel and the mean of the target epoch for that same channel, respectively.

Once the correlation coefficients r have been computed for each channel of the epoch in question, each sample from each channel in that epoch is multiplied by the weight w. The predicted row or column is the one which satisfies the following equation: Predicted row or column $= \arg\max[\sum w_i \cdot x_n]$. Where x_n are all samples from the channel in question multiplied by their corresponding weight w_i. The total sum of all samples from each channel

multiplied by their respective weights represents the score for a given row or column. The highest score for a row and column out of all the rows and columns represents the target letter.

This method, known as PCM (Pearson's Correlation Method), is computationally efficient, but it has been shown that it is inferior (in terms of classification accuracy) to other methods which are comparable in terms of their computational complexity [30]. One such method is Fisher's Linear Discriminant Analysis (FLDA). Linear classification models (e.g., FLDA, BLDA, Linear SVM) are an effective means of reducing dimensionality of a problem. For example, if we have eight input electrodes concatenated into a single long vector (one sample), linear methods such as FLDA project the n-D input features onto a single dimension using the following transformation:

$$y = \mathbf{w}^T \mathbf{x} \tag{13}$$

Where $\mathbf{x}$ is the matrix of input features and $\mathbf{w}$ are the calculated weights for each feature n. The weights $\mathbf{w}$ are set such that there is optimal separation of the two classes (target and non-target) on the projected dimension. Fisher's Linear Discriminant Analysis accomplishes this separation based on the means and the variances of the two data sets. Figure 7 illustrates this concept.

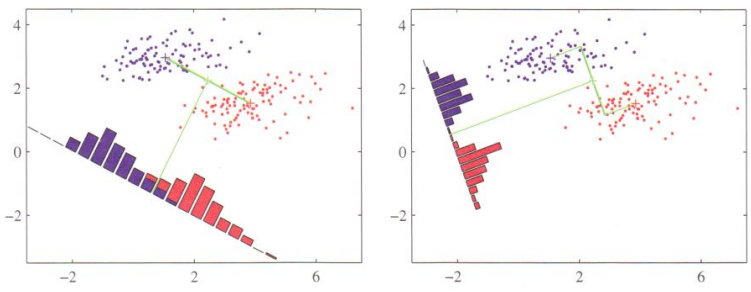

Figure 7: In the left plot, the blue and red classes are projected on a single dimension determined only by their means. In the right plot, the classes are separated with FLDA, which takes into account the mean and variance of each class to make the linear projection. [38]

As can be seen in Figure 7, less overlap of classes occurs when both the mean and the

variance of the two classes have been considered. This extra resolution makes FLDA a popular choice for P300 classification [30, 29, 35]. Weights are calculated for FLDA via the following Equation 14:

$$\mathbf{w} \propto \mathbf{S}_W^{-1}(\mathbf{m}_2 - \mathbf{m}_1) \tag{14}$$

Where $\mathbf{S}_W$ is the total within class covariance [38]:

$$\mathbf{S}_W = \sum_{n \in C_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^T + \sum_{n \in C_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^T \tag{15}$$

With $\mathbf{m}_n$ representing the respective class mean.

Note that in the above Equation 14, we need to calculate the matrix inverse, $\mathbf{S}_W^{-1}$. For high-dimensional problems, such as P300 detection, it is possible for the number of features to be larger than the number of training samples. In this case, $\mathbf{S}_W$ may not be invertible. To solve this problem, we can replace $\mathbf{S}_W^{-1}$ in Equation 14 with the Moore-Penrose pseudo-inverse, $\mathbf{S}_W^{\dagger}$. [29]

Once the data from target and non-target vectors have been projected down from multiple dimensions n into the single dimension y per Equation 13, a discriminant can be constructed to classify data. For example, once an n-feature vector $\mathbf{x}$ has been projected down into y, if the value of $y(\mathbf{x})$ is greater than some threshold y_0, then the sample belongs to class C_1. However, if the value of $y(\mathbf{x})$ is less than y_0, then the sample belongs to C_2. [38]

In P300 based BCI, however, the classification is usually made based on a maximum argument rather than comparison with a threshold y_0. For example, the output y is calculated as $y = \mathbf{w}^T\mathbf{x}$ for an input vector $\mathbf{x}$. The vector with the highest value ($\arg\max[\sum \mathbf{w}^T\mathbf{x}]$) out of all the rows or columns is the selected as the target row or column. The intersection of the target row and column represents a single letter or number from the speller matrix. [29]

Using FLDA for classification is a common way of determining target responses in P300 BCI speller applications, but there is another method which is preferred. The algorithm is

called Bayesian Linear Discriminant Analysis (BLDA). It is more complex mathematically, which should be taken into consideration for cost-benefit analysis, but studies have also shown that it is more accurate than FLDA (although it is only slightly more accurate [29]).

Bayesian Linear Discriminant Analysis improves on FLDA because it is built to avoid overfitting to noisy and high dimensional datasets. It has been used with great success in several P300 studies. In [29, 28], BLDA is directly compared to FLDA. It was shown in both cases that BLDA results in higher classification accuracy and requires fewer trials to reach acceptable correctness. The following method proposed by Hoffmann *et al.* [29] assumes that the targets $\mathbf{t}$ (where $t \in [-1, 1]$) and the feature vectors $\mathbf{x}$ are linearly related with some added noise (Equation 16).

$$\mathbf{t} = \mathbf{w}^T\mathbf{x} + n \tag{16}$$

Given the above assumption, it is possible to write the likelihood function for the weights $\mathbf{w}$:

$$p(\mathbf{D}|\beta, \mathbf{w}) = \left(\frac{\beta}{2\pi}\right)^{\frac{N}{2}} \exp\left(-\frac{\beta}{2}||\mathbf{X}^T\mathbf{w} - \mathbf{t}||^2\right) \tag{17}$$

Where $\mathbf{D}$ is the pair of $\mathbf{X}$ (the input vectors) and $\mathbf{t}$ (the target labels), β is the inverse variance of the noise, and N is the number of examples in the training set. Once the *a priori* distribution for $\mathbf{w}$, $p(\mathbf{w}|\alpha)$ has been written (see [29]), it becomes possible to express the posterior distribution using Bayes' rule:

$$p(\mathbf{w}|\beta, \alpha, \mathbf{D}) = \frac{p(\mathbf{D}|\beta, \mathbf{w})p(\mathbf{w}|\alpha)}{\int p(\mathbf{D}|\beta, \mathbf{w})p(\mathbf{w}|\alpha)d\mathbf{w}} \tag{18}$$

Because the expression for the prior and likelihood are both Gaussian, the posterior distribution is also Gaussian with mean $\mathbf{m}$ and covariance $\mathbf{C}$. Multiplying the likelihood function, Equation 17, with a new input vector $\hat{x}$ and the posterior distribution, Equation 18, it is possible to obtain the expression for the probability of the targets, $\hat{t}$:

$$p(\hat{t}|\beta, \alpha, \hat{x}, \mathbf{D}) = \int p(\hat{t}|\beta, \hat{x}, \mathbf{w})p(\mathbf{w}|\beta, \alpha, \mathbf{D})\ d\mathbf{w} \tag{19}$$

Equation 19 is Gaussian with mean μ and covariance σ. In the literature, the target rows and columns are selected by $\arg\max \sum_{i=1}^{T} \mu$, where T is the total number of trials for a given row or column. The expression for μ is:

$$\mu = \mathbf{m}^T \hat{x} \tag{20}$$

Where $\hat{x}$ is the input vector and $\mathbf{m} = \beta(\beta\mathbf{X}\mathbf{X}^T + \mathbf{I}'(\alpha))^{-1}\mathbf{X}\mathbf{t}$. A MATLAB implementation of the BLDA algorithm described here and featured in Hoffmann *et al.* [29] can be found at the following web address, courtesy of the EPFL BCI group: `http://bci.epfl.ch/p300`.

The performance gap between BLDA and FLDA is minimal [28]. The difference in the aforementioned methods is in their approach to the classification problem: the former classifies based on an optimal projection determined from class means and variances and the latter determines the class in a similar way but with the added assumption of Gaussian noise. The next (and last) classification algorithm this review will cover differs little in accuracy when compared to BLDA and FLDA [28, 30]. However, Support Vector Machines (SVMs) are becoming increasingly popular in P300 research and they have been used effectively by several groups [39, 30, 28].

The basic intuition behind SVMs is to find a hyperplane in n-dimensional space which separates two classes with the maximum possible margin. This concept of margin maximization is illustrated for two dimensions below, in Figure 8.

The points which lie on either side of the margin are called support vectors. A vector $\mathbf{w}$ is perpendicular from a given support vector to the optimal hyperplane. The distance ρ across the boundary is defined as: $\rho = \frac{2}{||\mathbf{w}||}$, where the denominator is the norm of $\mathbf{w}$. Therefore, we define maximization of the margin ρ as being equivalent to the minimization of $||\mathbf{w}||$. [38]

The optimal hyperplane for traditional SVM is easily found when the data are linearly separable (as in Figure 8). However, in EEG, data are noisy and it is possible that a target will lie on the non target region on occasion. A regularization term C which accounts for this noise can be included to mitigate such problems. Typically, the hyperparameter C is

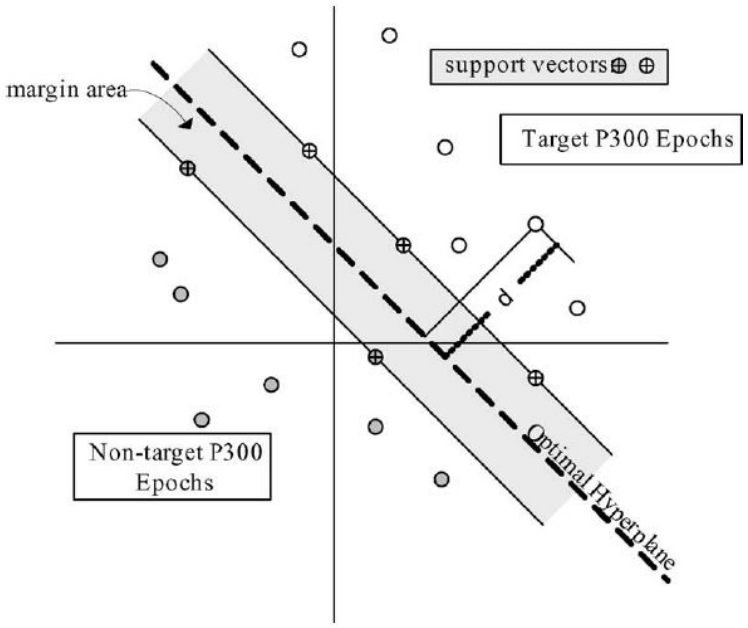

Figure 8: The plot depicts two classes of data, gray dots and white dots. The dotted line represents the optimal hyperplane separating the two classes such that the gray shaded margin is maximized. This is the basic idea of SVMs. [40]

determined by performing cross-validation and selecting the best solution. There are many efficient algorithms which can be used to solve the minimization problem. [39]

To select the target row or column, a calculation of distance from the sample to the separating hyperplane is performed (referred to as the SVM score). The row or column with distance close to the target data distances (based on training data) is selected as the target [28]. The formula for calculating SVM score is found below:

$$d = \sum_{i=1}^{l} \alpha_i y_i K(x_i, x) \tag{21}$$

Where d is the distance, or SVM score of a feature vector, i represents each point in the feature vector up to length l, α is the Lagrangian multiplier vector constrained to $0 \leq \alpha_i \leq C$, y is the class label, and K represents the kernel function used to account for the data not being linearly separable. Note that using a non-linear kernel function transforms the classifier from being simply a linear discriminant to a highly non-linear function. In

SVMs, however, the non-linear approximation comes at very little added computational cost. In P300 literature, when the non-linear SVM is used, the Gaussian kernel is popular [30]. This type of classifier would be referred to as GSVM, where the 'G' stands for the Gaussian kernel: $K(u, v) = \exp\left[\frac{|u-v|^2}{2\sigma^2}\right]$.

2.1.5 Summary

In Section 1, the Motivation and Background for this proposal were covered. The Background section, 1.2, covered basic concepts relating to BCI and neuroelectric potentials. The topics covered were: Brain-Computer Interfaces, the P300 Event Related Potential, the P300 speller matrix, common EEG recording hardware, and the hardware to be used in this research, the EPOC headset from Emotiv systems.

BCIs are useful on two main levels. The first is the most obvious: the medical level. A computer which is capable of ascertaining the thoughts and intentions of a paralyzed or dismembered patient has great rehabilitative potential. The ultimate goal of BCI on a medical level is to equal or surpass the human biological standard for a healthy individual in a disabled patient. On an academic level, BCI research has provided many insights into the complex workings of the human mind. For a general outline of a BCI, see Figure 1.

The P300 ERP can be elicited via three protocols (see Figure 2). The protocol of interest to this study is referred to as the 'Oddball Paradigm,' in which a rare stimulus presented occasionally in a batch of non-target stimuli initiates the ERP. The ERP itself is a small, approximately 6-20 μV, peak having a latency of 300 ms from the onset of the stimulus. The P300 is predominately located over the parietal region of the brain, and the signal is best resolved from those electrode positions (see Figure 5). Various biological and 'task-related' variables affect the amplitude and latency of the P300 potential (see Table 1).

The Oddball Paradigm can be exploited to spell words. Figure 3 depicts the 6×6 speller matrix used in P300 BCI studies. The rows and columns of the matrix flash. The user focuses on their target letter in the matrix. When the row and column containing their target letter is highlighted, a P300 ERP is visible for that corresponding EEG epoch.

Dataset IIb of the 2003 BCI Competition contains P300 data elicited from a speller matrix.

EEG hardware can range from low cost consumer devices, such as the EPOC, to high grade medical equipment. Electrode positions are described by the International 10-20 system (see Figure 5). The material used to make the electrodes and the wires used to carry the signals from the collector (electrode) to the amplifier vary. Silver chloride electrodes are regarded as the highest quality. Shielded cable is ideal because it mitigates noise pollution of the EEG from external sources. The amplifier used should be high gain (100-100,000), low noise, and have a high input impedance for optimum signal quality. The analog to digital converter used should be able to resolve, at minimum, 0.5 μV, requiring 12 bits of information. Artifact removal (e.g., EKG, EoG, or EMG contaminant signals) can either be performed via hardware or software.

The EPOC headset from Emotiv Systems has 14 saline electrodes with two reference, one behind each ear (see Figure 4). The device is powered by a rechargeable battery. It transmits EEG data wirelessly with a 2.4 GHz signal to a custom USB dongle. The analog to digital converter on the device is capable of resolving 0.51 μV. The EPOC has been used in prior academic studies to detect P300 [1, 2, 3].

In the Literature Review section, 2.1, four main topics were covered in detail: EEG preprocessing techniques, electrode positioning, spatial filtering methods, and commonly used classification algorithms. By no means is the literature review intended to be an in depth summary of over 40 years of P300 research. Instead, its aim is to cover the current state of the art and the most popular methods in recent academic literature.

Before EEG data are input to a classifier, oftentimes several 'cleaning' operations must be performed to enhance the classification accuracy. Individual pre-processing protocols differ from one paper to the next. In general, however, all procedures follow this basic outline: record n channels of data and at least one reference, filter the data (this requires selecting the optimal frequency range for which to filter the data), downsample the data to eliminate unnecessary features, select an epoch window (i.e., determine which segments of the recording are input to the classifier), and finally windsorize and scale the data so

the maximum and minimum amplitude for each channel are set to a fixed value. After this step, channels can be averaged with each other (a simple spatial filtering technique known as Grand Averaging), or multiple trials can be recorded for each target letter. As the number of trials increases, the average of each channel becomes less noisy leading to better classification accuracy.

Another factor which affects classification accuracy is electrode position. Several studies have attempted to answer the question, 'which electrodes influence classification accuracy the most in a P300 based BCI?' In Section 1.2.2 it was determined that the parietal electrodes give the clearest picture of the P300 potential. Figure 5 depicts the International 10-20 electrode array with the most important electrodes (i.e., those electrodes which have the greatest influence on the classification accuracy) circled in red. Other electrodes are circled in green to distinguish them from the more salient red circled electrodes.

Experimental variables such as preprocessing steps and channel selection differ slightly in literature, but there is some general consensus on what is acceptable. This is not the case for the last two topics covered: spatial filtering and classification algorithms. Due to the noisy and patient-specific nature of EEG, depending on the data set used, the best spatial filtering or classification algorithm varies. A summary of the most common techniques can be found in the Table 2.

Table 2. Summary of Spatial Filtering (SF) and Classification Algorithms (C).

Algorithm Name	Description
Grand Averaging (SF)	The simplest algorithm. Takes data from sensors and averages all of them over an epoch. A variant of this scheme is to incorporate *a priori* knowledge by assigning higher weights to the more relevant electrodes prior to averaging.
ICA (Independent Component Analysis) (SF)	Computationally expensive spatial filtering algorithm. Requires no *a priori* knowledge of sensor significance (although such knowledge can lead to enhanced detection when employed [34]). The algorithm attempts to minimize information shared between electrodes and maximize the non-Gaussianity of data to separate the source 'voices' from each other.
PCA (Principle Component Analysis) (SF)	PCA is actually just a special case of ICA, with a few differences. It is faster in comparison since it only uses second-order statistics, but it cannot explain data as well as ICA. The frame of reference calculated by PCA is necessarily orthogonal, which is not the case in ICA. Additionally, in ICA, the number of components computed depends on the number of sources chosen to compute. PCA is performed via SVD.
SVD (Singular Value Decomposition) (SF)	SVD is a linear algebra operation which decomposes a matrix (e.g., a matrix of P300 epochs) into two orthogonal matrices and a diagonal matrix. The highest singular values in the diagonal matrix correspond to columns in the left orthogonal matrix. These vectors can be concatenated to define a feature space.
xDAWN (SF)	The xDAWN algorithm is perhaps the most effective spatial filtering technique to date. It calculates a subspace which contains P300 potentials while taking into account and minimizing the noise of the recordings (maximizing the SSNR). This is accomplished by solving a least-squares problem, performing an SVD, and two QR factorizations.
PCM (Pearson's Correlation Method) (C)	PCM calculates weights w for each feature (channel) by determining its correlation with the target (P300 containing) signal (Equation 12). Targets are selected by choosing the maximum value of the sum of each electrode's samples multiplied by their respective weight. The chosen row or column is the one with the highest sum. This is the simplest method computationally, but it is also not a good performer.
FLDA (Fisher's Linear Discriminant Analysis) (C)	Projects data from multiple dimensions into a single dimension y and maximizes class separation based on the means and within class variances of the two possible classes. After training the algorithm with multiple target and non-target epochs, a value y_0 is selected which can be used to classify future samples (e.g., if $y(\mathbf{x}) \geq y_0$, then $\mathbf{x} \in C_1$). In P300 BCI, maximum argument is used instead of discriminant: $\arg\max[\sum \mathbf{w}^T\mathbf{x}]$.

Table 2. Summary of Spatial Filtering (SF) and Classification Algorithms (C) (cont'd).

Algorithm Name	Description
BLDA (Bayesian Linear Discriminant Analysis) (C)	The BLDA algorithm assumes a linear relationship between targets $\mathbf{t}$ and feature vectors $\mathbf{x}$ with an added Gaussian noise term. The idea is to obtain the distribution for the probability of the targets (Equation 19) and use the mean (μ) of that distribution to determine the target row or column. Instead of calculating a discriminant function, targets are computed via the following equation: $\arg\max \sum_{i=1}^{T} \mu$, where T is the total number of trials for a given row or column. BLDA has been shown to be more accurate and require fewer trials to reach an acceptable correctness level. [29, 28]
SVM (Support Vector Machines) (C)	The SVM algorithm is a powerful tool for binary classification because it can easily describe non-linear data with little added computational complexity (via the use of kernels). The basic idea of SVM is described in Figure 8. The goal is to maximize the gray shaded margin between the two classes of data. The separating hyperplane is by definition in the center of the margin. The minimization of the parameter $\|\mathbf{w}\|$ accomplishes this optimal separation. This minimization is accomplished by maximizing a dual problem that incorporates Lagrangian multipliers.

2.2 Goals for this Study

The body of work on P300 based BCI is enormous. The Literature Review section of this proposal serves as a general outline, but there are many approaches to P300 detection. This project, however, is unique because there have been a relatively small number of papers published on low cost consumer EEG headsets. The potential for such mobile devices in the field of BCI is enormous if software can accurately and efficiently detect the P300 potential. This project is a first step in that direction. The research question for this experiment is as follows: Given the inferior signal to noise ratio of EPOC from Emotiv Systems relative to clinical grade EEG systems, is it possible to develop a fast and efficient algorithm for P300 event related potential detection in the context of the 6×6 speller matrix? In other words, given the poor quality of the EPOC headset, how can letter prediction be sped up without sacrificing accuracy?

It is possible that the answer to this last question lies in the procedure for acquiring the EEG data, as discussed in Section 1.2.3. In every study reviewed, data is averaged over

multiple trials to mitigate noise. For example, in [28], classification accuracy improved from 5% to 25% for 0 and 5 averaged intensification cycles, respectively. In order to collect five trials of data for each row and column in a 6×6 matrix, depending on the ISI (Inter-Stimulus Interval) and stimulus length, this may take a long time. As an example, we choose 100 ms for the stimulus and 75 ms for the ISI (values used in collection of BCI Competition 2003 dataset IIb): $(6 \times 175ms) \times 2 = 2.1s$. Here we see that we have 2.1 seconds per round of intensification. Multiply that by five repetitions, that is over ten seconds per character (assuming the computation takes no time at all). To spell a seven letter word it would take 73.5 seconds (again, assuming zero time for computations).

The hypothesis of this project is that it is possible, after a limited number of trials (e.g., two), to predict which rows and columns are not targets and then leave them out them of future row and column intensifications. This elimination is justifiable because the rarity of the target required to elicit P300 can be as high as 45% [18] without affecting P300 amplitude or latency. If such a method were to work without sacrificing classifier accuracy, it would improve speller throughput. Using the previous example to illustrate, we have 4.2 seconds of full rounds (when two intensification cycles are averaged). After those two rounds, we eliminate the unlikely rows and columns (half of them). This leaves us with 1.05 seconds per trial to complete our remaining three data collections. The total time to spell a seven letter word with the proposed method is therefore $(4.2 + (3 \times 1.05)) \times 7 = 51.5s$, which is a 30% improvement in speed.

To summarize, the goals for this study are as follows:

1. Test various spatial filtering algorithms and classifiers on the BCI 2003 Competition Dataset and compare accuracy with time required for computation.
2. Collect P300 data from the Emotiv EPOC headset. Use eight parietal electrodes. As a control, also use a high grade amplifier to concurrently collect data from the same electrode locations. Doing this will provide a baseline with which the EPOC can be compared.
3. Hold the preprocessing, spatial filter, and classifier constant. Vary the number of averaged rounds of intensifications used to determine row and column eliminations and plot the effect of this on classification accuracy. Also vary the number of rows and columns eliminated after n averaged rounds of intensification. Determine the

effect on classification accuracy. Classification speed will increase with fewer rounds of intensification and higher numbers of rows and columns eliminated.

3 Methods

Section 2.2 concluded with a list of specific project objectives. The first item on that objective list mentioned the BCI 2003 Competition Dataset. Section 3.1 provides a description of this dataset, which is commonly used in studying P300 classification. The two other sections in this chapter, Section 3.2 and 3.3, provide a description of data collection and data processing, respectively. The following chapter, Results, will discuss findings.

3.1 BCI 2003 Competition Dataset

The BCI 2003 Competition Dataset IIb was recorded at the Wadworth Center of the New York State Department of Health. It is available online and has been widely used in research [21]. The file is packaged with the data and is accessible from (`http://www.bbci.de/competition/iii/`). The data were collected from two subjects over five sessions per subject. Subjects were instructed to focus on a specific character as the rows and columns of the 6×6 speller matrix (described in Section 1.2.3) were randomly intensified.

Before the start of each character trial, the speller matrix was displayed with uniform intensity (i.e., the matrix was blank) for 2.5 seconds. Subsequent row/column intensifications were 100 ms in duration with an ISI of 75 ms. Fifteen flashings of each row and column were performed per character, resulting in a total of 180 intensification per character epoch. The matrix was blank for 2.5 seconds after completion of an epoch. Signals were recorded from a 64 channel surface electrode array. Before signals were digitized, they were bandpass filtered from 0.1 to 60 Hz. The sampling rate was 240 Hz. For each of the two subjects, `*.mat` data files have been created which contain the raw signals (85 training and 100 test characters per subject).

Table 3. Description of variables in data file.

Variable	Description
`Flashing`	Takes value 1 when row/column is intensified, 0 otherwise.
`StimulusCode`	Takes value 0 when matrix is blank. Takes a value between 1 and 12 if a row/column is intensified. The values 1 thru 12 correspond to the rows and columns in the P300 speller matrix (Figure 3).
`StimulusType`	Takes value 0 when matrix is blank or row/column does not contain desired character. Takes a value between 1 when intensified row/column does contain desired character.
`TargetChar`	The correct character label for each epoch in the training data.

3.2 Data Collection

P300 ERP data are collected from two sources simultaneously (Emotiv EPOC and Grass amplifiers). Data from both sources are collected on a single computer. The BCI dataset collection procedure described in Section 3.1 is followed as closely as possible.

3.2.1 Acquiring P300 Data from Emotiv EPOC Headset

OpenViBE is free, open-source BCI software. It was developed in a collaborative effort by researchers at several French Universities [41]. There is an implementation of the P300 speller matrix available for OpenViBE. It is possible to export intensification tags which indicate exactly when a row or column was intensified to a COM port with OpenViBE. Users can also set parameters like flash interval and ISI. The EPOC has its own proprietary data collection suite, TestBench, which can receive intensification tags when given a COM port number. It is possible to send tag data from OpenViBE to TestBench. TestBench records raw EEG from all electrodes on the EPOC in `*.edf` file format. Emotiv provides a tool which converts this data file into the more popular `*.csv` format. The data collection proceeds as follows:

1. The Emotiv EPOC saline electrodes are pre-moistened and attached to the wireless headset.
2. The patient enters the recording room and sits facing a large flatscreen monitor.

3. The headset is placed (backwards) on the head of the subject. The hair underneath the contacting electrodes is moved aside as much as possible.

4. At this point, the Grass gold cup electrodes are installed adjacent to the EPOC recording electrodes according to the procedure outlined in the following Section 3.2.2.

5. TestBench recording software is initialized and provided with the COM port number to receive intensification tags (Target and Non-Target) from OpenViBE. The EPOC is powered on. The Grass amplifier is powered on and the recording software is initialized (see Section 3.2.2).

6. The patient is instructed to mentally count the number of times the target character is highlighted during the row/column intensifications of the P300 matrix. The target character to spell appears on the screen highlighted in green prior to each round of row/column intensifications (there are fifteen rounds of intensifications, so each row and column is flashed at least fifteen times per character).

7. Data recording begins and the OpenViBE simulation is started.

8. To avoid patient fatigue, only 15-20 characters are spelled during each recording session (approximately 30 minutes to record 20 characters).

The recording takes place in a dimly lit room with a high contrast screen. Flashing and between flash times are 210 ms and 265 ms, respectively. Small fluctuations in flash timing are tolerable due to the COM port data tagging mechanism. During the flashing, the patient sits at eye level one meter from the spelling matrix, which is approximately 30.5 cm wide and 28 cm in height. Thirty-five spelled characters (with 15 rounds of intensification for each character) are collected from each patient. The collection process takes place in two parts (20 and 15 characters) with a five minute break in between. The electrodes are not removed during the break but the overhead lights are illuminated. The time to spell 35 characters takes about an hour.

3.2.2 Acquiring P300 Data from Grass Amplifiers

After the EPOC headset has been placed on the patient's head (see Steps 3 and 4 in Section 3.2.1), eight Grass gold cup electrodes are placed on the patient's head using Tensive ® conductive adhesive gel and medical tape. The electrodes are located 1 mm above each saline electrode (i.e., closer to the top and center of the head). A cloth cap is placed over

the electrodes to keep them on the head. The EPOC electrodes fit through holes on the cloth cap to make contact with the scalp (see Figure 9).

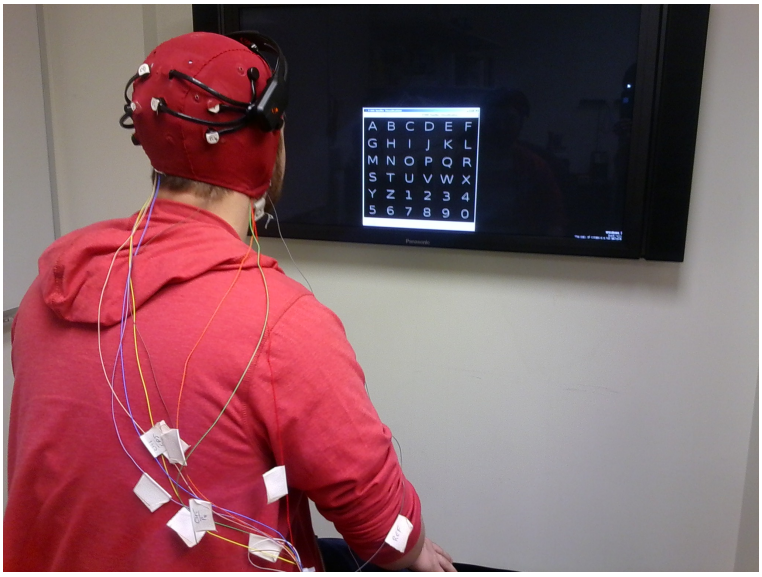

Figure 9: A subject is seated facing the screen with the 6×6 character matrix displayed. The subject is wearing a cloth cap which helps to secure the gold cup electrodes. Holes have been cut in the cap for the EPOC electrodes to contact the scalp. The overhead lighting is turned on in this picture, but is turned off during actual trials.

Each gold cup electrode transmits EEG data to its individual analog Grass amplifier in the Grass Model 12 Neurodata Acquisition System, which is set to bandpass filter the signals between 0.1 Hz and 300 Hz. The amplifier has a notch filter at 60 Hz for power line interference. After filtering, the signals are amplified 20,000 times and passed to a National Instruments SCB-68 analog to digital converter. SignalExpress (a LabView product) is used to interface the converter with the computer and save the data in a text file. Data are sampled at 240 Hz.

To ensure proper synchronization of data (i.e., to tag data precisely each time a row or column is intensified), OpenViBE is configured to export tag data from a COM port on the simulation computer. This COM port tag occurs at the exact same time as another tag is exported to TestBench (see Section 3.2.1). A circuit has been built to transform this serial port signal from an 11 V 10 ms square pulse (Clear to Send pin) to a 1.5 V 10 ms square pulse. The output is hard wired to a ninth channel in the National Instruments SCB-68.

3.2.3 Data Collection Summary

Figure 10 summarizes all parts of the experimental setup. Arrows with the label 'COM Port Tag' represent target/non-target labels generated by OpenViBE at the time a given row or column is flashed. Arrows which are not labeled stand for EEG signals. The red and blue boxes enclose the recording hardware and software, respectively.

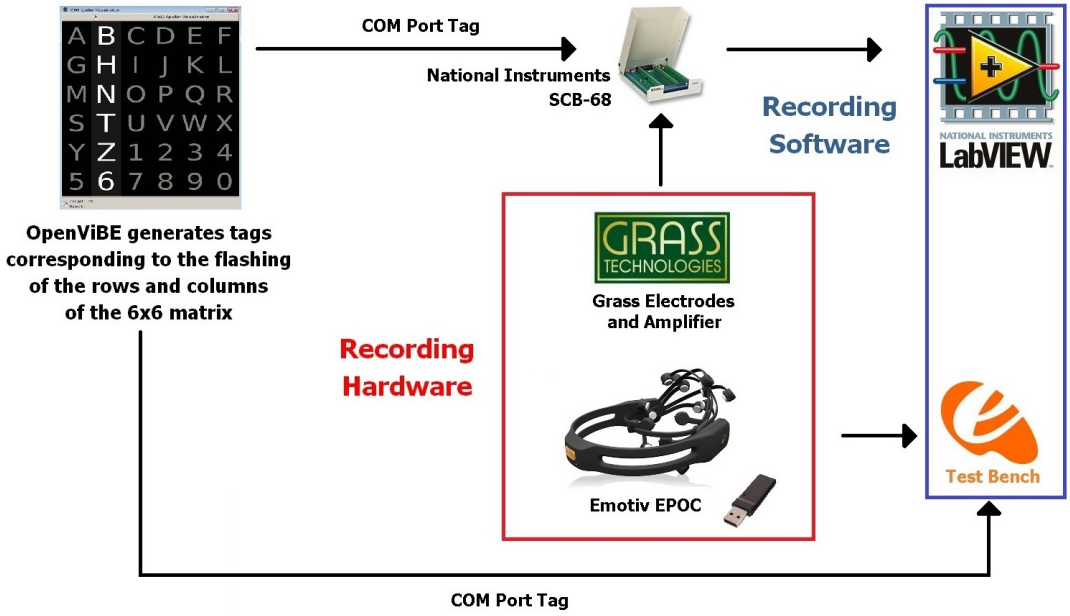

Figure 10: Summary of the experimental setup. Arrows represent signals (either COM port tags from OpenViBE or EEG data from the EPOC or Grass System).

3.3 Data Processing

For purposes of this project, data processing is completed in MATLAB.

3.3.1 Preprocessing P300 Data with MATLAB

Several preprocessing steps are performed on all data (this includes data from the BCI 2003 Dataset, EPOC data, and Grass amplifier data). For a more detailed description of each of these steps, see EEG Preprocessing, Section 2.1.1.

1. Only 8 parietal electrode traces from the data are considered: CP_5, CP_6, P_7, P_3, P_4, P_8, PO_3, and PO_4. These positions correspond to electrode locations from which the EPOC is capable of recording.
2. The data are filtered to the range 0.1 to 15 Hz with a fourth order Butterworth filter.
3. The data are downsampled to 40 Hz, thereby leaving 40 features for a one second epoch.
4. The time window selected is from stimulus onset to 1 second after the stimulus. (Therefore each epoch will contain 40 samples due to the downsampling in the previous step).
5. Data are scaled to the interval -1 to 1.
6. The data are averaged over n intensification cycles. For example, for a given character epoch, a row or column is intensified n times (e.g., 2 or 15) and the response for each individual electrode for that row or column is averaged. This is the most common way of dealing with noise in P300 based BCI.
7. If a spatial filtering method (e.g., ICA) is to be used, it is applied at this point.

Leave-one-out cross validation is used to determine classifier accuracies: a given classifier is trained on 34 characters and then tested on the 35th character. This process is repeated 35 times until all of the characters in the data set are tested. This method is used to gauge accuracy of the BCI dataset and the collected data (EPOC and Grass Amplifier Data). Thirty five characters have been randomly selected from the 85 character BCI dataset to be tested.

3.3.2 MATLAB Implementation of Spatial Filter and Classification Algorithms

A spatial filter is an algorithm which reduces dimensionality of data (Figure 11). For EEG data, it is assumed that each electrode 'hears' *some* of what the other electrodes are hearing. The commonalities between the electrodes can be used to construct a 'component' (or components) which explains most of the data. In Figure 11, eight parietal electrode traces are used to construct such a component. A detailed overview of common spatial filtering algorithms used in P300 EEG data analysis can be found in Section 2.1.3. The MATLAB implementation of Grand Averaging, PCA, and ICA are described in Table 4.

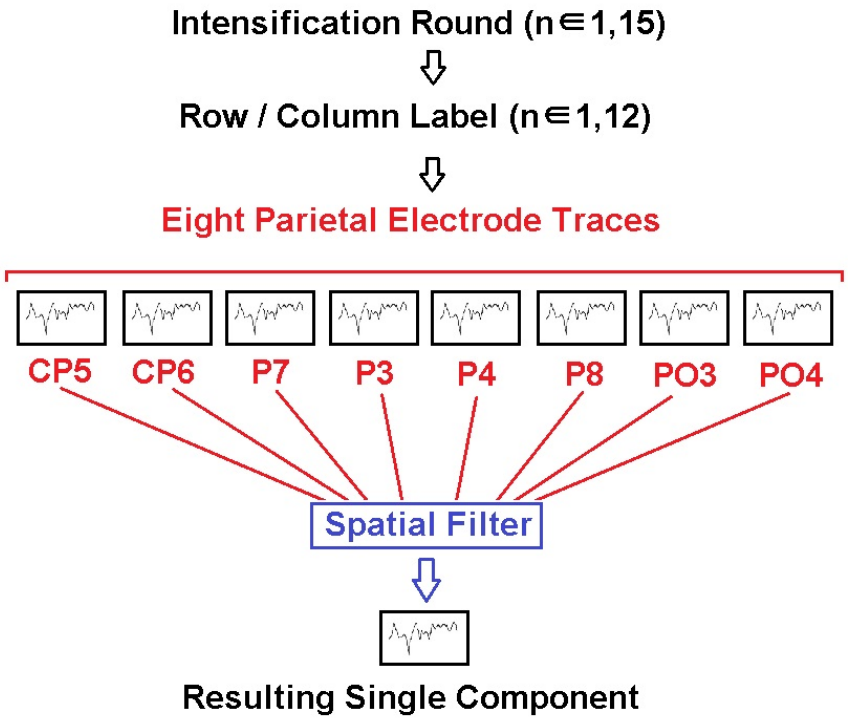

Figure 11: Illustration of how a spatial filter reduces the dimensionality of EEG data.

Table 4. Description of spatial filtering algorithm implementation.

Algorithm	Description of MATLAB Implementation
Grand Averaging	The average value across all eight parietal electrodes for a given sample is selected as the value for that sample in the resulting single component.
PCA (Principle Component Analysis)	The command `princomp` is used to generate the principle components for eight electrodes in an epoch. Only the first principle component is used.
ICA (Independent Component Analysis)	The FastICA version 2.5 package available from Aalto University (`http://research.ics.aalto.fi/ica/fastica/`) was used to generate the first principle component for eight electrodes in an epoch. The component data were re-scaled between -1 and 1 after calculation.

If a spatial filter is not used to reduce the dimensionality of the data, the alternative is to construct eight classifiers (one for each electrode) and make the 'Target' or 'Non-Target' decision based on a majority vote. This idea is pictured below in Figure 12.

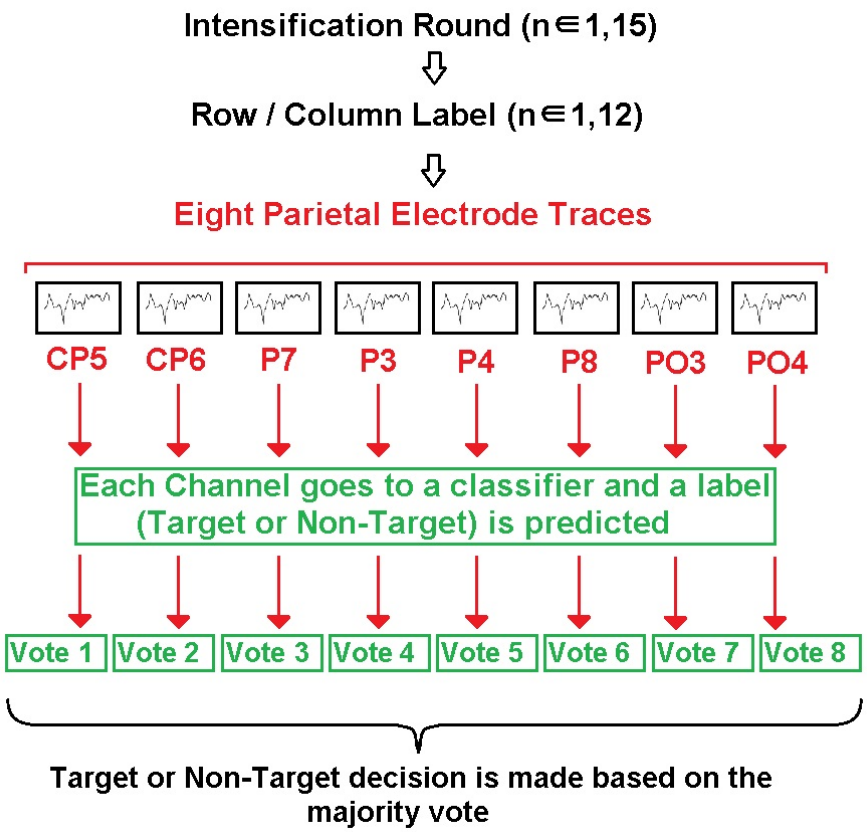

Figure 12: Illustration of how multiple classifiers can be trained on each channel. Each of the eight classifiers votes 'Target' or 'Non-target.' The majority vote wins.

Four different types of classification algorithms are studied in this project. They are PCM, FLDA, BLDA, and SVM. Table 5 describes the MATLAB implementation of each.

Table 5. Description of classification algorithm implementation.

Algorithm	Description of MATLAB Implementation
PCM (Pearson's Correlation Method)	The correlation coefficient, r, between each new sample row or column and all target rows and columns was computed using MATLAB commands `cov` and `std` (see Equation 12). The average of all such correlations was taken to be the r value for that sample. The highest two values of r were predicted to be the target row and column of the P300 matrix.
FLDA (Fisher's Linear Discriminant Analysis)	The binary FLDA classifier used in this project was obtained from the MATLAB file exchange (`http://www.mathworks.com/matlabcentral/fileexchange/42785-binary-fisher-lda`). The code uses MATLAB functions `cov` and `mean` to find the covariance and mean, respectively. Once each weight w had been computed from training data, each test sample feature x was multiplied by its respective weight (Equation 13). The sum of all features multiplied by their respective weights was then computed. The maximum two such values for a given character epoch were chosen as the predicted row and column.
BLDA (Bayesian Linear Discriminant Analysis)	The MATLAB command `classify` with the optional string argument `'diaglinear'` was used to implement a linear Naive Bayes classifier. Target rows and columns were determined by selecting the two maximum posterior probabilities from this calculation.
SVM (Support Vector Machines)	An SVM with a linear kernel function (dot product) was implemented via the MATLAB command `svmtrain`. The distances of each test point to the separating hyperplane were computed using the `svmtrain` object. The two distances with the maximum value (i.e., furthest from the separating hyperplane and thereby most likely to be targets) were predicted to be the target row and column. In order to force the convergence on a solution, the Karush-Kuhn-Tucker (KKT) conditions had to be relaxed. Ten percent of data points were allowed to violate the conditions. Additionally, the soft margin `'boxconstraint'` was changed from 1 to 0.25.

In Section 4.1.1 the results of using the spatial filtering algorithms described in Table 4 and the classification algorithms in Table 5 on the BCI Dataset are reported. The accuracy of each combination of algorithm (spatial filter and classification algorithm) and the computational power required to implement each on the BCI 2003 Dataset IIb are measured. The percent accuracy measures were obtained by performing leave-one-out cross validation on all 35 characters.

3.3.3 Implementation of Row and Column Eliminating Algorithms

The goal of each algorithm is to choose which rows and columns are not likely to be targets (i.e., not likely to contain the P300 signal) after a low number of intensifications. There are four different algorithms which are tested. The algorithms were chosen because they were not computationally expensive in comparison to other algorithms (refer to Figure 17). The algorithms are:

1. A PCA spatial filter with a BLDA classifier.
2. A PCA spatial filter with an FLDA classifier.
3. A parallel BLDA classifier (see Figure 12).
4. A parallel FLDA classifier.

The algorithms are programmed to select n least likely rows or columns. In the case of BLDA, this is accomplished by selecting the lowest posterior probabilities for n rows and columns out of a total 12. In the case of FLDA, this is done by selecting the n smallest sums of weight w multiplied by sample feature x. Refer to Table 5 in the previous section for a description of the MATLAB commands.

4 Results

The goal of this project is to discover an algorithm which can reliably exclude non-target rows and columns from the P300 speller matrix after a small (e.g., two) number of intensifications. The computationally inexpensive algorithm must successfully determine a percentage of non-targets (e.g., 50% of the rows and columns for a given character epoch). If the algorithm is successful in this task with data taken from the wireless EPOC headset, then that demonstrates the feasibility of a mobile P300 based BCI with higher throughput and lower computational complexity versus the current state of the art.

The first subsection in Results (Section 4.1) presents character prediction accuracies of spatial filtering and classification algorithm combinations. Computational complexity of those combination is also presented. Subsections after 4.1 report results on data collected by the researcher from subjects at Temple University. There are two categories of findings presented in those subsections: the character prediction accuracies of various spatial filtering and classification algorithms, and the error rate of various row and column eliminating algorithms.

4.1 BCI Dataset

The 2003 BCI Dataset IIb was the dataset used in the 2003 BCI Competition. It is a publicly available resource and is used as a proof of concept in this thesis. The following Figure 13 comes from Subject A of the dataset. It is an average of all the target signals (85 character epochs) from one electrode, C_z. The P300 response is clearly visible in the blue 'Targets' trace. The y-axis shows that the amplitude of the response is about 3 μV, and the x-axis shows that the peak occurs 300 ms after the stimulus.

The BCI data contains recordings from 64 positions sampled at 240 Hz. In order to ensure that the simulations with the BCI dataset can be compared to the Emotiv EPOC and the Grass amplifiers, only eight parietal electrodes from the BCI Dataset (corresponding to the electrodes on the EPOC) are used for analysis.

Once the BCI data are preprocessed as in Section 3.3.1, the accuracy of the spatial

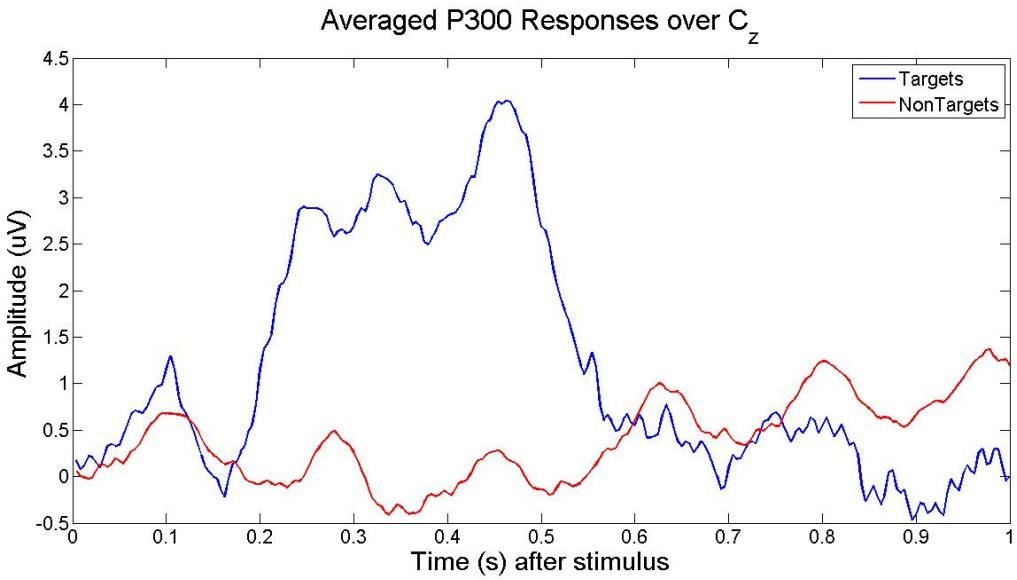

Figure 13: This plot uses data from the BCI 2003 Dataset IIb Subject A. The blue and red traces are the average of all the target and non-target epochs (85 characters), respectively. The data were only averaged from one central electrode, C_z.

filtering and classification algorithms are evaluated. Because there are only 35 characters available for each subject, leave-one-out cross validation is used: a given classifier is trained on 34 characters and then tested on the 35th character. This process is repeated 35 times until all of the characters in the data set are tested. For this preliminary work, two and fifteen intensification cycles are tested.

4.1.1 Accuracy and Performance of Select Spatial Filtering and Classification Algorithm Combinations

In the previous Section 4.1, it was mentioned that the BCI dataset contains 64 electrode traces (refer to Figure 5 for their names and 10-20 locations). In order to compare BCI data results with EPOC and Grass amplifier data, this number was reduced to only the eight parietal electrodes from which the EPOC and Grass amplifiers are capable of recording (these positions are indicated in Section 3.3.1). The eight positions were chosen based on previous work (refer to Section 2.1.2). This type of analysis is problematic, however, because it is possible that not all electrode traces in the BCI dataset are of equal recording

quality. This is evident when comparing averaged ERP's from two proximal electrodes: C_z (Figure 13) and CP_5 (Figure 14).

Figure 13 depicts the averaged recordings of the central electrode, C_z, from the BCI dataset. In the Figure, it is simple to visually distinguish between the target P300 signal (blue trace) and the nontarget signal (red trace). In contrast, Figure 14 is a plot of a different electrode trace from the same subject and same experiment. The electrode, CP_5, is one of the parietal electrodes from the BCI dataset which is used in the analysis. Figure 14 was generated in the exact same way as Figure 13, but the difference between the target and non-target traces is not as clear. Marked variability in signal quality can occur even though CP_5 is very close to C_z on the scalp. There is no good explanation as to why two electrodes so close to each other acquire such drastically different waveforms, but the effect on classification accuracy is clear. Better results are achieved by using just one 'good' electrode than by using eight inferior electrodes.

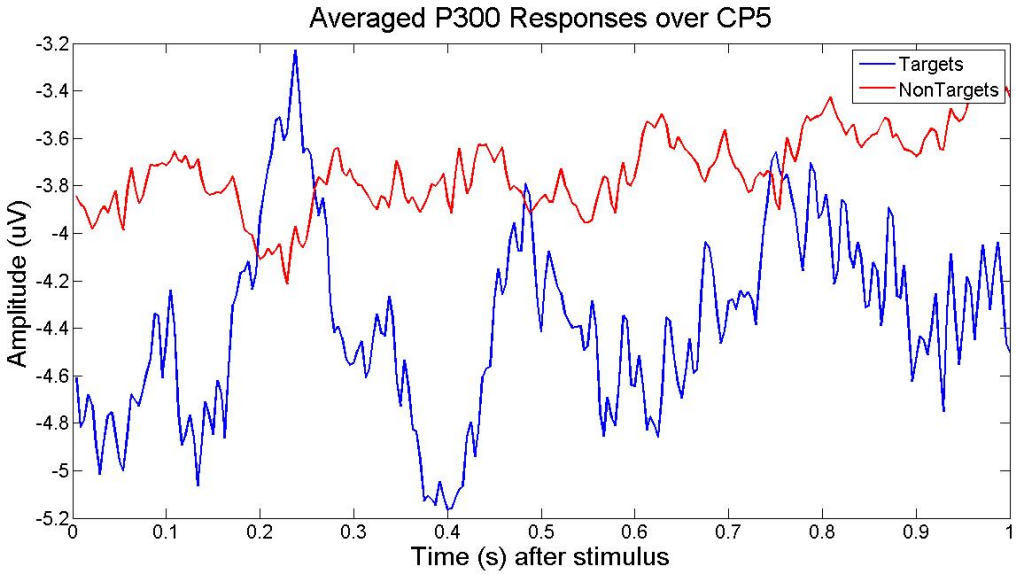

Figure 14: This plot uses data from the BCI 2003 Dataset IIb Subject A. The blue and red traces are the average of all the target and non-target epochs (85 characters), respectively. The data were only averaged from one electrode, CP_5.

When only the electrode trace C_z is used as an input to the classification algorithms, the resulting accuracies are good (refer to Figure 15). The bars on the left half of the figure

represent the character accuracy (the odds of selecting a character correctly at random are $\frac{1}{36}$, or 2.8%). The bars to the right represent the row/column accuracy (the odds of selecting a row or column correctly at random are $\frac{1}{6}$, or 16.7%). The graph shows that the character accuracy achieved by FLDA and SVM using only electrode C_z is near 40%. The highest accuracy achieved with the eight parietal electrode traces, on the other hand, is only 30% (see Figure 18). The accuracies achieved when the Grass Technologies amplifiers were used to collect data at Temple University from those same electrode positions are much higher than the BCI dataset results (the best participant scored 90% accuracy). It is impossible to know whether this discrepancy is the result of differences in experiment preparation (e.g., quality of electrode contact), inter-patient variability, the experimental environment variables, or some other cause.

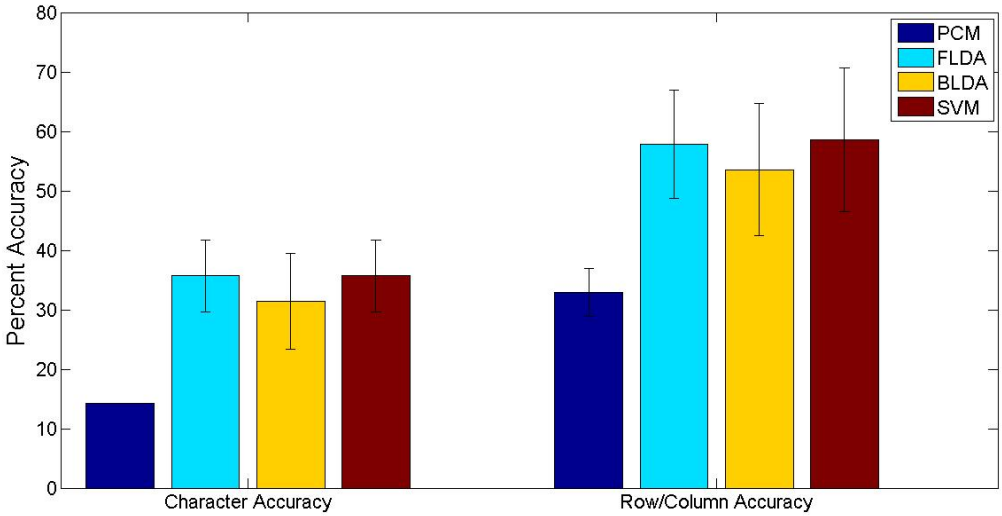

Figure 15: This bar graph shows the results from classification of BCI data for Subjects A and B electrode C_z. The data were averaged over 15 intensification cycles before classification. The error bars represent one standard deviation from the mean. The variance is across subject percent accuracies, not trials (e.g., FLDA 35% character accuracy can be interpreted as: 'The mean between the accuracies that two subjects achieved individually').

The winners of the 2003 BCI competition achieved 96.5% character accuracy on $n = 15$ sequences using SVM ensembles on all 'important' channels. These optimal channels were selected using a recursive channel elimination algorithm on all 64 channels. The same

algorithm which produced the 96.5% accuracy achieved lower accuracies when tested on all channels and specific subsets of channels not determined by the recursive algorithm. The results from their work are shown in Figure 16. [42]

	Performances after 5 and 15 sequences					
Dataset	Optimal channels		64 channels		8 channels	
A_1	26	66	22	55	24	60
A_2	41	69	22	61	15	54
A_3	28	64	19	59	5	27
A_4	36	81	24	56	17	53
A_5	39	75	27	69	23	52
B_1	62	93	52	80	41	76
B_2	61	90	49	73	31	54
B_3	56	81	45	65	36	65
B_4	57	89	49	81	47	65
B_5	53	89	59	88	33	70

Figure 16: Character percent correct accuracies of the algorithm which won the 2003 BCI Competition for different numbers of electrodes after $n = 5$ and $n = 15$ sequences. Here the data for Subjects A and B (85 characters each) have been subdivided into groups of five characters each (A_i and B_i where $i = 1...5$). The training data for a particular trial were constructed from a single partition (12 rows/columns $\times$ 15 intensifications $\times$ 5 characters = 900 examples) and the testing data were formed from the other partitions. The eight channels referred to are: F_z, C_z, P_z, C_3, C_4, P_3, P_4, and O_z. [42]

The winners used 15300 examples to train the algorithm which won the 2003 competition (85 characters $\times$ 15 intensifications $\times$ 12 rows/columns = 15300 examples). In this thesis, leave-one-out validation was used on 35 characters, and epochs were averaged before training the algorithms on eight parietal channels. In summary, training for a given algorithm in a single leave-one-out cycle was accomplished with 408 examples (34 characters $\times$ 12 rows/columns = 408).

When considering this difference in the number of training samples and also considering the difference in the quality of electrodes used for analysis, it is not possible to directly

compare the BCI winner's results to the results in Figure 18. So, while the BCI dataset does not prove valuable in this sense, it is valuable in another. One important use of the dataset is benchmarking of spatial filtering and classification algorithm combinations in terms of their relative computational complexity. Below is a graph of the time (in seconds) taken by a computer to execute combinations of select spatial filtering and classification algorithms for a set amount of data. The computations are performed on the same 64 bit machine in the same session to avoid variability. The computer which is used for the simulations has 3 Gb of RAM and a 2.10 GHz dual core processor. The clock for the computation time is started before the spatial filtering and stopped after the final classification percent correct measurement is returned. The MATLAB commands `tic` and `toc` are used to time the experiments.

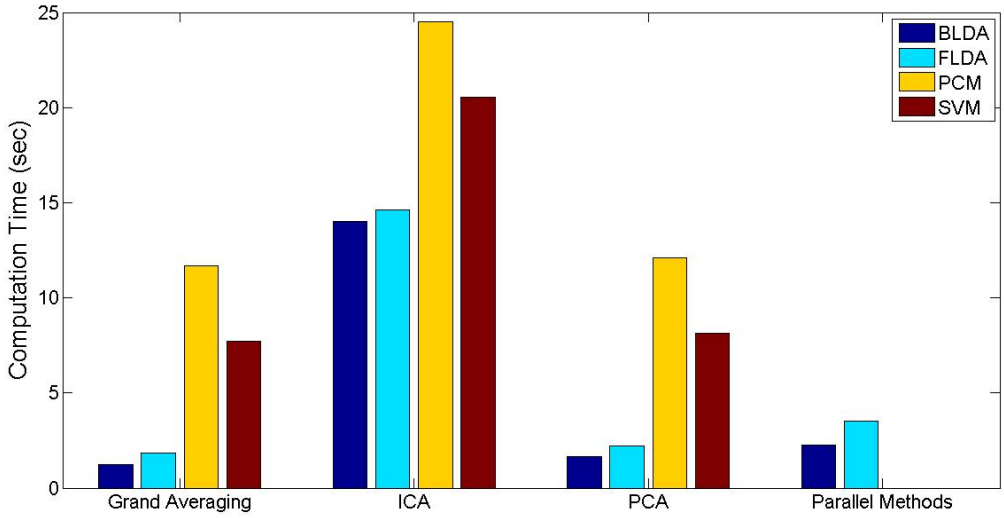

Figure 17: The relative computational complexity of each combination of spatial filtering and classification algorithm tested on the BCI dataset. Time is on the y axis and is measured in seconds.

From Figure 17, it is apparent that ICA is not the best choice for a spatial filter. In comparison to the other spatial filters, the ICA computations take a long time. Unfortunately for ICA, this delay does not result in an improved classification accuracy, which will be shown later. PCA and Grand Averaging seem to be evenly matched; it takes significantly

less time to implement them than it does to run ICA. This is interesting because PCA and ICA are closely related. The discrepancy could be attributed to the efficiency of the code: the PCA algorithm is developed by the MathWorks, but the algorithm used to implement ICA was obtained from a University in Finland. Parallel BLDA and FLDA do not use a spatial filter. Instead they construct a classifier for each individual electrode trace (refer to Figure 12). This takes just slightly more time than using Grand Averaging or PCA in combination with a single classifier.

The graphs on the next page (Figure 18) are the aforementioned graphs of classifier accuracy on the BCI dataset subjects A and B. Each graph depicts character accuracy when the number of averaged intensification cycles are varied, the spatial filter is varied, and the classification algorithm is varied.

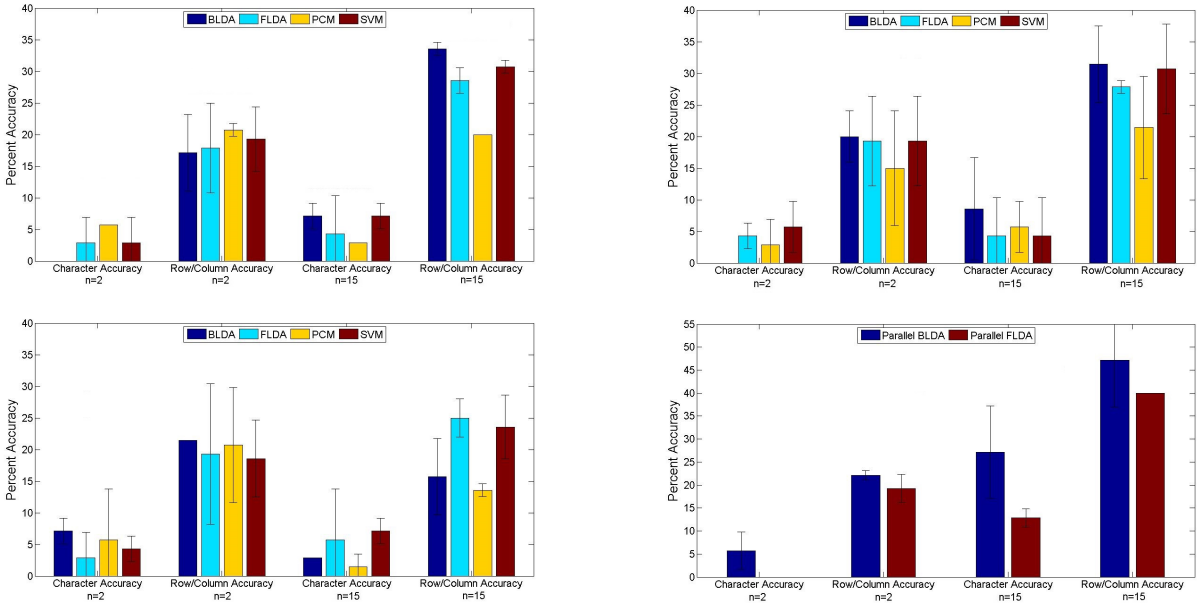

Figure 18: The percent accuracy (character and row/column accuracy) of four classification algorithms when Grand Averaging (top left), PCA (top right), ICA (bottom left), and Parallel Voting Algorithms (bottom right) are used as a spatial filters on Subjects A and B of the BCI dataset. The x axis labels $n = 2$ and $n = 15$ refer to the averaged number of intensifications used to generate the input data for the classifiers. The variance is across subject percent accuracies, not trials

According to Figure 18, when Grand Averaging and PCA are used as spatial filters,

BLDA and SVM are the best performing classification algorithms. The parallel methods are the best overall performers (see Figure 18, bottom right) with parallel BLDA achieving the highest character accuracy of near 30% for $n = 15$ averaged intensifications. When ICA is used as a spatial filter, a notable increase in accuracy with fewer intensifications is observed.

A series of one-way ANOVA tests were run in MATLAB to determine the statistical significance of these findings. Using the command `anova1`, it is possible to determine whether two sets of data come from the same population. If the p-value returned by the test is greater than a certain threshold, the null hypothesis (that the two sets of data come from the same population) is confirmed. Common thresholds are $p < 0.05$ and $p < 0.01$. We can reject the null hypothesis with more confidence when the p-value is small.

The results of the test show that the poor performance of ICA in comparison to PCA, Grand Averaging, and parallel methods, is statistically significant ($p < 0.05$) when considering the $n = 15$ row/column accuracy results. Additionally, the superior performance of the parallel methods with respect to all other spatial filters is statistically significant ($p < 0.05$), again when considering the $n = 15$ row/column accuracy results. These results give an indication of what can be expected from collected data in the following sections.

4.2 Collected Data - Part One

After collecting data from the first three participants, the collection procedure was refined in order to improve the quality of the recordings. It was discovered that there was a persistent 2.5 Hz noise present in the Grass amplifier recordings of the first three subjects (the EPOC recordings were unaffected). The noise was consistent in its periodicity and amplitude. It was eliminated before data were collected from participants in Section 4.3.

The presence of the 2.5 Hz noise does not have a great adverse effect on the classification accuracy. This is due to the nature of machine learning algorithms. If a noise source is consistent and predictable, the algorithms learn to ignore it in the training period. A simple example of this is linear regression. If a linear regression model is trained on ten features

and only three of those features are relevant (i.e., only those three affect the output), then the weights assigned to the features which are not relevant will be negligible. Even though this is true, for the sake of data integrity, results are divided into two sections.

4.2.1 Accuracy of Select Spatial Filtering and Classification Algorithm Combinations

The following figures (Figures 19 through 22) show classification accuracy results for the first three subjects for $n = 2$ and $n = 15$ averaged intensifications. In each graph, the data are averaged across subject percent accuracies, not characters spelled, to produce the percent accuracies. For example, the percent correct value of the parallel FLDA algorithm in Figure 22, which is 70%, was calculated by averaging the percent correct value attained by each of the three subjects using the parallel FLDA algorithm on data recorded from the Grass amplifiers.

Each figure contains two bar graphs: the top bar graph contains the results for the EPOC and the bottom contains the results for the Grass amplifiers. A unique feature of this project is the collection of EEG data from two sources simultaneously (collection procedure described in Section 3.2): the EPOC from Emotiv Systems, and the high quality amplifiers from Grass Technologies. As in the previous section, the errors bars on each graph represent one standard deviation from the mean. ANOVA tests were run using the command `anova1` in MATLAB.

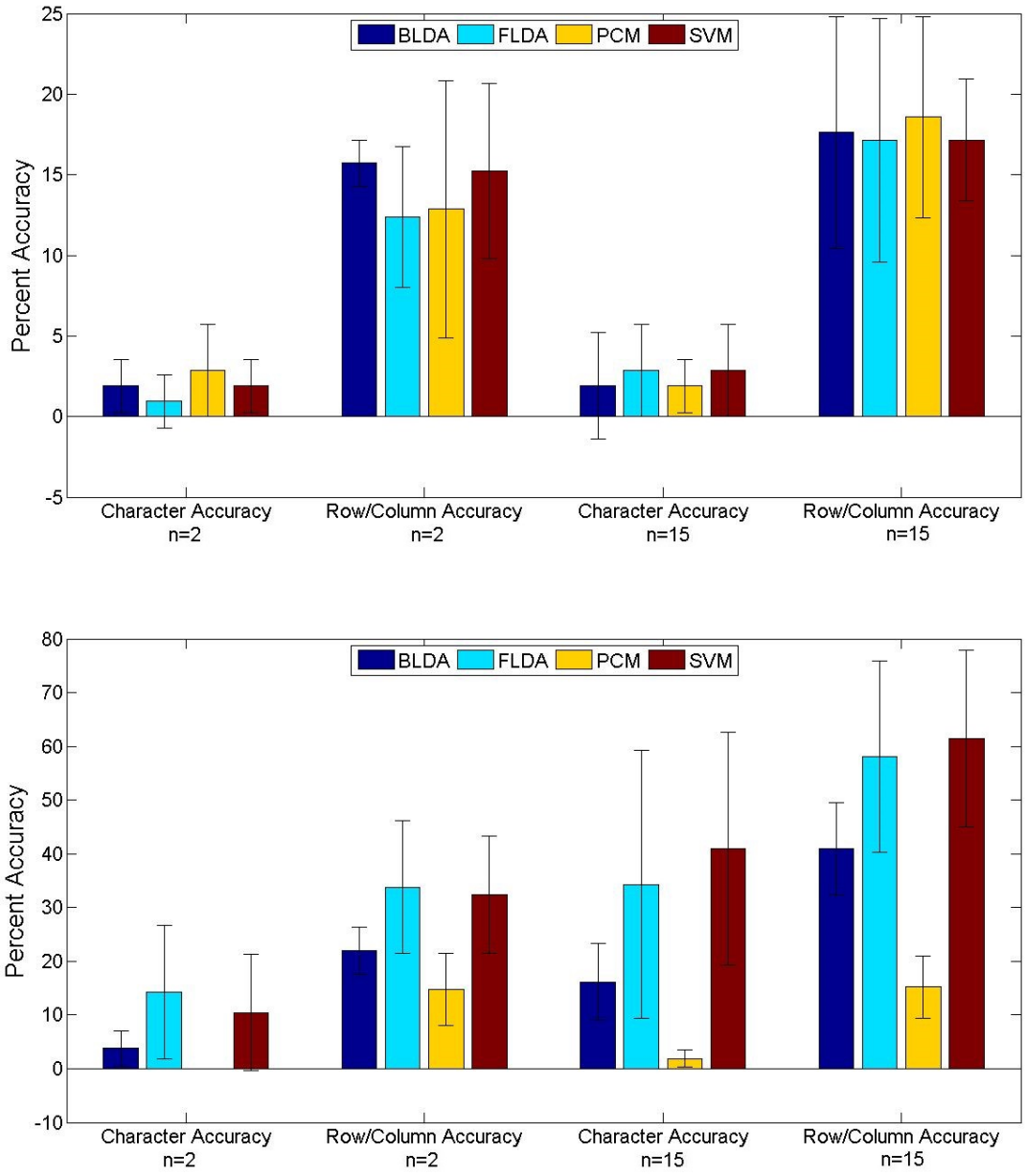

Figure 19: Grand averaging was used as a spatial filter for these data (from three subjects). These graphs show the percent accuracy (character and row/column accuracy) of four classification algorithms when the Emotiv EPOC (top graph) is used to collect data and when the Grass amplifiers (bottom graph) are used to collect data. The x axis labels $n = 2$ and $n = 15$ refer to the averaged number of intensifications used to generate the input data for the classifiers.

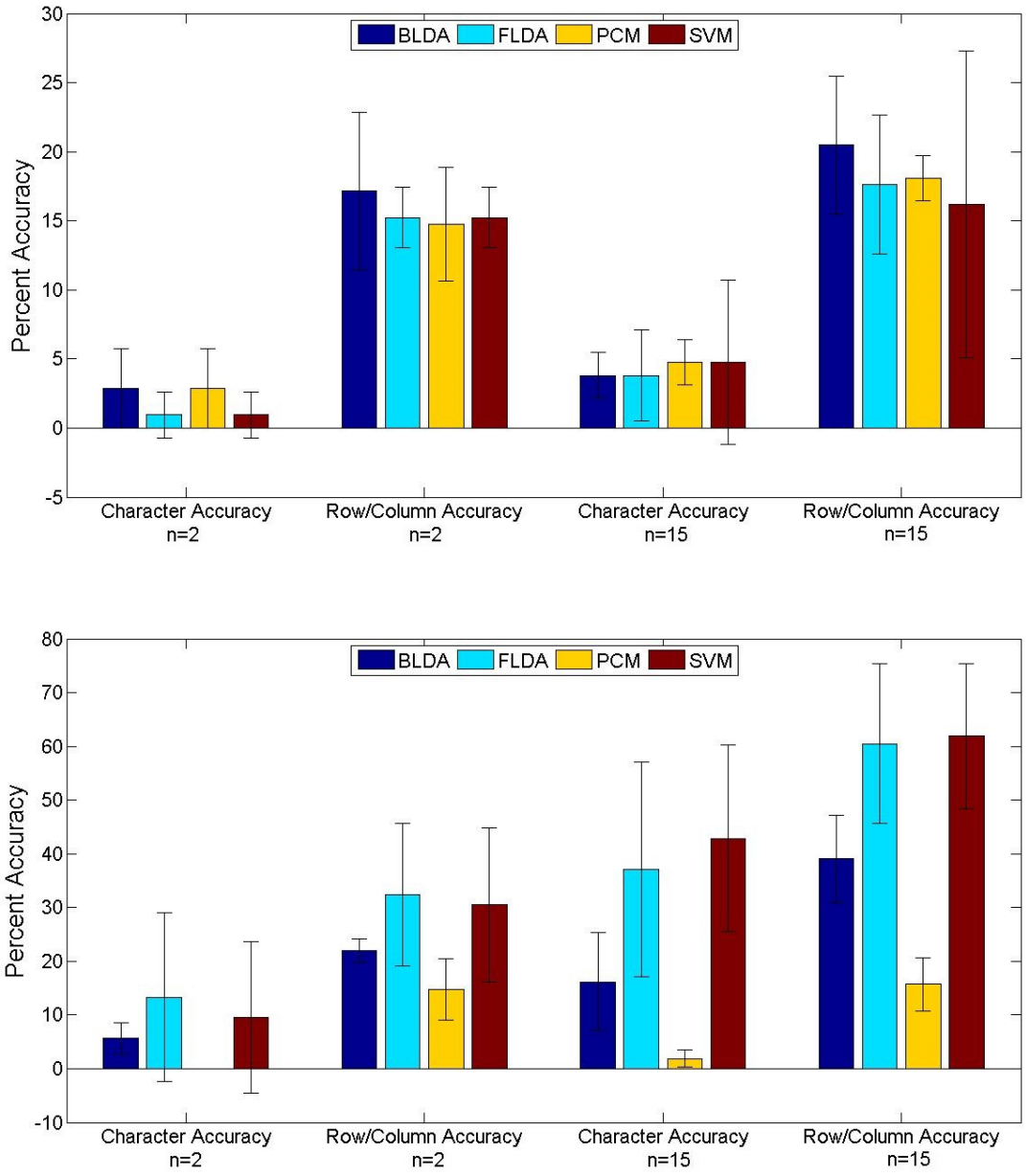

Figure 20: PCA was used as a spatial filter for these data (from three subjects). These graphs show the percent accuracy (character and row/column accuracy) of four classification algorithms when the Emotiv EPOC (top graph) is used to collect data and when the Grass amplifiers (bottom graph) are used to collect data. The x axis labels $n = 2$ and $n = 15$ refer to the averaged number of intensifications used to generate the input data for the classifiers.

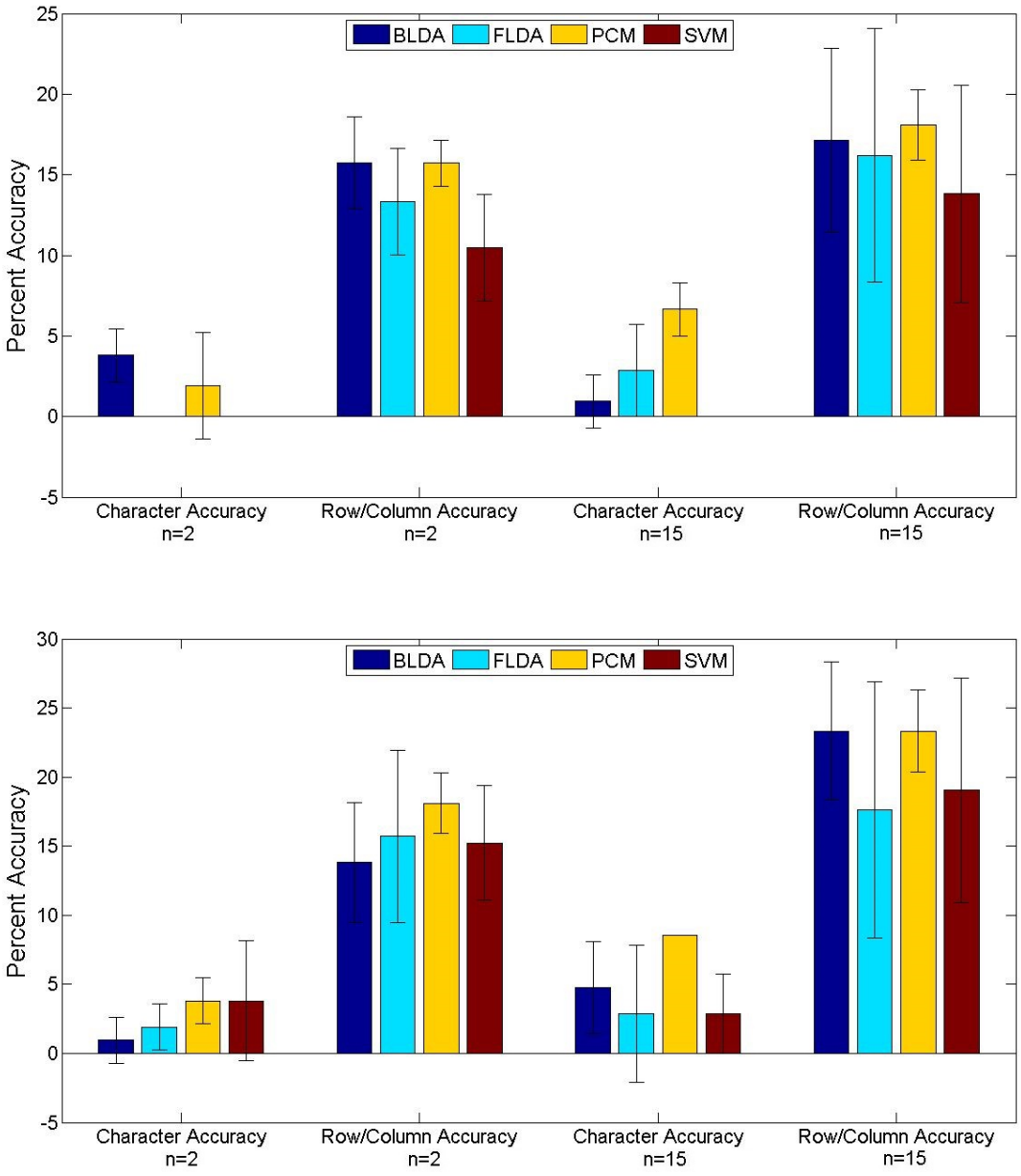

Figure 21: ICA was used as a spatial filter for these data (from three subjects). These graphs show the percent accuracy (character and row/column accuracy) of four classification algorithms when the Emotiv EPOC (top graph) is used to collect data and when the Grass amplifiers (bottom graph) are used to collect data. The x axis labels $n = 2$ and $n = 15$ refer to the averaged number of intensifications used to generate the input data for the classifiers.

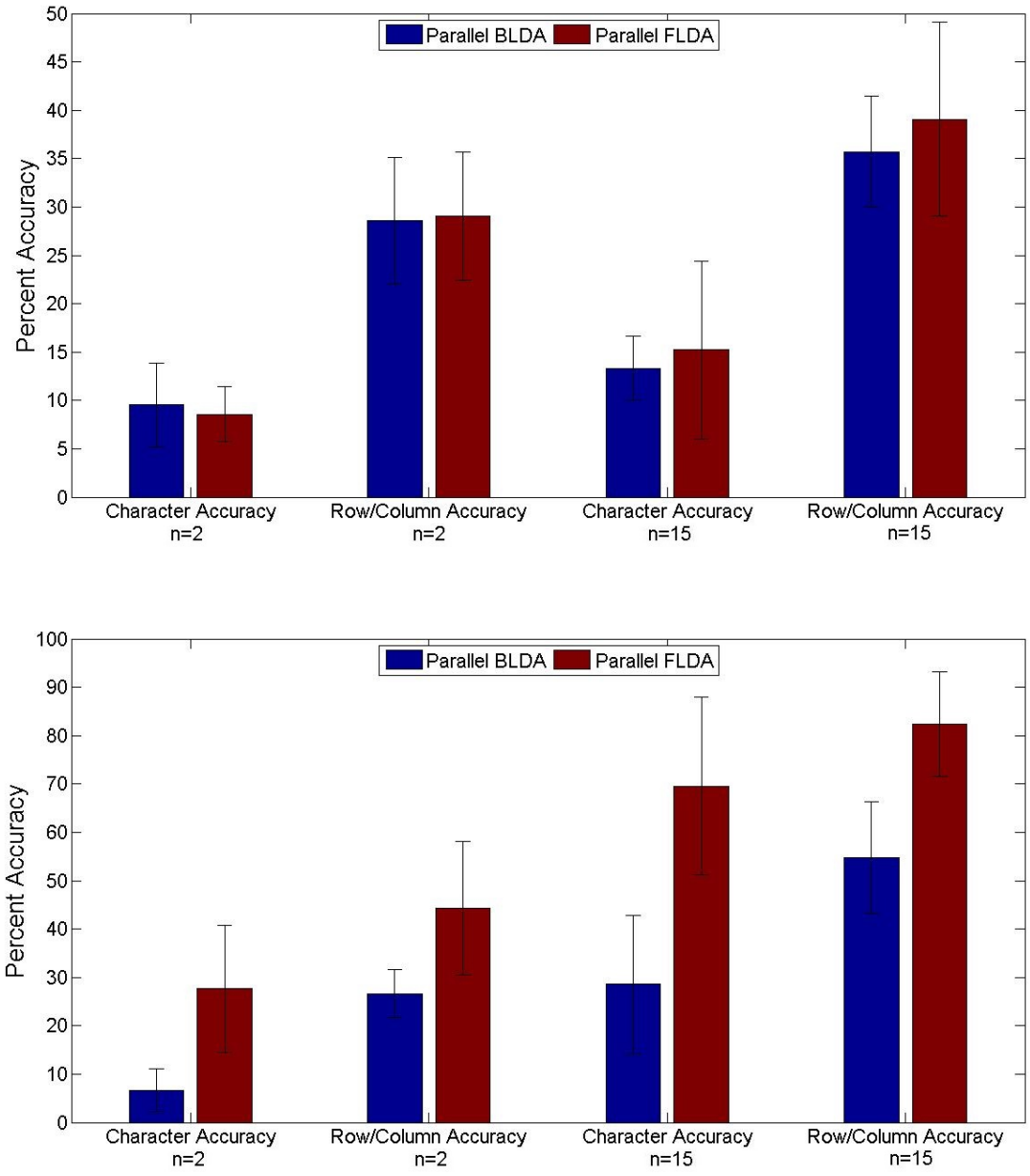

Figure 22: No spatial filter was used on these data (from three subjects). These graphs show the percent accuracy (character and row/column accuracy) of two parallel classification algorithms when the Emotiv EPOC (top graph) is used to collect data and when the Grass amplifiers (bottom graph) are used to collect data. The x axis labels $n = 2$ and $n = 15$ refer to the averaged number of intensifications used to generate the input data for the classifiers.

The first result which is apparent from these data is that the Emotiv EPOC is capable of detecting the P300 signal. Grand Averaging, PCA, and ICA all fail to improve P300 detection using the Emotiv EPOC to better than random chance (Figures 19 through 21 show this). However, when parallel BLDA and FLDA are used (Figure 22), character detection accuracy for $n = 15$ averaged intensifications improves to 15% for both algorithms (compare with random chance at 2.8%). ANOVA analyses of row/column accuracy results for the EPOC at $n = 15$ averaged intensifications were performed. Results show that $p < 0.01$ when comparing parallel BLDA and FLDA accuracy results to FLDA and BLDA results when Grand Averaging, PCA, and ICA are used as spatial filters.

Another way to confirm that the EPOC is detecting the P300 is to examine the statistical significance of accuracy improvement over increasing values of averaged intensifications. Accuracy should increase with greater numbers of averaged intensifications. An example of this can be observed when the Grass recording system is used to collect data (bottom graph in Figure 22). For all algorithms tested with Grand Averaging, PCA, and ICA as spatial filters, there is no statistically significant increase in accuracy for the EPOC from $n = 2$ averaged intensifications to $n = 15$ ($p > 0.1$). However, when comparing the row/column accuracies in the top graph of Figure 22, the increase in accuracy is statistically significant ($p = 0.0527$). Although the EPOC does not record signals which lead to accuracies as great as those attained by the Grass recording system, the fact that an inexpensive commercial EEG headset can detect a 5 μV potential with any reliability is encouraging.

The signals collected by the Grass recording system consistently result in greater accuracies than the signals collected by the EPOC, which is expected. Figure 19 (Grand Averaging) and Figure 20 (PCA) show that all algorithms improve accuracy over averaged intensifications except PCM. (When ICA is used as a spatial filter, this trend is not visible because all results are near random). PCM works by looking for correlations between specific points summed over epochs, so a possible explanation for this poor performance might be the 2.5 Hz noise. This hypothesis will be tested in Section 4.3. Section 4.3 will also test whether the better performance of parallel FLDA compared to parallel BLDA (Fig-

ure 22, bottom) is due to the 2.5 Hz noise. This performance gap is statistically significant ($p < 0.05$) for $n = 2$ character accuracy, $n = 15$ character accuracy, and $n = 15$ row/column accuracy, but not $n = 2$ row/column accuracy.

4.2.2 Error Rate of Row and Column Eliminating Algorithms

It has been demonstrated in this thesis that both the Emotiv EPOC and the Grass recording setup are capable of capturing EEG which contains the P300 signal. The aim of this section is to address whether it is possible to confidently eliminate row and column flashings based on limited data from a low number of averaged intensifications. To answer this question, FLDA and BLDA classification algorithms were adjusted to choose rows and columns which are the least likely to be targets. See Section 3.3.3 for a more detailed description of how the algorithms work. Results are reported in number of errors. The maximum possible number of errors is 35 characters $\times$ 2 possible errors per epoch = 70. Figure 23 reports results when PCA was used as a spatial filter prior to classification and Figure 24 reports results when parallel methods were used to eliminate rows and columns.

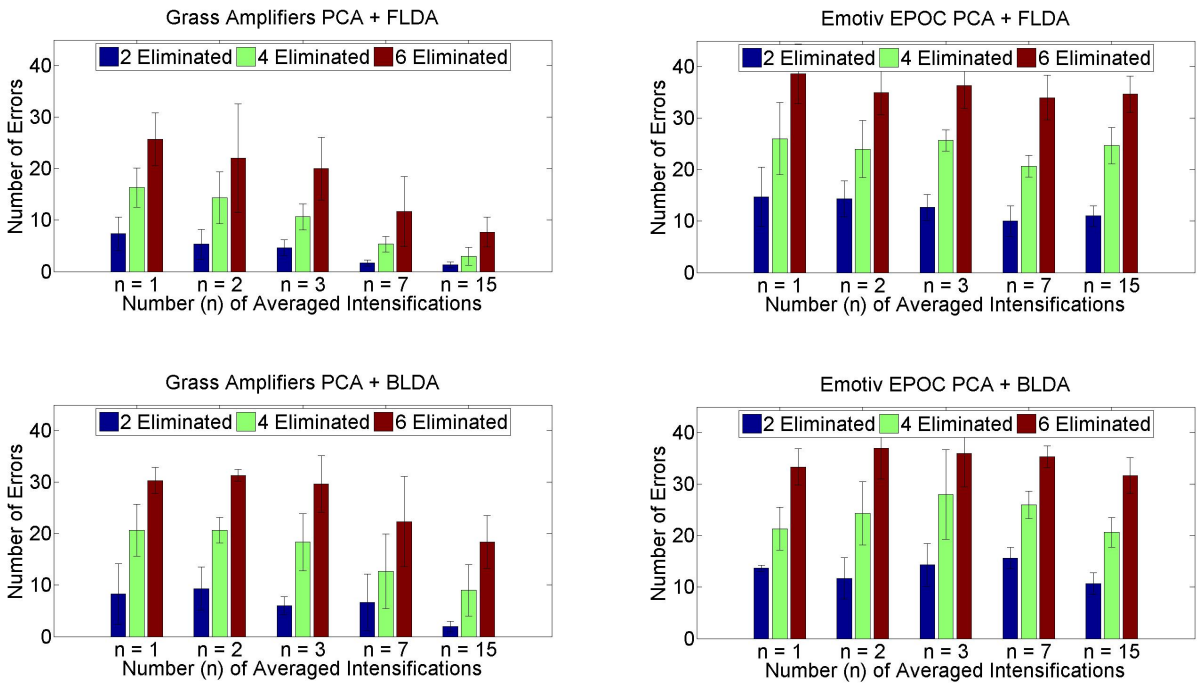

Figure 23: These graphs show the effectiveness of row and column eliminating algorithms when PCA was used as a spatial filter. The x-axis on these plots shows the number of intensification cycles averaged to test the row and column eliminating algorithms. With increasing numbers of intensifications, there should be fewer errors made. The number of rows and columns eliminated is indicated by the colors: blue (2), green (4), and red (6). The maximum possible number of errors is 70. Leave-one-out validation for 35 characters was used to generate these results. Results from three subjects were used to generate these plots. The error bars represent one standard deviation from the mean.

The measure of performance in Figures 23 and 24 differs from what is used in previous sections, classification accuracy. The number of errors made (maximum 70) is a better indicator of performance. This is because if a character is classified wrongly, it is not clear from the character accuracy whether this is due to two errors made or one. In addition to the difference in performance measure, interpretation of these results with respect to random chance also differs. This is because random chance changes based on the number of rows and columns eliminated. For example, if only 2 rows and columns are eliminated, the chance of randomly eliminating non-targets (desirable) is $\frac{10}{12} \times \frac{9}{11} = 0.68182$ or 68.2%. Therefore, the chance of deleting a target row or column randomly (not desirable) is $1 - 0.68182 = 0.31818$ or 31.8%. Since the maximum number of errors that can be made in any given algorithm

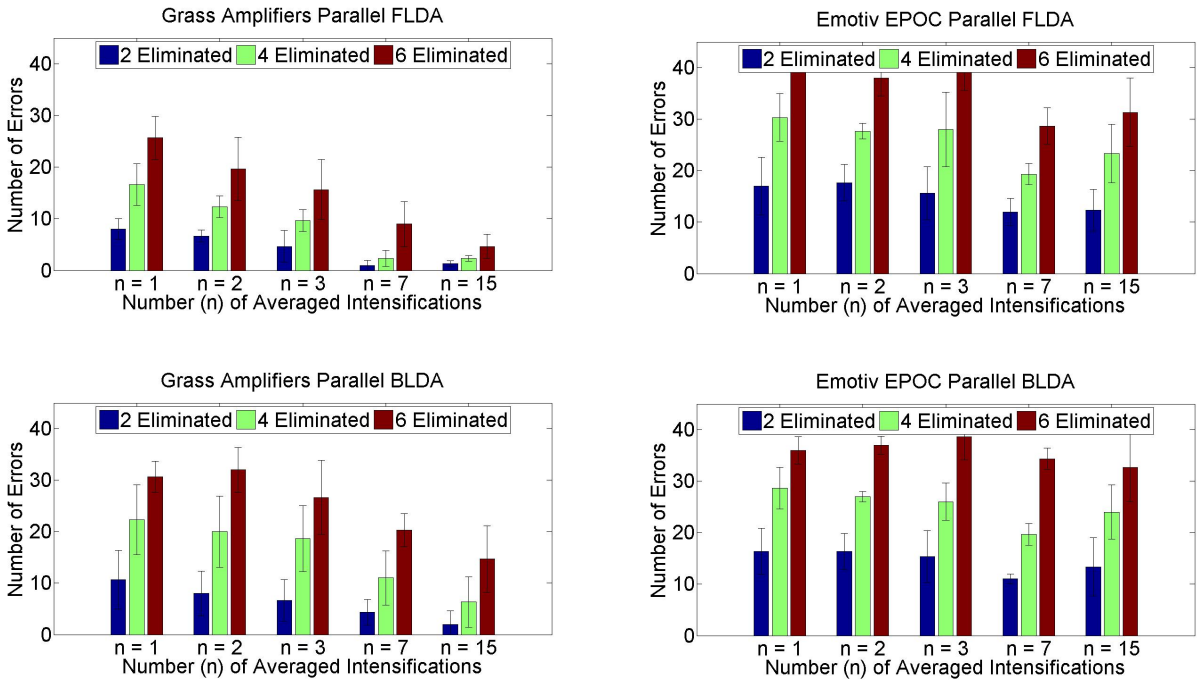

Figure 24: These graphs show the effectiveness of row and column eliminating algorithms when parallel methods (FLDA and BLDA) were used to eliminate rows and columns (three subjects). The number of rows and columns eliminated is indicated by the colors: blue (2), green (4), and red (6). Results from three subjects were used to generate these plots. The error bars represent one standard deviation from the mean.

is 70, the number of errors that this percentage corresponds to is $70 \times 0.31818 = 22.273$, or about 22 errors. When four rows and columns are eliminated (green data series), this value is 40 errors. When six are eliminated (red data series), this value is 54 errors.

Therefore, successfully predicting six rows and columns to eliminate is a more difficult job than predicting four. Additionally, predicting correct rows and columns for elimination becomes a more difficult task when fewer rounds of intensification have been averaged. Figures 23 and 24 reflect this. Fewer errors are made as the number of averaged intensification goes up and the number of eliminated rows and columns goes down. At least, this is the case with the data acquired with the Grass system. Errors made by algorithms on EPOC data, although below random, do not significantly decrease as the number of intensifications increases. Even when parallel FLDA is used, there is not a statistically significant difference ($p < 0.05$) in error number between $n = 1$ and $n = 15$ averaged intensifications. In one

case, between $n = 1$ and $n = 15$ averaged intensifications when parallel FLDA was used to eliminate rows and columns, the p-value equaled 0.0549. Overall, however, the results are not significantly different. This is in contrast to Figure 22 in the previous section. It was found that, in addition to accuracies being better than chance, there was also a statistically significant ($p = 0.0527$) increase in row/column accuracy between $n = 2$ and $n = 15$ averaged rounds of intensification when parallel methods were used for classification.

4.3 Collected Data - Part Two

The results are divided into two parts (4.2 and 4.3) because of a noise component in the EEG recordings from the Grass amplifiers which was discovered after recordings from the first three subjects were collected. This section reports results from five subjects, two of which had also been recorded from in the first phase of data collection. For details on the demographics of the subjects, see Appendix C.

The results for the Grass recording system without the 2.5 Hz noise are better than when the noise is present. To demonstrate this, the mean character accuracy for 15 averaged intensifications was computed across all algorithms for GA, PCA, and parallel voting algorithms for the first and second phase of data collection. The results show that performance of classification algorithms on data collected from the second set of subjects (using the Grass amplifiers) is, on average, two times (100%) better than the accuracy on the first set of subjects. Using the same analysis, there is an average improvement of 15% for the EPOC from phase one to phase two of data collection. The fact that the EPOC improvement percent is not on the same order as the Grass amplifier improvement suggests that some of the accuracy increase is due to noise, not just better performing subjects.

Additional analysis of these results revealed that there is no statistically significant (ANOVA, $p > 0.05$) difference between parallel BLDA and FLDA accuracies on Grass amplifier data for phase two of data collection. These results differ from phase one of data collection, where a statistically significant performance gap could be seen between parallel BLDA and FLDA. This is likely attributable to the 2.5 Hz noise. The accuracy of the BLDA

algorithm, which is similar to FLDA except that it assumes an added Gaussian noise term, was greatly affected by the non-Gaussian 2.5 Hz noise.

The performances of the row and column eliminating algorithms on data collected from the Grass amplifiers in the second phase of data collection were better than the error rates from the first phase. The best performance of any row and column eliminating algorithm was parallel FLDA on $n = 7$ averaged intensifications. On average (across five subjects), only six errors were made if six rows and columns were eliminated after seven averaged rounds of intensifications. Since there are 70 rows and columns in a spelling session 35 characters long, this means that the algorithm, if implemented, would limit the maximum possible accuracy to 64 rows and columns correct out of 70, a row/column accuracy of 91%.

This negative effect is tolerable, however, because of the time savings that result from the elimination. To spell a single character in the BCI dataset takes 31.5 seconds. By eliminating the need to flash 6 rows and columns 8 times, this equates to a time savings of 175 milliseconds $\times$ 6 $\times$ 8 = 8.4 milliseconds per character, or $8.4 \div 31.5 = 27\%$. Increasing the speed of P300 spelling by 27% is a vast improvement over the existing protocol.

4.3.1 Accuracy of Select Spatial Filtering and Classification Algorithm Combinations

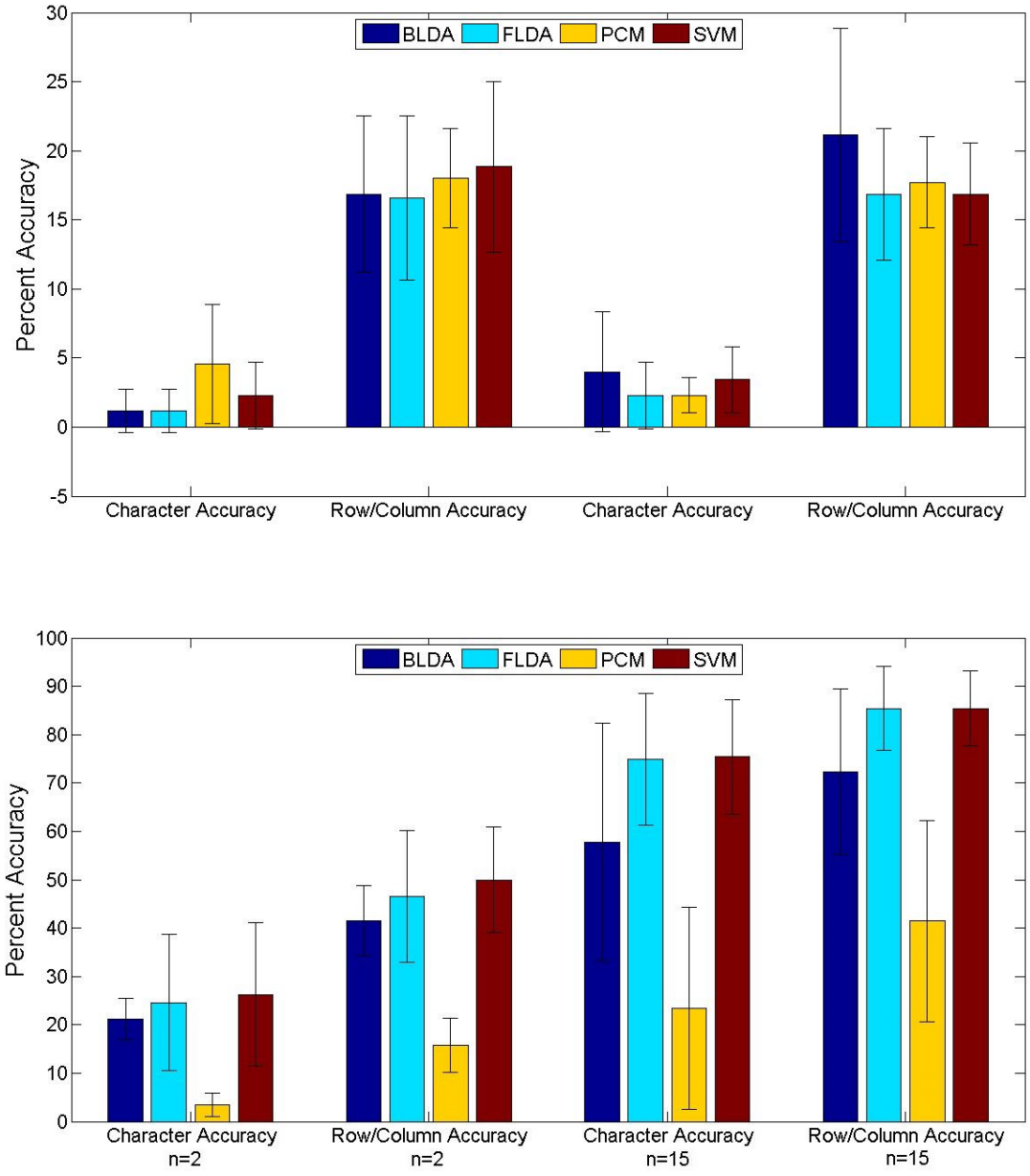

Figure 25: Grand averaging was used as a spatial filter for these data (from five subjects). These graphs show the percent accuracy (character and row/column accuracy) of four classification algorithms when the Emotiv EPOC (top graph) is used to collect data and when the Grass amplifiers (bottom graph) are used to collect data. The x axis labels $n = 2$ and $n = 15$ refer to the averaged number of intensifications used to generate the input data for the classifiers.

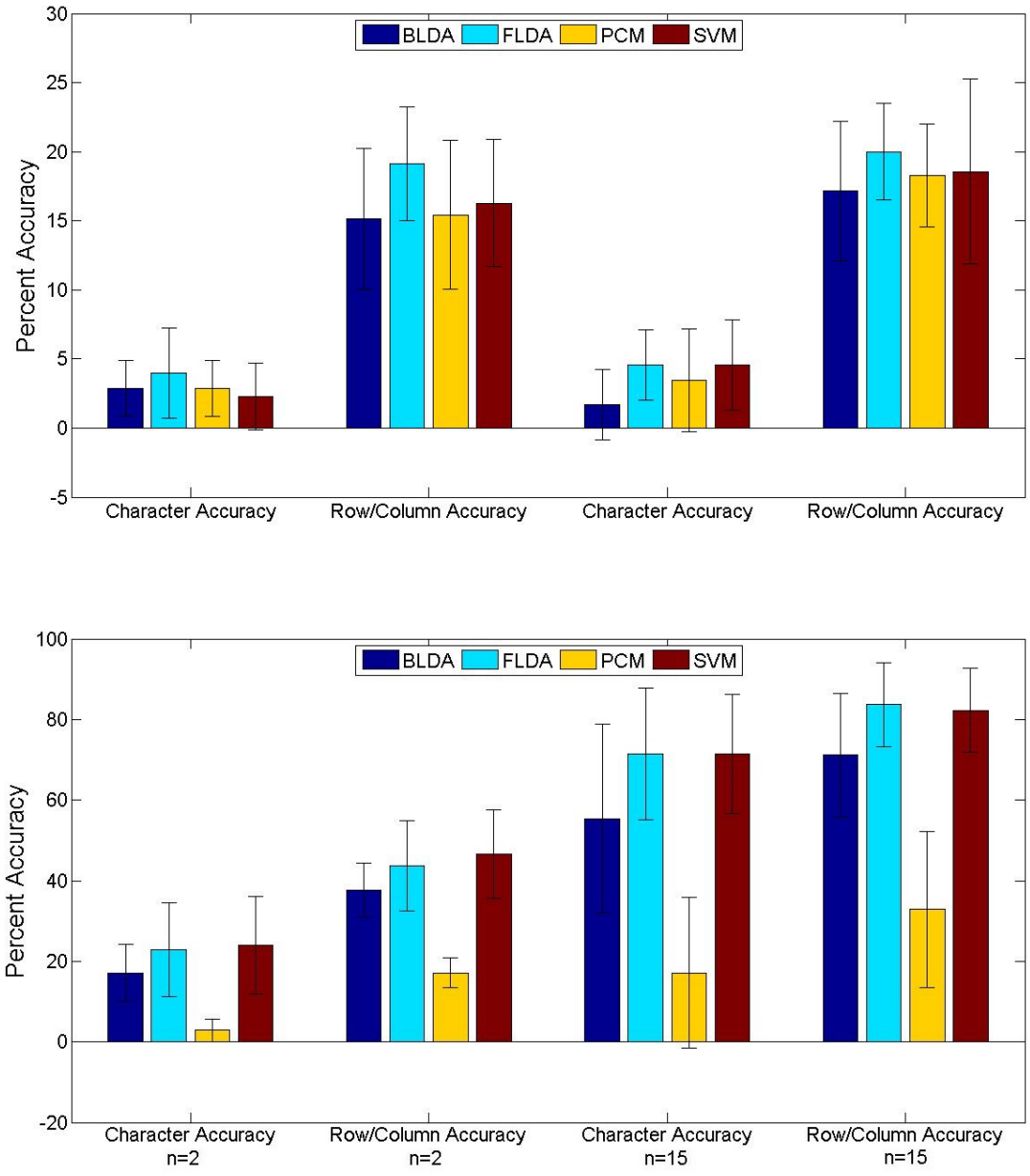

Figure 26: PCA was used as a spatial filter for these data (from five subjects). These graphs show the percent accuracy (character and row/column accuracy) of four classification algorithms when the Emotiv EPOC (top graph) is used to collect data and when the Grass amplifiers (bottom graph) are used to collect data. The x axis labels $n = 2$ and $n = 15$ refer to the averaged number of intensifications used to generate the input data for the classifiers.

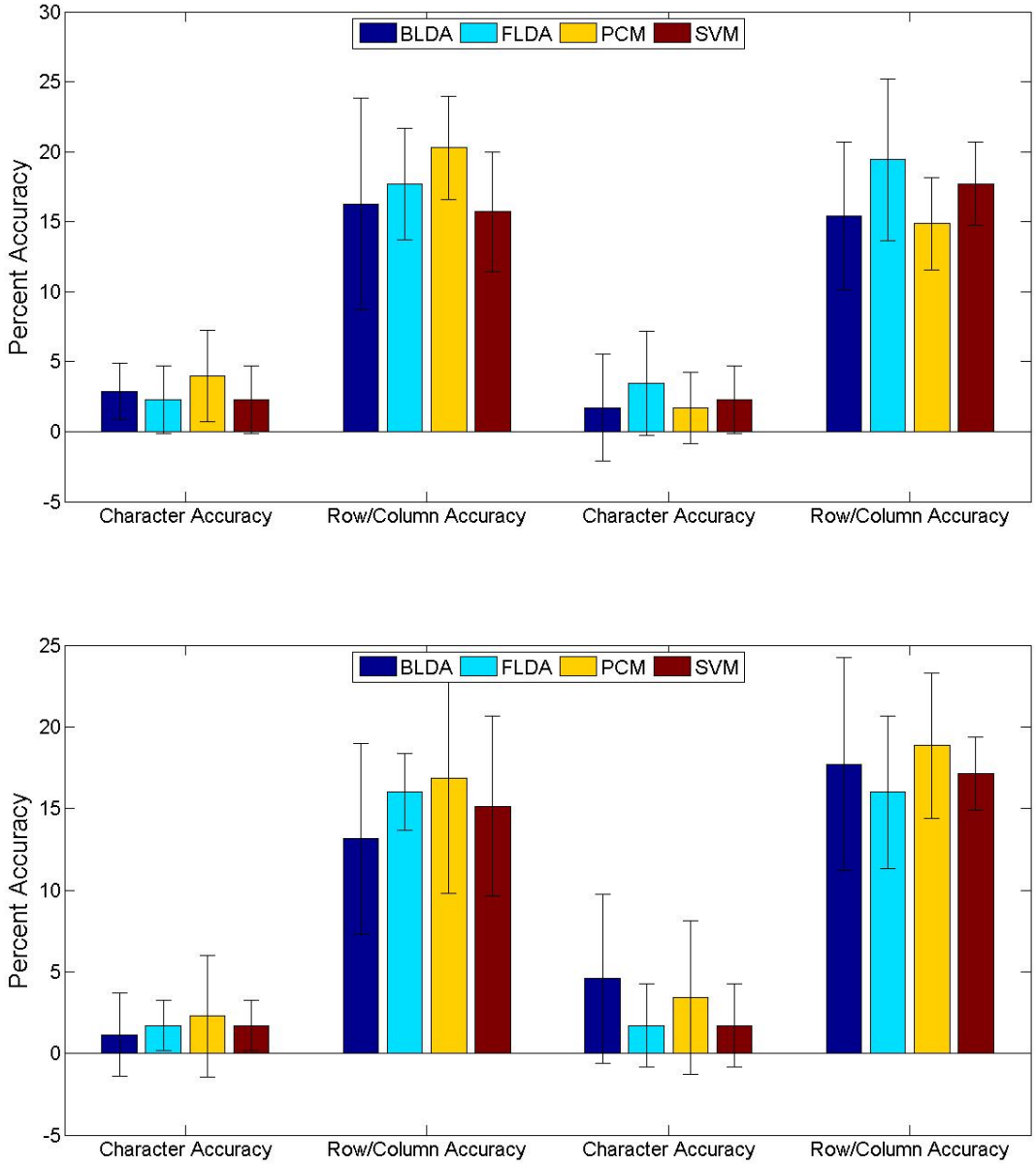

Figure 27: ICA was used as a spatial filter for these data (from five subjects). These graphs show the percent accuracy (character and row/column accuracy) of four classification algorithms when the Emotiv EPOC (top graph) is used to collect data and when the Grass amplifiers (bottom graph) are used to collect data. The x axis labels $n = 2$ and $n = 15$ refer to the averaged number of intensifications used to generate the input data for the classifiers.

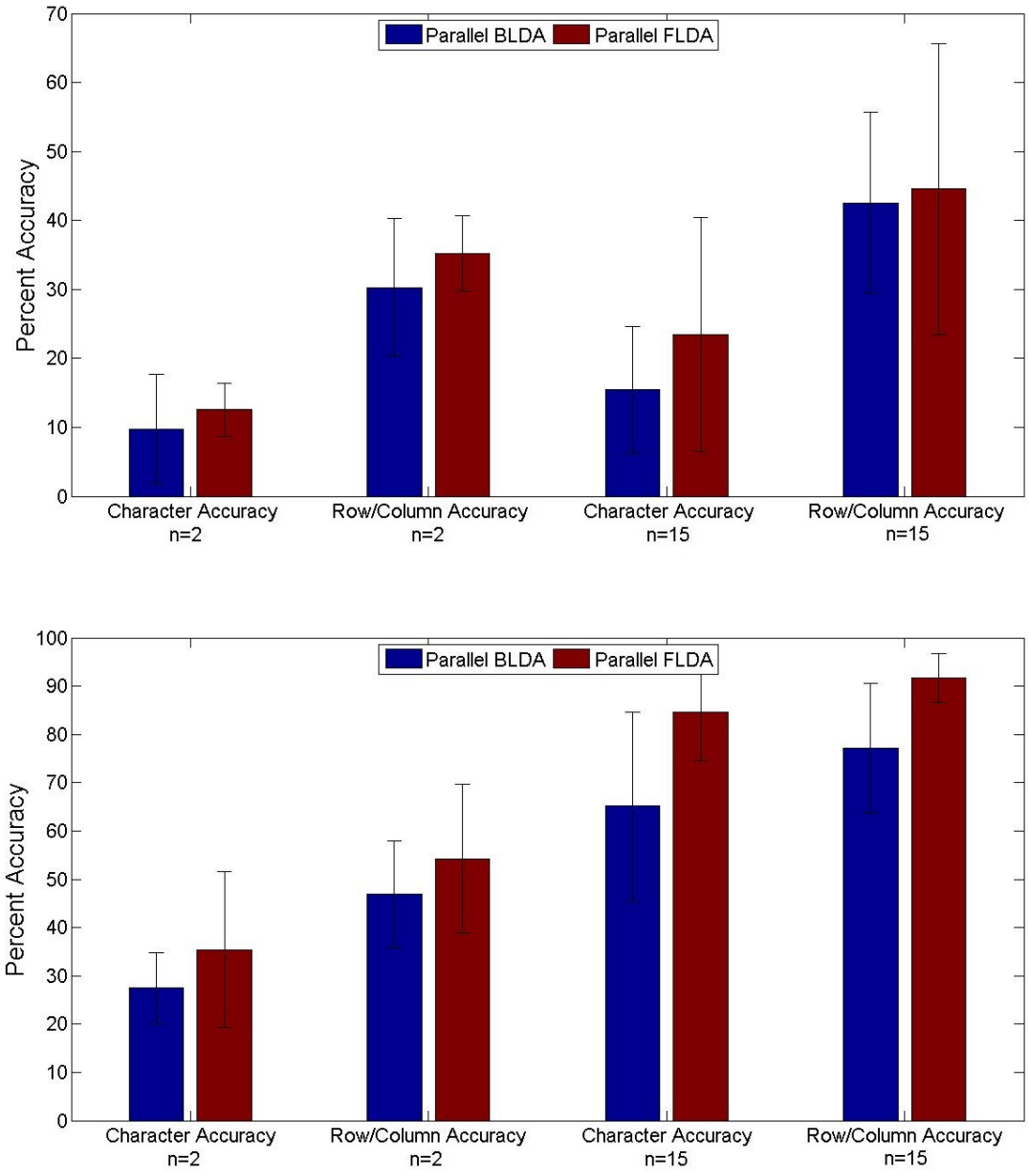

Figure 28: No spatial filter was used on these data (from five subjects). These graphs show the percent accuracy (character and row/column accuracy) of two parallel classification algorithms when the Emotiv EPOC (top graph) is used to collect data and when the Grass amplifiers (bottom graph) are used to collect data. The x axis labels $n = 2$ and $n = 15$ refer to the averaged number of intensifications used to generate the input data for the classifiers.

4.3.2 Error Rate of Row and Column Eliminating Algorithms

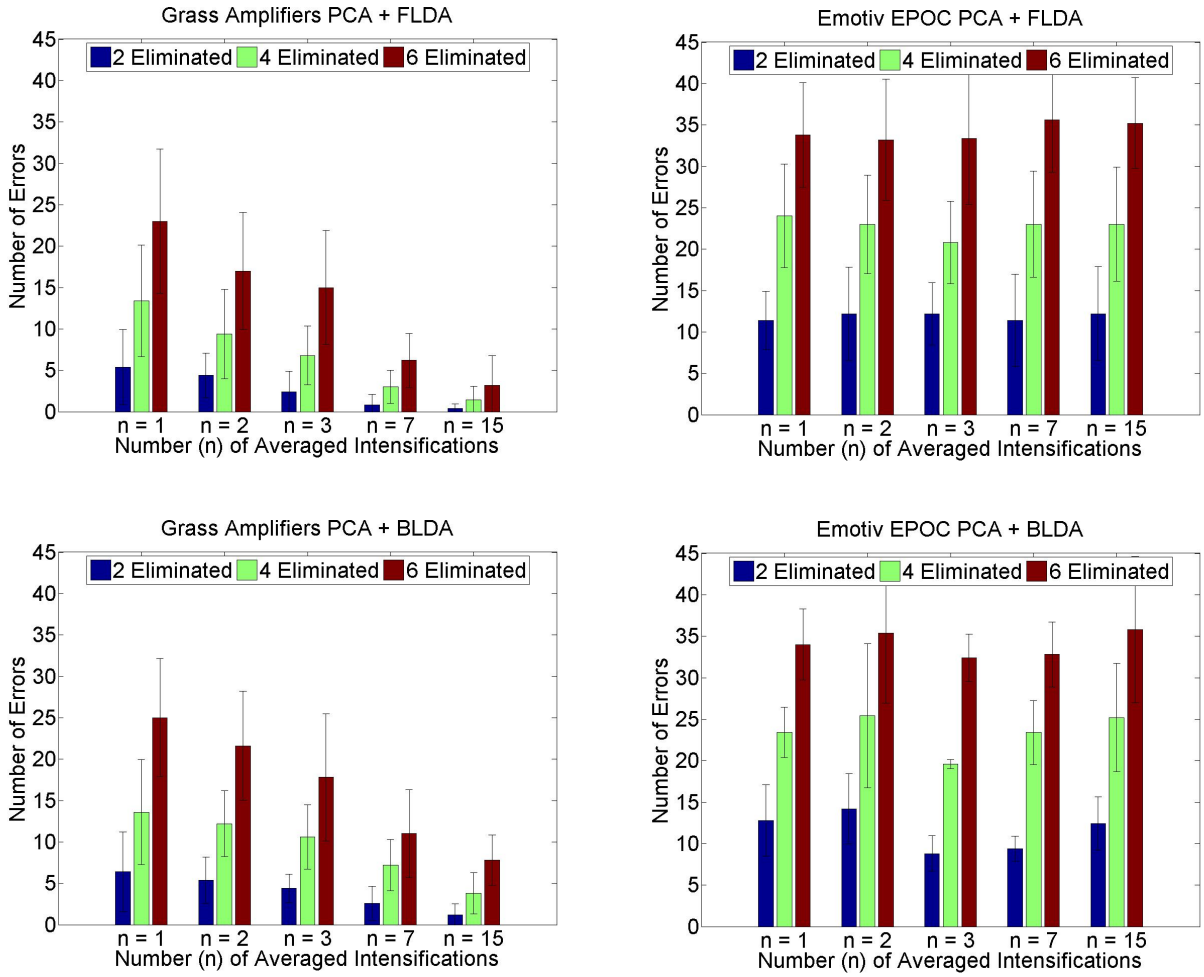

Figure 29: These graphs show the effectiveness of row and column eliminating algorithms when PCA was used as a spatial filter. The x-axis on these plots shows the number of intensification cycles averaged to test the row and column eliminating algorithms. With increasing numbers of intensifications, there should be fewer errors made. The number of rows and columns eliminated is indicated by the colors: blue (2), green (4), and red (6). The maximum possible number of errors is 70. Leave-one-out validation for 35 characters was used to generate these results. Results from five subjects were used to generate these plots. The error bars represent one standard deviation from the mean.

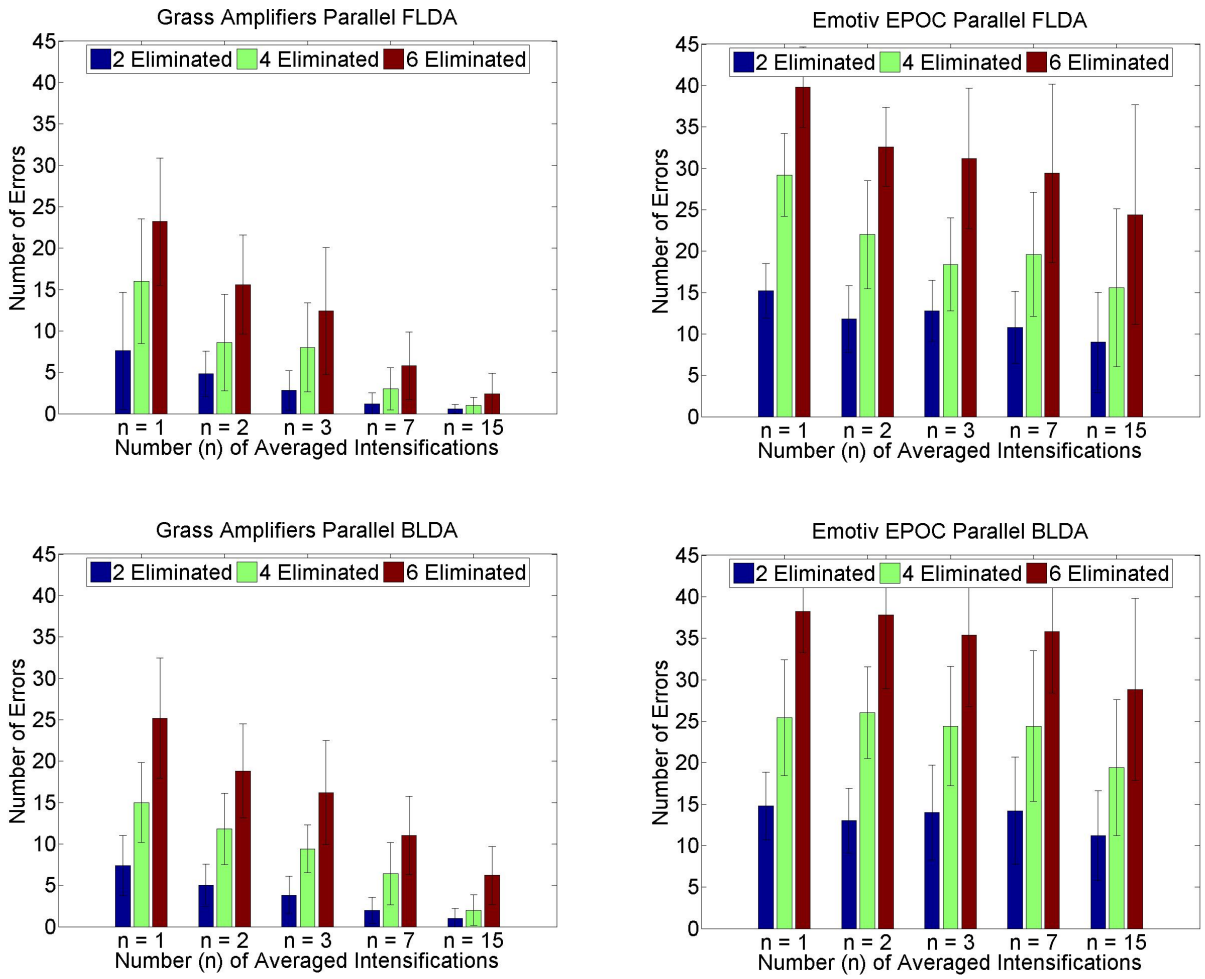

Figure 30: These graphs show the effectiveness of row and column eliminating algorithms when parallel methods (FLDA and BLDA) were used to eliminate rows and columns. The number of rows and columns eliminated is indicated by the colors: blue (2), green (4), and red (6). Results from five subjects were used to generate these plots. The error bars represent one standard deviation from the mean.

5 Conclusions

The potential for EEG based mobile BCIs exists. It is possible to use the well studied Oddball Paradigm in a speller application to elicit the P300 Event Related Potential in the brain, as is shown in this thesis and in prior work. Additionally, it is possible to utilize an inexpensive, commercially available EEG headset to detect this ERP and exploit it in a speller application. Training the parallel FLDA algorithm on 408 examples resulted in a 23% character accuracy on data acquired from the Emotiv EPOC (Section 4.3). This accuracy is a significant improvement over random chance (2.8%).

A unique feature of this work is the direct comparison between the EPOC and the Grass Neurodata Acquisition System. This is accomplished via parallel EEG collection (Section 3.2). In comparison to the EPOC's performance of 23% character accuracy, the Grass Acquisition System is able to acquire data which yields 85% character accuracy (Parallel FLDA, Figure 28). Future work should investigate the possibility of using all 14 electrodes on the EPOC to improve character accuracy further, although accuracies do not necessarily improve with the use of more electrodes (Figure 16). The best approach to this problem is likely to be one which recursively eliminates 'noisy' channels from analysis. The most important electrode recording positions are known to vary from one patient to another. Therefore, as the winners of the 2003 BCI Competition show (Section 4.1.1), an approach where the number and location of electrodes are fixed is not best.

Half of the engineering challenge in BCI research is accuracy. The other half of the problem is speed of detection and classification. Row and column eliminating algorithms are designed to address this problem and improve the speed of ERP classification. The principle behind these algorithms is to successfully determine which rows and columns are not likely to be targets after a low number of intensification cycles have been averaged. Preventing these 'non-targets' from flashing in future intensification cycles decreases the amount of recording time necessary to determine a character's identity. There is a balance between how many intensifications are necessary to eliminate rows and columns without greatly jeopardizing character detection accuracy. For example, averaging $n = 7$ intensification

cycles will give the algorithm better signals from which non-targets can be eliminated than $n = 2$ intensification cycles would. Figures 23 and 24 clearly show this. The downside of averaging more cycles is that it takes much more time to collect them.

In addition to this consideration of averaged intensification rounds and their effect on data quality, there is also the question of how many rows and columns can be eliminated without greatly compromising accuracy. Clearly, eliminating more rows and columns from future intensifications increases the probability that a target row or column will be eliminated by accident. The best balance between time savings and error rate was found to be parallel FLDA for $n = 7$ averaged intensifications and 6 eliminated rows and columns (see Figure 30). Only 6 errors were made on average, corresponding to a maximum decrease in row/column accuracy of about 9%. The corresponding time savings is about 27%. Increasing the speed of P300 spelling by 27% is a vast improvement over the existing protocol. Future work should investigate implementing row and column eliminating algorithms in a closed-loop P300 speller for real-time evaluation of character spelling accuracy.

While time savings is critical, accuracy is also paramount in BCI applications. Disabled patients rely on the functionality of the prosthesis. If the device is unusable because it does not accurately classify brain signals, then speed means nothing and the device is useless. So there is a trade-off between time savings and accuracy in BCIs. Future work should investigate ways to decrease errors made by row and column eliminating algorithms. One potential way to do this is to train the algorithms with massive numbers of examples (in this work algorithms were trained with 408 examples). As a general rule, the more data a machine learning algorithm is given, the better it is able to make classifications.

In conclusion, for an EEG based mobile BCI to be useful, it must be inexpensive and portable. The Emotiv EPOC and devices like it are certainly helping to meet this need. The BCI must also have accuracy and speed, which can be improved by better signal quality, more training examples, and algorithms which save time on data collection. Better solutions to this problem will come in three ways: commercial EEG technology improvement, increasing algorithm accuracy, and faster classification speed.

References

[1] M. Duvinage, T. Castermans, T. Dutoit, M. Petieau, T. Hoellinger, C. D. Saedeleer, K. Seetharaman, and G. Cheron, "A P300-based Quantitative Comparison between the Emotiv Epoc Headset and a Medical EEG Device," *Biomedical Engineering / 765: Telehealth / 766: Assistive Technologies*, 2012.

[2] A. T. Campbell, T. Choudhury, S. Hu, H. Lu, M. Rabbi, R. D. S. Raizada, and M. K. Mukerjee, "NeuroPhone : Brain-Mobile Phone Interface using a Wireless EEG Headset Categories and Subject Descriptors," *Design*, vol. 28, no. 5, pp. 3–8, 2010.

[3] J. Ramírez-Cortes, "P-300 rhythm detection using ANFIS algorithm and wavelet feature extraction in EEG signals," *Proceedings of the ...*, vol. I, 2010.

[4] "Emotiv Systems." `http://www.emotiv.com`. Accessed: 2013-09-09.

[5] U. Hoffmann and G. Garcia, "A boosting approach to P300 detection with application to brain-computer interfaces," *Neural Engineering, ...*, 2005.

[6] Z. Iscan, "Detection of P300 wave from EEG data for brain-computer interface applications," *Pattern Recognition and Image Analysis*, vol. 21, pp. 481–485, Sept. 2011.

[7] H. Cecotti and A. Graser, "Convolutional neural networks for P300 detection with application to brain-computer interfaces," *Pattern Analysis and Machine Intelligence ...*, vol. 33, no. 3, pp. 433–445, 2011.

[8] C. Guger, S. Daban, E. Sellers, C. Holzner, G. Krausz, R. Carabalona, F. Gramatica, and G. Edlinger, "How many people are able to control a p300-based brain-computer interface (bci)?," *Neuroscience Letters*, vol. 462, no. 1, pp. 94 – 98, 2009.

[9] J. R. Wolpaw, G. Editor, N. Birbaumer, W. J. Heetderks, D. J. Mcfarland, P. H. Peckham, G. Schalk, E. Donchin, L. A. Quatrano, C. J. Robinson, T. M. Vaughan, and G. Editor, "Brain - Computer Interface Technology : A Review of the First International Meeting," vol. 8, no. 2, pp. 164–173, 2000.

[10] "University of Texas Neuroscience Textbook." `http://neuroscience.uth.tmc.edu/s1/introduction.html`. Accessed: 2013-09-09.

[11] M. A. Lebedev and M. A. L. Nicolelis, "Brain - machine interfaces : past , present and future," vol. 29, no. 9, 2006.

[12] J. R. Wolpaw, N. Birbaumer, D. J. Mcfarland, G. Pfurtscheller, and T. M. Vaughan, "Brain - computer interfaces for communication and control," vol. 113, pp. 767–791, 2002.

[13] G. Schalk, D. J. Mcfarland, T. Hinterberger, N. Birbaumer, J. R. Wolpaw, and A. B.-c. I. B. C. I. Technology, "BCI2000 : A General-Purpose Brain-Computer Interface (BCI) System," vol. 51, no. 6, pp. 1034–1043, 2004.

[14] T. Budzynski, *Introduction to quantitative EEG and neurofeedback : advanced theory and applications*. Amsterdam: Academic Press/Elsevier, 2009.

[15] J. Polich, "Updating P300: an integrative theory of P3a and P3b.," *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology*, vol. 118, pp. 2128–48, Oct. 2007.

[16] J. Polich and A. Kok, "Cognitive and biological determinants of P300: an integrative review.," *Biological psychology*, vol. 41, pp. 103–46, Oct. 1995.

[17] A. Kok, "On the utility of P3 amplitude as a measure of processing capacity.," *Psychophysiology*, vol. 38, pp. 557–77, May 2001.

[18] B. H. Jansen, A. Allam, P. Kota, K. Lachance, A. Osho, and K. Sundaresan, "An exploratory study of factors affecting single trial P300 detection.," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 975–978, 2004.

[19] Y. Liu, Z. Zhou, D. Hut, and G. Dong, "T-weighted Approach for Neural Information Processing in P300 based Brain-Computer Interface," pp. 1535–1539.

[20] P. Meinicke and M. Kaper, "Improving transfer rates in brain computer interfacing: a case study," *Advances in ...*, 2002.

[21] B. Blankertz, K.-r. Müller, G. Curio, T. M. Vaughan, G. Schalk, R. Jonathan, A. Schlögl, C. Neuper, G. Pfurtscheller, T. Hinterberger, M. Schröder, and N. Birbaumer, "The BCI Competition 2003 :," vol. XX, pp. 100–106, 2004.

[22] M. Teplan, "Fundamentals of eeg measurement," *Measurement science review*, vol. 2, no. 2, pp. 1–11, 2002.

[23] "NR Sign EEG Accessories." `http://www.nrsign.com/accessories/eeg-wire`. Accessed: 2013-09-17.

[24] "Indiana University Psychology Department." `http://cognitrn.psych.indiana.edu/busey/eegseminar/pdfs/EEGPrimerCh2.pdf`. Accessed: 2013-09-17.

[25] M. van Vliet, A. Robben, N. Chumerin, N. V. Manyakov, A. Combaz, and M. M. Van Hulle, "Designing a brain-computer interface controlled video-game using consumer grade EEG hardware," *2012 ISSNIP Biosignals and Biorobotics Conference: Biosignals and Robotics for Better and Safer Living (BRC)*, pp. 1–6, Jan. 2012.

[26] N. Manyakov and N. Chumerin, "On the selection of time interval and frequency range of EEG Signal Preprocessing for P300 Brain-Computer Interfacing," *... Conference on Medical ...*, pp. 8–11, 2010.

[27] L. Bougrain, "Finally, what is the best filter for P300 detection?," *TOBI Workshop lll-Tools ...*, vol. 2012, no. September 2011, pp. 2–3, 2012.

[28] N. V. Manyakov, N. Chumerin, A. Combaz, and M. M. V. Hulle, "P300 BRAIN-COMPUTER INTERFACE ON DISABLED SUBJECTS," 2008.

[29] U. Hoffmann, J.-M. Vesin, T. Ebrahimi, and K. Diserens, " brain-computer interface for disabled subjects.," *Journal of neuroscience methods*, vol. 167, pp. 115–25, Jan. 2008.

[30] D. J. Krusienski, E. W. Sellers, F. Cabestaing, S. Bayoudh, D. J. McFarland, T. M. Vaughan, and J. R. Wolpaw, "A comparison of classification techniques for the P300 Speller.," *Journal of neural engineering*, vol. 3, pp. 299–305, Dec. 2006.

[31] H. Cecotti, B. Rivet, M. Congedo, C. Jutten, O. Bertrand, E. Maby, and J. Mattout, "A robust sensor-selection method for P300 brain-computer interfaces.," *Journal of neural engineering*, vol. 8, p. 016001, Feb. 2011.

[32] C. Bugli and P. Lambert, "Comparison between Principal Component Analysis and Independent Component Analysis in Electroencephalograms Modelling," *Biometrical Journal*, vol. 49, pp. 312–327, Apr. 2007.

[33] N. Xu, X. Gao, B. Hong, X. Miao, S. Gao, and F. Yang, "BCI Competition 2003–Data set IIb: enhancing P300 wave detection using ICA-based subspace projections for BCI applications.," *IEEE transactions on bio-medical engineering*, vol. 51, pp. 1067–72, June 2004.

[34] O. I. Khan, S.-H. K. S.-H. Kim, T. Rasheed, a. Khan, and T.-S. K. T.-S. Kim, "Extraction of P300 using constrained independent component analysis.," *Conference Proceedings of the International Conference of IEEE Engineering in Medicine and Biology Society*, vol. 2009, pp. 4735–4738, 2009.

[35] S. Lee, Y. Nam, and S. Choi, "Self-labeling for P300 detection," *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 268–273, Oct. 2012.

[36] B. Rivet, A. Souloumiac, V. Attina, and G. Gibert, "xDAWN algorithm to enhance evoked potentials: application to brain-computer interface.," *IEEE transactions on bio-medical engineering*, vol. 56, pp. 2035–43, Aug. 2009.

[37] S. Vardeman, *Basic engineering data collection and analysis.* Pacific Grove, CA: Brooks/Cole, 2001.

[38] C. Bishop, *Pattern recognition and machine learning.* New York: Springer, 2006.

[39] a. Combaz, N. Chumerin, N. Manyakov, a. Robben, J. Suykens, and M. Van Hulle, "Towards the detection of error-related potentials and its integration in the context of a P300 speller brain-computer interface," *Neurocomputing*, vol. 80, pp. 73–82, Mar. 2012.

[40] H. Zhang, C. Guan, and C. Wang, "Asynchronous P300-Based Brain â Computer Interfaces : A Computational Approach," vol. 55, no. 6, pp. 1754–1763, 2008.

[41] Y. Renard, F. Lotte, G. Gibert, M. Congedo, E. Maby, V. Delannoy, O. Bertrand, and A. Lécuyer, "Openvibe: an open-source software platform to design, test, and use brain-computer interfaces in real and virtual environments," *Presence: teleoperators and virtual environments*, vol. 19, no. 1, pp. 35–53, 2010.

[42] A. Rakotomamonjy and V. Guigue, "Bci competition iii: Dataset ii- ensemble of svms for bci p300 speller," *Biomedical Engineering, IEEE Transactions on*, vol. 55, pp. 1147–1154, March 2008.

Appendix A: Phase I Raw Data

Table 6. Percent accuracy values for Grass Amplifiers (Three Subjects).

Algorithm	$n = 2$ Character Accuracy	$n = 2$ Row and Column Accuracy	$n = 15$ Character Accuracy	$n = 15$ Row and Column Accuracy
GA and BLDA	3.81 ± 3.30	21.90 ± 4.36	16.19 ± 7.19	40.95 ± 8.61
GA and FLDA	14.29 ± 12.45	33.81 ± 12.32	34.29 ± 24.91	58.10 ± 17.75
GA and PCM	0.00 ± 0.00	14.76 ± 6.75	1.90 ± 1.65	15.24 ± 5.77
GA and SVM	10.48 ± 10.82	32.38 ± 10.91	40.95 ± 21.63	61.43 ± 16.48
PCA and BLDA	5.71 ± 2.86	21.90 ± 2.18	16.19 ± 9.18	39.05 ± 8.12
PCA and FLDA	13.33 ± 15.74	32.38 ± 13.27	37.14 ± 20.00	60.48 ± 14.87
PCA and PCM	0.00 ± 0.00	14.76 ± 5.77	1.90 ± 1.65	15.71 ± 4.95
PCA and SVM	9.52 ± 14.09	30.48 ± 14.38	42.86 ± 17.38	61.90 ± 13.50
ICA and BLDA	0.95 ± 1.65	13.81 ± 4.36	4.76 ± 3.30	23.33 ± 5.02
ICA and FLDA	1.90 ± 1.65	15.71 ± 6.23	2.86 ± 4.95	17.62 ± 9.29
ICA and PCM	3.81 ± 1.65	18.10 ± 2.18	8.57 ± 0.00	23.33 ± 2.97
ICA and SVM	3.81 ± 4.36	15.24 ± 4.12	2.86 ± 2.86	19.05 ± 8.12
Parallel BLDA	6.67 ± 4.36	26.67 ± 5.02	28.57 ± 14.29	54.76 ± 11.55
Parallel FLDA	27.62 ± 13.20	44.29 ± 13.78	69.52 ± 18.37	82.38 ± 10.82

Table 7. Percent accuracy values for Emotiv EPOC (Three Subjects).

Algorithm	$n = 2$ Character Accuracy	$n = 2$ Row and Column Accuracy	$n = 15$ Character Accuracy	$n = 15$ Row and Column Accuracy
GA and BLDA	1.90 ± 1.65	15.71 ± 1.43	1.90 ± 3.30	17.62 ± 7.19
GA and FLDA	0.95 ± 1.65	12.38 ± 4.36	2.86 ± 2.86	17.14 ± 7.56
GA and PCM	2.86 ± 2.86	12.86 ± 7.95	1.90 ± 1.65	18.57 ± 6.23
GA and SVM	1.90 ± 1.65	15.24 ± 5.41	2.86 ± 2.86	17.14 ± 3.78
PCA and BLDA	2.86 ± 2.86	17.14 ± 5.71	3.81 ± 1.65	20.48 ± 5.02
PCA and FLDA	0.95 ± 1.65	15.24 ± 2.18	3.81 ± 3.30	17.62 ± 5.02
PCA and PCM	2.86 ± 2.86	14.76 ± 4.12	4.76 ± 1.65	18.10 ± 1.65
PCA and SVM	0.95 ± 1.65	15.24 ± 2.18	4.76 ± 5.95	16.19 ± 11.10
ICA and BLDA	3.81 ± 1.65	15.71 ± 2.86	0.95 ± 1.65	17.14 ± 5.71
ICA and FLDA	0.00 ± 0.00	13.33 ± 3.30	2.86 ± 2.86	16.19 ± 7.87
ICA and PCM	1.90 ± 3.30	15.71 ± 1.43	6.67 ± 1.65	18.10 ± 2.18
ICA and SVM	0.00 ± 0.00	10.48 ± 3.30	0.00 ± 0.00	13.81 ± 6.75
Parallel BLDA	9.52 ± 4.36	28.57 ± 6.55	13.33 ± 3.30	35.71 ± 5.71
Parallel FLDA	8.57 ± 2.86	29.05 ± 6.60	15.24 ± 9.18	39.05 ± 10.03

Table 8. Number of errors for Grass Amplifiers (Three Subjects).

Algorithm	Number of Averaged Intensifications	2 Eliminated	4 Eliminated	6 Eliminated
PCA and BLDA	$n = 1$	8.33 ± 5.86	20.67 ± 5.03	30.33 ± 2.52
	$n = 2$	9.33 ± 4.16	20.67 ± 2.52	31.33 ± 1.15
	$n = 3$	6.00 ± 1.73	18.33 ± 5.51	29.67 ± 5.51
	$n = 7$	6.67 ± 5.51	12.67 ± 7.23	22.33 ± 8.74
	$n = 15$	2.00 ± 1.00	9.00 ± 5.00	18.33 ± 5.13
PCA and FLDA	$n = 1$	7.33 ± 3.21	16.33 ± 3.79	25.67 ± 5.13
	$n = 2$	5.33 ± 2.89	14.33 ± 5.03	22.00 ± 10.54
	$n = 3$	4.67 ± 1.53	10.67 ± 2.52	20.00 ± 6.08
	$n = 7$	1.67 ± 0.58	5.33 ± 1.53	11.67 ± 6.81
	$n = 15$	1.33 ± 0.58	3.00 ± 1.73	7.67 ± 2.89
Parallel BLDA	$n = 1$	10.67 ± 5.69	22.33 ± 6.81	30.67 ± 3.06
	$n = 2$	8.00 ± 4.36	20.00 ± 6.93	32.00 ± 4.36
	$n = 3$	6.67 ± 4.04	18.67 ± 6.43	26.67 ± 7.23
	$n = 7$	4.33 ± 2.52	11.00 ± 5.29	20.33 ± 3.21
	$n = 15$	2.00 ± 2.65	6.33 ± 4.93	14.67 ± 6.43
Parallel FLDA	$n = 1$	8.00 ± 2.00	16.67 ± 4.04	25.67 ± 4.16
	$n = 2$	6.67 ± 1.15	12.33 ± 2.08	19.67 ± 6.11
	$n = 3$	4.67 ± 3.06	9.67 ± 2.08	15.67 ± 5.86
	$n = 7$	1.00 ± 1.00	2.33 ± 1.53	9.00 ± 4.36
	$n = 15$	1.33 ± 0.58	2.33 ± 0.58	4.67 ± 2.31

Table 9. Number of errors for the Emotiv EPOC (Three Subjects).

Algorithm	Number of Averaged Intensifications	2 Eliminated	4 Eliminated	6 Eliminated
PCA and BLDA	$n = 1$	13.67 ± 0.58	21.33 ± 4.16	33.33 ± 3.51
	$n = 2$	11.67 ± 4.04	24.33 ± 6.11	37.00 ± 6.00
	$n = 3$	14.33 ± 4.16	28.00 ± 8.72	36.00 ± 6.56
	$n = 7$	15.67 ± 2.08	26.00 ± 2.65	35.33 ± 2.08
	$n = 15$	10.67 ± 2.08	20.67 ± 2.89	31.67 ± 3.51
PCA and FLDA	$n = 1$	14.67 ± 5.77	26.00 ± 7.00	38.67 ± 5.77
	$n = 2$	14.33 ± 3.51	24.00 ± 5.57	35.00 ± 4.36
	$n = 3$	12.67 ± 2.52	25.67 ± 2.08	36.33 ± 4.51
	$n = 7$	10.00 ± 3.00	20.67 ± 2.08	34.00 ± 4.36
	$n = 15$	11.00 ± 2.00	24.67 ± 3.51	34.67 ± 3.51
Parallel BLDA	$n = 1$	16.33 ± 4.51	28.67 ± 4.04	36.00 ± 2.65
	$n = 2$	16.33 ± 3.51	27.00 ± 1.00	37.00 ± 1.73
	$n = 3$	15.33 ± 5.03	26.00 ± 3.61	38.67 ± 4.51
	$n = 7$	11.00 ± 1.00	19.67 ± 2.08	34.33 ± 2.08
	$n = 15$	13.33 ± 5.69	24.00 ± 5.29	32.67 ± 6.66
Parallel FLDA	$n = 1$	17.00 ± 5.57	30.33 ± 4.62	42.00 ± 1.73
	$n = 2$	17.67 ± 3.51	27.67 ± 1.53	38.00 ± 3.46
	$n = 3$	15.67 ± 5.13	28.00 ± 7.21	39.67 ± 4.04
	$n = 7$	12.00 ± 2.65	19.33 ± 2.08	28.67 ± 3.51
	$n = 15$	12.33 ± 4.04	23.33 ± 5.69	31.33 ± 6.66

Appendix B: Phase II Raw Data

Table 10. Percent accuracy values for Grass Amplifiers (Five Subjects).

Algorithm	$n = 2$ Character Accuracy	$n = 2$ Row and Column Accuracy	$n = 15$ Character Accuracy	$n = 15$ Row and Column Accuracy
GA and BLDA	21.14 ± 4.33	41.43 ± 7.28	57.71 ± 24.61	72.29 ± 17.13
GA and FLDA	24.57 ± 14.08	46.57 ± 13.58	74.86 ± 13.61	85.43 ± 8.65
GA and PCM	3.43 ± 2.39	15.71 ± 5.62	23.43 ± 20.84	41.43 ± 20.87
GA and SVM	26.29 ± 14.76	50.00 ± 10.83	75.43 ± 11.88	85.43 ± 7.79
PCA and BLDA	17.14 ± 7.00	37.71 ± 6.75	55.43 ± 23.44	71.14 ± 15.27
PCA and FLDA	22.86 ± 11.61	43.71 ± 11.28	71.43 ± 16.41	83.71 ± 10.43
PCA and PCM	2.86 ± 2.86	17.14 ± 3.64	17.14 ± 18.63	32.86 ± 19.30
PCA and SVM	24.00 ± 12.05	46.57 ± 10.95	71.43 ± 14.71	82.29 ± 10.48
ICA and BLDA	1.14 ± 2.56	13.14 ± 5.84	4.57 ± 5.19	17.71 ± 6.52
ICA and FLDA	1.71 ± 1.56	16.00 ± 2.35	1.71 ± 2.56	16.00 ± 4.67
ICA and PCM	2.29 ± 3.73	16.86 ± 7.03	3.43 ± 4.69	18.86 ± 4.45
ICA and SVM	1.71 ± 1.56	15.14 ± 5.50	1.71 ± 2.56	17.14 ± 2.26
Parallel BLDA	27.43 ± 7.45	46.86 ± 11.13	65.14 ± 19.52	77.14 ± 13.40
Parallel FLDA	35.43 ± 16.11	54.29 ± 15.42	84.57 ± 10.02	91.71 ± 5.09

Table 11. Percent accuracy values for Emotiv EPOC (Five Subjects).

Algorithm	$n = 2$ Character Accuracy	$n = 2$ Row and Column Accuracy	$n = 15$ Character Accuracy	$n = 15$ Row and Column Accuracy
GA and BLDA	1.14 ± 1.56	16.86 ± 5.66	4.00 ± 4.33	21.14 ± 7.72
GA and FLDA	1.14 ± 1.56	16.57 ± 5.94	2.29 ± 2.39	16.86 ± 4.78
GA and PCM	4.57 ± 4.33	18.00 ± 3.59	2.29 ± 1.28	17.71 ± 3.29
GA and SVM	2.29 ± 2.39	18.86 ± 6.18	3.43 ± 2.39	16.86 ± 3.70
PCA and BLDA	2.86 ± 2.02	15.14 ± 5.11	1.71 ± 2.56	17.14 ± 5.05
PCA and FLDA	4.00 ± 3.26	19.14 ± 4.12	4.57 ± 2.56	20.00 ± 3.50
PCA and PCM	2.86 ± 2.02	15.43 ± 5.38	3.43 ± 3.73	18.29 ± 3.70
PCA and SVM	2.29 ± 2.39	16.29 ± 4.58	4.57 ± 3.26	18.57 ± 6.70
ICA and BLDA	2.86 ± 2.02	16.29 ± 7.53	1.71 ± 3.83	15.43 ± 5.29
ICA and FLDA	2.29 ± 2.39	17.71 ± 3.99	3.43 ± 3.73	19.43 ± 5.77
ICA and PCM	4.00 ± 3.26	20.29 ± 3.70	1.71 ± 2.56	14.86 ± 3.29
ICA and SVM	2.29 ± 2.39	15.71 ± 4.29	2.29 ± 2.39	17.71 ± 2.96
Parallel BLDA	9.71 ± 7.98	30.29 ± 9.97	15.43 ± 9.17	42.57 ± 13.15
Parallel FLDA	12.57 ± 3.83	35.14 ± 5.50	23.43 ± 16.95	44.57 ± 21.08

Table 12. Number of errors for Grass Amplifiers (Five Subjects).

Algorithm	Number of Averaged Intensifications	2 Eliminated	4 Eliminated	6 Eliminated
PCA and BLDA	$n = 1$	6.40 ± 4.83	13.60 ± 6.35	25.00 ± 7.14
	$n = 2$	5.40 ± 2.79	12.20 ± 3.96	21.60 ± 6.58
	$n = 3$	4.40 ± 1.67	10.60 ± 3.91	17.80 ± 7.66
	$n = 7$	2.60 ± 2.07	7.20 ± 3.11	11.00 ± 5.29
	$n = 15$	1.20 ± 1.30	3.80 ± 2.49	7.80 ± 3.03
PCA and FLDA	$n = 1$	5.40 ± 4.51	13.40 ± 6.73	23.00 ± 8.72
	$n = 2$	4.40 ± 2.70	9.40 ± 5.41	17.00 ± 7.07
	$n = 3$	2.40 ± 2.51	6.80 ± 3.56	15.00 ± 6.89
	$n = 7$	0.80 ± 1.30	3.00 ± 2.00	6.20 ± 3.27
	$n = 15$	0.40 ± 0.55	1.40 ± 1.67	3.20 ± 3.56
Parallel BLDA	$n = 1$	7.40 ± 3.65	15.00 ± 4.85	25.20 ± 7.26
	$n = 2$	5.00 ± 2.55	11.80 ± 4.32	18.80 ± 5.67
	$n = 3$	3.80 ± 2.28	9.40 ± 2.88	16.20 ± 6.30
	$n = 7$	2.00 ± 1.58	6.40 ± 3.78	11.00 ± 4.74
	$n = 15$	1.00 ± 1.22	2.00 ± 1.87	6.20 ± 3.49
Parallel FLDA	$n = 1$	7.60 ± 7.06	16.00 ± 7.55	23.20 ± 7.66
	$n = 2$	4.80 ± 2.77	8.60 ± 5.81	15.60 ± 5.98
	$n = 3$	2.80 ± 2.39	8.00 ± 5.39	12.40 ± 7.67
	$n = 7$	1.20 ± 1.30	3.00 ± 2.55	5.80 ± 4.09
	$n = 15$	0.60 ± 0.55	1.00 ± 1.00	2.40 ± 2.51

Table 13. Number of errors for the Emotiv EPOC (Five Subjects).

Algorithm	Number of Averaged Intensifications	2 Eliminated	4 Eliminated	6 Eliminated
PCA and BLDA	$n = 1$	12.80 ± 4.32	23.40 ± 3.05	34.00 ± 4.30
	$n = 2$	14.20 ± 4.21	25.40 ± 8.68	35.40 ± 8.44
	$n = 3$	8.80 ± 2.17	19.60 ± 0.55	32.40 ± 2.88
	$n = 7$	9.40 ± 1.52	23.40 ± 3.85	32.80 ± 3.90
	$n = 15$	12.40 ± 3.21	25.20 ± 6.53	35.80 ± 8.79
PCA and FLDA	$n = 1$	11.40 ± 3.51	24.00 ± 6.24	33.80 ± 6.30
	$n = 2$	12.20 ± 5.63	23.00 ± 5.96	33.20 ± 7.33
	$n = 3$	12.20 ± 3.77	20.80 ± 4.97	33.40 ± 7.96
	$n = 7$	11.40 ± 5.59	23.00 ± 6.40	35.60 ± 6.31
	$n = 15$	12.20 ± 5.67	23.00 ± 6.89	35.20 ± 5.50
Parallel BLDA	$n = 1$	14.80 ± 4.09	25.40 ± 6.99	38.20 ± 4.97
	$n = 2$	13.00 ± 3.94	26.00 ± 5.52	37.80 ± 8.84
	$n = 3$	14.00 ± 5.70	24.40 ± 7.20	35.40 ± 8.62
	$n = 7$	14.20 ± 6.46	24.40 ± 9.07	35.80 ± 7.40
	$n = 15$	11.20 ± 5.40	19.40 ± 8.20	28.80 ± 10.99
Parallel FLDA	$n = 1$	15.20 ± 3.27	29.20 ± 5.02	39.80 ± 4.87
	$n = 2$	11.80 ± 4.02	22.00 ± 6.52	32.60 ± 4.77
	$n = 3$	12.80 ± 3.70	18.40 ± 5.59	31.20 ± 8.50
	$n = 7$	10.80 ± 4.32	19.60 ± 7.50	29.40 ± 10.76
	$n = 15$	9.00 ± 6.04	15.60 ± 9.53	24.40 ± 13.26

Appendix C: Subject Data

Table 14. Subject information.

Phase of Data Collection	Age	Gender	Occupation
Phase I and II	25	Male	Engineering Graduate Student
Phase I and II	21	Male	Undergraduate Neuroscience Student
Phase I	23	Male	Undergraduate Engineering Student
Phase II	20	Male	Undergraduate Student
Phase II	22	Male	Nurse
Phase II	23	Female	Undergraduate English/Pre-Law Student