UNIVERSITY OF CALIFORNIA

Los Angeles

Toward real-time communication using brain-computer interface systems

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy in
Biomedical Engineering

by

William Farran Speier IV

2014

ABSTRACT OF THE DISSERTATION

Toward real-time communication using brain-
computer interface systems

By

William Farran Speier IV
Doctor of Philosophy in Biomedical Engineering
University of California, Los Angeles, 2014
Professor Nader Pouratian, Chair

The ability to communicate using language is a fundamental human function. When this ability is compromised, as it can be in neuromuscular diseases such as amyotrophic lateral sclerosis (ALS) and brainstem strokes, patients stand to lose a significant source of functional independence. Brain-computer interface (BCI) systems help restore communication to these "locked-in" patients, usually relying on P300 evoked response potentials (ERPs) to identify a target character among repetitive serial presentation of possible characters. While the so-called "P300 speller" was first described over 25 years ago, little overall progress has been made with respect to clinical implementation, with major system limitations related to practicality, speed, and accuracy. This work addresses these concerns by using machine learning techniques to optimize the system design, accelerate the character selection process, and integrate natural language domain knowledge into the classifier. This effort has involved several different projects, including selecting the optimal electrode positions using Gibbs sampling, performing unsupervised training with the Baum-Welch algorithm, and incorporating prior language knowledge using particle filtering. The result is an online system requiring only four electrodes that allows users to communicate at an average bit rate that is 75% higher than when using standard methods. These improvements can help to make the P300 speller system a more viable solution for “locked-in” patients, leading to increased functional independence and improved quality of life.

Corey Arnold

Alex Bui

Mark Cohen

Ricky K. Taira

Alan Yuille

Nader Pouratian, Committee Chair

University of California, Los Angeles

2014

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Acknowledgments

I would like to thank UCLA for providing the opportunity to pursue my graduate education and for funding my final year with the Dissertation Year Fellowship. I would also like to that the NIH for funding my research through the NLM Biomedical Informatics Research Training Fellowship (T15-LM007356) and the NIH/NIBIB Medical Imaging Informatics Training Grant (T32-EB016640).

I would like to thank my committee, Dr. Nader Pouratian, Dr. Alex Bui, Dr. Ricky Taira, Dr. Corey Arnold, Dr. Alan Yuille, and Dr. Mark Cohen for providing guidance and feedback through various stages of this work. I would particularly like to thank Dr. Bui and the Medical Imaging Informatics group for admitting me, providing an intellectually stimulating environment, and giving me the freedom to pursue the many directions my interests led me over the years. I would also like to thank the chair of my committee, Dr. Pouratian, for challenging me to apply my engineering knowledge to new fields while maintaining the patient-centered perspective, as well as for setting up the Neurosurgical Brain Mapping and Restoration lab with the equipment and access to resources and subjects necessary for this research.

Chapter 2 contains material published with Corey Arnold, Jessica Lu, Ricky K. Taira, and Nader Pouratian in the Journal of Neural Engineering, volume 9, issue 1, page 016004.

Chapter 3 contains material published with Corey Arnold, Jessica Lu, Aniket Deshpande, and Nader Pouratian in IEEE Transactions on Neural Systems and Rehabilitation Engineering, volume 22, issue 3, pages 678-684.

Chapter 4 contains material submitted for publication with Corey Arnold, Aniket Deshpande, Jennifer Knall, and Nader Pouratian in the Journal of Neural Engineering as “Incorporating Advanced Language Models into the P300 Speller using Particle Filtering.”

Chapter 5 contains material published with Corey Arnold and Nader Pouratian in PLOS ONE, volume 8, issue 10, page e78342.

Chapter 6 contains material published with Aniket Deshpande and Nader Pouratian in Clinical Neurophysiology, doi: 10.1016/j.clinph.2014.09.021.

Chapter 7 contains material published with Itzhak Fried and Nader Pouratian in Clinical Neurophysiology, volume 124, issue 7, pages 1321-1328.

Chapter 9 contains material published with Jennifer Knall and Nader Pouratian in the proceeding of 2013 IEEE EMBS Conference on Neural Engineering, pages 707-710.

VITA

EDUCATION

M.S.E. Computer Science	Johns Hopkins University	2008
B.S. Biomedical Engineering; Applied Mathematics	Johns Hopkins University	2005

PROFESSIONAL EXPERIENCE

University of California, Los Angeles - *Graduate Student* Medical Imaging Informatics (MII) Group Neurosurgical Brain Mapping and Restoration Lab	2009-2015
EDAC Systems, Inc. - *Senior Project Developer*	2008-2009
Johns Hopkins University - *Student Researcher* Oncology Biostatistics Department	2007-2009
Fox Chase Cancer Center - *Visiting Graduate Student* Bioinformatics Department Population Sciences Division	2006-2007
Johns Hopkins University - *Student Researcher* Brain-Computer Interface (BCI) Lab	2005-2006

PUBLICATIONS

Speier, Deshpande, Pouratian. A Method for Optimizing EEG Electrode Number and Configuration for Signal Acquisition in P300 Speller Systems. *Clinical Neurophysiology* 2014 doi:10.1016/j.clinph.2014.09.021.

Singleton, **Speier**, Bui, Hsu. Motivating the Additional Use of External Validity: Examining Transportability in a Model of Glioblastoma Multiforme. *AMIA* 2014.

Ho, **Speier**, El-Saden, Liebeskind, Saver, Bui, Arnold. Predicting Discharge Mortality after Acute Ischemic Stroke Using Balanced Data. *AMIA* 2014.

Speier, Arnold, Lu, Deshpande, Pouratian. Integrating language information with a hidden Markov model to improve communication rate in the P300 speller. *IEEE Transactions on Neural Systems & Rehabilitation Engineering* 2014;22(3):678-684.

Speier, Arnold, Pouratian. Evaluating true BCI communication rate through mutual information and language models. *PLOS One* 2013;8(10): e78432.

Hollada, Marfori, Tognolini, **Speier**, Ristow, Ruehm. Successful Patient Recruitment in CT Imaging Clinical Trials: What Factors Influence Patient Participation? *Academic Radiology* 2013;21(1):52-57.

Speier, Knall, Pouratian. Unsupervised training of brain-computer interface systems using expectation maximization. *IEEE EMBS Conference on Neural Engineering* 2013:707-710.

Speier, Fried, Pouratian. Improved P300 speller performance using electrocorticography, spectral features, and natural language processing. *Clinical Neurophysiology* 2013;124(7):1321-1328.

Lu, **Speier**, Hu, Pouratian. The effects of stimulus timing features on P300 speller performance. *Clinical Neurophysiology* 2012;124(2):306-314.

Speier, Ochs. Updating annotations with the distributed annotation system and the automated sequence annotation pipeline. *Bioinformatics* 2012;28(21):2858-2859.

Hsu, **Speier**, Taira. Automated Extraction of Reported Statistical Analyses: Towards a Logical Representation of Clinical Trial Literature. *AMIA* 2012.

Arnold, **Speier**. A Topic Model of Clinical Reports. *ACM SIGIR* 2012.

Speier, Arnold, Lu, Taira, Pouratian. Natural Language Processing with Dynamic Classification Improves P300 Speller Accuracy and bit Rate. *Journal of Neural Engineering* 2012; 9(1):016004.

Speier, Iglesias, El-Kara, Tu, Arnold. Robust Skull Stripping of Clinical Glioblastoma Multiforme Data. *Medical Image Computing and Computer-Assisted Intervention* 2011; 14(3): 659-666.

Favorov, Lvovs, **Speier**, Parmigiani, Ochs. OionTree XML: A Format to Exchange Gene-Related Probabilities. *J. Biomol Struct Dyn* 2011; 29(2): 417-423.

Wong, Meropol, **Speier**, Sargent, Goldberg, Beck. Cost Implications of New Treatments for Advanced Colorectal Cancer. *Cancer* 2009; 115(10): 2081-91.

Konski, **Speier**, Hanlon, Beck, Pollack. Is Proton beam therapy Cost-Effective in the treatment of Adenocarcinoma of the Prostate? *Journal of Clinical Oncology* 2007; 25(24): 3603-8.

Yu, Weinberger, Sasaki, Egleston, **Speier**, Haffty, Kowalski, Camp, Rimm, Vairaktaris, Burtness, Psyrri. Phosphorylation of Akt (Ser^{473}) Predicts Poor Clinical Outcome in Oropharyngeal Squamous Cell Cancer. *Cancer Epidemiology Biomarkers & Prevention* 2007; 16: 553-558.

Moloshok, Klevecz, Grant, Manion, **Speier**, Ochs. Application of Bayesian Decomposition for Analysing Microarray Data. *Bioinformatics* 2002; 18: 566-575.

1. INTRODUCTION

High brain stem injuries and motor neuron diseases such as amyotrophic lateral sclerosis (ALS) can interrupt the transmission of signals from the central nervous system to effector muscles. In severe cases, these diseases can impair a patient's ability to interact with the environment or to communicate, causing them to become "locked-in" (Laureys et al., 2005). Brain–computer interfaces (BCI) restore some of this ability by detecting electrical signals from the brain and translating them into computer commands (Wolpaw et al., 2002). The computer can then perform actions dictated by the user, whether it be typing text (Farwell and Donchin, 1988), moving a cursor (Wolpaw et al., 1991), or even controlling a robotic prosthesis (Lauer et al., 2000, Pfurtscheller et al., 2000).

The P300 speller is the most commonly used BCI approach for restoring linguistic communication to "locked-in" patients (Farwell and Donchin, 1988). In this system, a user observes a grid of characters on a computer screen (analogous to a visual keyboard) while subsets of characters are illuminated (i.e., flashed) in pseudo-random patterns. When the target character flashes, it provides a visual stimulus that elicits an evoked electroencephalographic (EEG) response. A classifier detects these signals and selects the corresponding target letter or symbol. Traditionally, system noise requires that stimuli be presented many times so that the corresponding signals can be averaged to improve signal-to noise. The resulting typing speeds are therefore slow, prompting research aimed at improving many of the aspects of the system.

Several approaches have been developed to improve performance, including using different stimulus paradigms (Townsend et al., 2010; Jin et al., 2011; Wang et al., 2012), adjusting system parameters (Sellers et al., 2006; McFarland et al., 2011; Lu et al., 2012), and implementing different classifiers (Kaper et al., 2004; Xu et al., 2004; Serby et al., 2005). Alternative methodologies to the P300 speller have also been explored including auditory stimuli (Furdea et al., 2009; Schreuder et al., 2011) and different neurological phenomena such as motor imagery (Blankertz et al., 2006) and steady state visually evoked potentials (SSVEP) (Cecotti, 2010; Xu et al., 2013; Yin et al., 2013). Recent studies have suggested that electrocorticography (ECoG) signals could be used in BCI communication due to its increased signal-to-noise ratio, high spatial resolution, and superior spectral content (Leuthardt et al., 2004; Wilson et al., 2006; Brunner et al., 2009; Miller et al., 2010).

Traditionally, most classification algorithms for BCI communication systems have treated the task as a series of independent signal classification problems. Integrating domain knowledge has the potential to improve the speed and accuracy of communication. In the context of communication, successive character selections are highly dependent and rely heavily on context. These dependencies have been studied in statistical natural language processing (NLP), an engineering field dedicated to achieving computer understanding of natural language. One application often used in domains such as speech recognition is to apply a language model to create probabilities for different interpretations of an input audio signal (Jelinek, 1998). This method can be using in a BCI system by finding the probabilities of possible continuations of text entered in previous trials. This probability distribution provides a prior for subsequent character selections so that text agreeing with the language model is more likely to be selected. Using such a model can potentially increase selection accuracy as well as system speed by reducing the amount of data necessary to make an accurate decision.

While significant research has been conducted on improving classification performance, comparatively little time has been spent on making the system more practical for patients. Existing research systems are expensive, require constant maintenance and debugging, involve lengthy training and setup times, and generally produce a low signal to noise ratio (SNR) despite expert supervision. Because the ultimate goal is to restore a level of autonomy for these patients, the end system cannot rely on heavy expert involvement; the system must be fast and easy to set up and the signal must be stable and have sufficient SNR. Minimization of the required hardware is paramount for a practical BCI solution and work in this area has increased in recent years. Most of these methods, however, are either created empirically for healthy subjects (Kaper et sl., 2004; Krusienski et al., 2008; Hoffmann et al., 2008) or require an initial test using a larger system (Cecotti et al., 2011; Xu et al., 2013; Colwell et al., 2014), so the translation into a more practical system for the target population is uncertain. Modifications to the stimulus paradigm (Townsend et al., 2010; Jin et al., 2012) and signal acquisition method (Brunner et al., 2011; Krusienski and Shih, 2011) have recently been presented with promising results, but the methods have not been optimized and require more thorough evaluation. Recently, unsupervised methods have been adapted (Kindermans et al., 2012) that could reduce the time required for training and eventually could be modified to provide an adaptive classifier for BCI communication.

The goal of this project is to create an advanced BCI communication system by optimizing the interface and incorporating domain knowledge into the existing P300 speller system. This project will advance the field along two complimentary paths: 1) advancing BCI communication through integration of natural language information, and 2) improving system practicality by improving signal quality while reducing cost and setup time. Natural language integration involves designing language models for BCI output, representing the spelling task as a process model, adapting sampling methods for creating prior probabilities based on language information, and creating evaluation metrics for BCI communication. Improving system practicality involves optimizing the number and placement of EEG electrodes, optimizing and evaluating different acquisition and stimulus presentation paradigms, and employing unsupervised training methods for the P300 speller.

1.1. System overview

There are several communication BCI systems currently in development. The most widely used is the visual P300 speller system (Farwell and Donchin, 1988). This system works by presenting a matrix of characters on a graphical display and asking the subject to focus on one target character. Sets of characters, usually consisting of rows and columns of the matrix, are then highlighted sequentially (called a flash), each for a period of about 100-200ms (Fig. 1.1). The order of the row and column flashing is randomized so that it is unpredictable for the subject. In order to assure attentiveness, the subject is instructed to count the number of times that the target character is intensified.

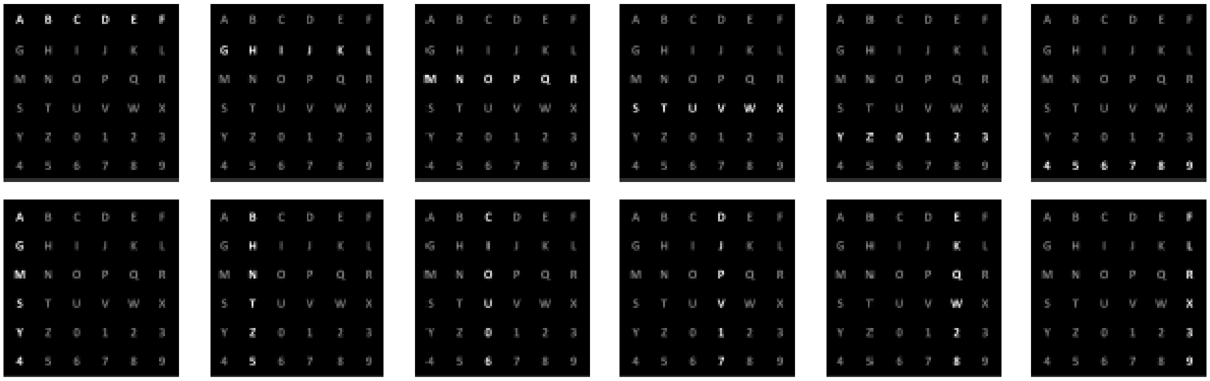


Figure 1.1 Row and column flashes for the P300 speller.

Traditionally, a set number of stimuli are presented to the character for each trial. Then, the system pauses to find the EEG responses for each stimulus, determine the average result for each character, and determine the target character. The process is then repeated for the next character.

In a preliminary training session, subjects are told to spell a specific target word or phrase. Because the goal is known, each stimulus can be labeled based on whether that stimulus contained the target word. This produces a labeled set of stimulus responses that can be used in a traditional machine learning classifier. The output of the training session is generally a set of feature weights where features are average response amplitudes in one of the EEG electrodes (i.e., channels) after a set time delay. In online sessions, these feature weights are applied to the EEG signal in order to make online classifications. Because EEG characteristics vary between user and between sessions, every use of the P300 speller is usually preceded by a new training session.

1.2. Acquisition modality

Traditionally, neurological signals for the P300 speller are acquired via electroencephalography (EEG). EEG measures differences in electrical activity through scalp electrodes. When recording EEG, electrodes are placed in standardized locations and the voltage for each electrode is recorded with respect to either a common reference or a reference montage. Electric potentials recorded by these electrodes reflect the average of the electrical activity in the cortical neurons below them.

The number and placement of electrodes used for the P300 speller varies between studies. The classical P300 space consists of three electrodes: Fz, Cz, and Pz (Sharbrough et al., 1991), but it has been shown that occipital electrodes significantly improve performance in healthy subjects

(Krusienski et al., 2008). Most studies choose instead to collect data from a large set of electrodes and rely on classification algorithms to determine which electrodes are useful (Fig. 1.2). While this strategy ensures that important features are not missed, it increases setup time, patient discomfort, and system hardware requirements. Furthermore, it increases the amount of data and the time required to process it while potentially decreasing classifier performance by increasing the complexity without adding useful data. It is therefore necessary to find an optimal set of electrodes to use in this system.

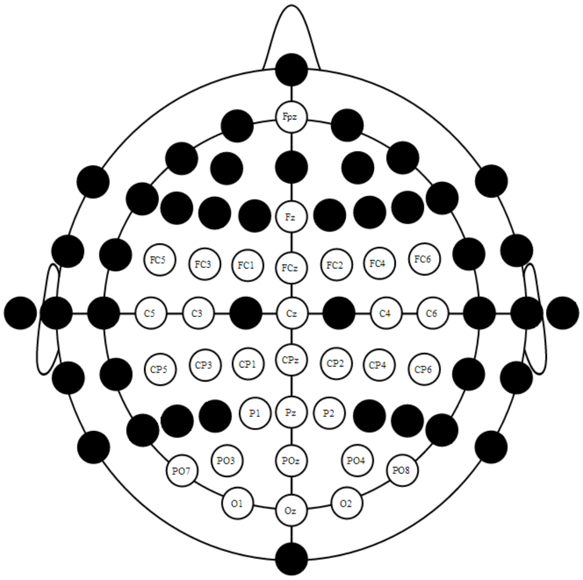


Figure 1.2 A diagram of the EEG electrodes used in a study by Lu et al. (2012). Black circles represent electrodes that are in the 10-20 system that were not included in the Lu study (Sharbrough et al., 1991).

Electrocorticography (ECoG) is the use of subdural electrode grids to record electrical signals directly from the cortex. In order to acquire ECoG signals, a section of skull must be removed and electrodes are placed directly on the brain. These electrodes record the local electrical potential which is the sum of the action potentials of the surrounding neurons. ECoG collects the same signals as EEG with the advantage that it avoids the impedance caused by the skull and scalp. The electrodes can therefore be placed much closer together to get higher spatial precision. The obvious disadvantage is that it is invasive, adding cost, discomfort, and risk to the patient.

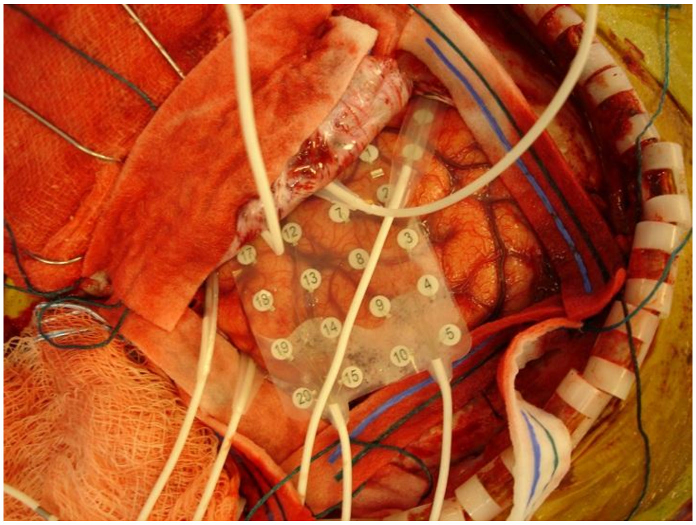


Figure 1.3 Image of an implantation of an electrocorticography (ECoG) grid

The potential of the use of ECoG in the P300 speller system can be tested on epilepsy patients who are candidates for resection surgery. MRI and EEG studies have already been performed on these patients to determine the existence of a lesion, but they do not provide enough information on the extent of the lesion and the amount of surrounding tissue affected. These patients have subdural electrode grids implanted and they are observed for a period up to a week so that data can be collected during seizures.

The most significant downside to using this patient population is that the data collected is dictated by the size and location of the lesion. Because we are using an existing data stream from a clinical grid, we cannot choose the type of grid or its location. If we are concerned with a specific location on the cortex, we are required to wait until a seizure patient is admitted with an implanted grid in the desired location. These patients are also heavily medicated, which can affect the neural signals as well as the patients' attentiveness during the study. For these reasons, it is difficult to find subjects for ECoG-based P300 speller studies (Brunner et al., 2011; Krusienski and Shih, 2011b).

1.3. Neurological Signal

When the target character is illuminated, an evoked response is elicited in the subject's EEG called the P300 (Squires et al., 1975). The P300 is an evoked recognition signal which occurs in the presence of a rare audio or visual cue. It is governed by the so-called "oddball paradigm," which states that a P300 will occur when an uncommon target stimulus occurs during a series of random signals.

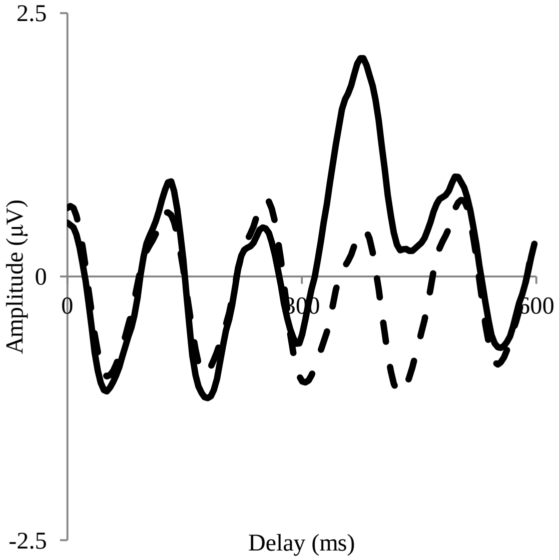


Figure 1.4 Averaged EEG signals for trials with P300 (solid) and without P300 (dashed) collected in the PO_Z location.

After each stimulus, the subject's EEG is measured for a period of time. Because the P300 signal is inconsistent and low in amplitude compared to noise, the results of many sets of flashes need to be combined before a decision is made. For each character, the neural signals corresponding to that character are averaged and the signal is compared to the "ideal" P300 (Fig. 1.4).

Traditionally, the character with the best average signal after a predetermined number of stimuli is accepted as the desired response.

In more occipital electrodes there is often an earlier negative peak in signal amplitude (Fig. 1.5). This signal has been shown to relate to eye gaze, leading to patients with impaired gaze control experiencing a considerable decrease in performance (Brunner et al., 2011). These subjects are still able to use the system, but the performance drops considerably when they cannot fixate directly on the character.

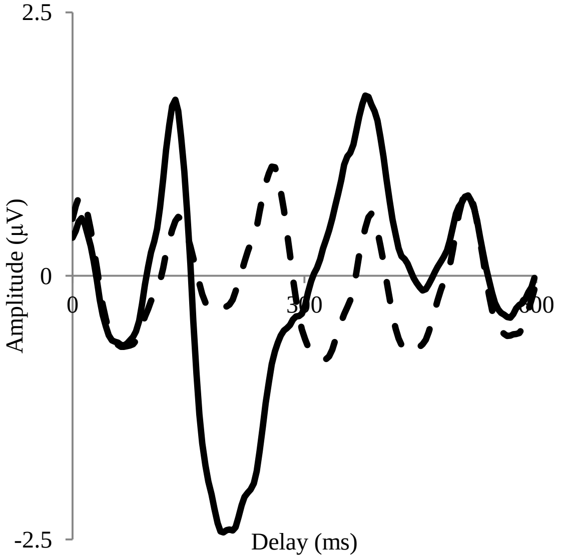


Figure 1.5 Averaged EEG response for stimuli with the target (solid) and without (dashed) collected in the PO_8 location.

Auditory P300 systems have been introduced that alleviate this issue, but introduce more problems as they require a silent environment, involve more complicated tasks, and take much longer to present stimuli (Furdea et al., 2009; Schreuder et al., 2011). Alternative systems based on neurological signal paradigms other than the P300 have recently proposed. The hex-o-spell system uses motor imagery to move a rotating arrow to point at target characters (Blankertz et al., 2006). Cecotti et al. (2010) developed a system that allows users to choose from menus using steady state visually evoked potentials (SSVEP). Yin et al. have developed a system that combines SSVEP and P300 signals to improve classification accuracy (Yin et al., 2013). While promising, none of these methods are widely used, so this project focuses on the visual P300 speller. Because many of the analysis methods discussed here are agnostic to the front end system, they could be incorporated into these alternative BCI systems and potentially lead to similar improvements.

1.4. Dissertation Goals and Organization

The two main goals of this dissertation are to improve the efficacy of the P300 speller by incorporating natural language information into the classifier, and to increase the system's practicality by modifying the system to increase signal quality while minimizing setup time and hardware cost.

The first goal is composed of four parts:

1. Incorporate prior probability based on a model of natural language into the standard classifier.
2. Use a process model to characterize the temporal aspect of BCI output, providing the means for automatic error correction.
3. Implement sampling methods to integrate more sophisticated language models, allowing for more accurate prior probabilities.
4. Design evaluation metrics that better reflect language output, providing a more standardize means for assessing the value of system changes and guiding the development of future systems.

The second goal is composed of four parts:

1. Develop methods for finding the best electrode montages across a population of subjects and determine the best electrode placement in future experiments using healthy volunteers.
2. Evaluate the feasibility and efficacy of invasive approaches to signal acquisition to determine whether the benefits could potentially outweigh the risks and costs of invasive surgery.
3. Optimize the stimulus presentation paradigm to reduce the time required for eliciting ERPs for classification.
4. Exploit the structure of natural language to create methods for unsupervised system training, reducing system setup time and allowing for automatic adaptation to changes in the user's state.

This dissertation is separated into 11 chapters, including the introduction and conclusion.

Chapter 2 describes natural language processing and the method of incorporating it into the P300 Speller using a naïve Bayes classifier. The concept of dynamic stopping is introduced as selections are made once a target confidence threshold is reached. These methods are tested in an offline study consisting of six healthy volunteers.

Chapter 3 uses the prior probabilities described in chapter 2 as transition probabilities in a hidden Markov model. The Forward Backward algorithm is used to track the probability of characters and the Viterbi algorithm is implemented to automatically correct some errors. The dataset from chapter 2 is expanded to 15 healthy subjects for offline evaluation and an online pilot study consisting of five healthy subjects is conducted.

In chapter 4, the trigram language model is replaced by a probabilistic automaton that models a dictionary in addition to local language patterns. Sequential importance sampling is implemented in order to compensate for the additional computational complexity of the larger model. The offline dataset from chapter 3 is used for offline evaluation and a 15 subject online study is conducted for evaluation.

Chapter 5 derives and compares the existing evaluation metrics in BCI communication literature. Shortcomings and inconsistencies in the evaluation of these systems are discussed and a new metric based on mutual information is proposed. A cross-section of BCI communication studies is reevaluated using each of these metrics to show how well each reflects true communication performance.

Chapter 6 proposes a Gibbs sampling method for determining the optimal placement of EEG electrodes for P300 studies. Using the offline dataset from chapter 3, this method is applied to determine the best configuration in healthy subjects. A four channel set is proposed that performs comparably to a full EEG montage. An online study consisting of 15 volunteers is conducted to validate this configuration for prospective studies in healthy subjects.

In chapter 7, two subjects with implanted ECoG grids are tested using the methods from chapter 2. Additionally, frequency features are extracted from the ECoG signal and added to the temporal features traditionally used in the classifier. Spatial analysis is conducted to determine the optimal locations for implanted electrodes and the subjects’ performances are compared to the EEG dataset from chapter 2 to evaluate the potential improvement when using invasive electrodes.

Chapter 8 proposes an optimized stimulus presentation paradigm for the P300 speller. This paradigm is compared to the two main methods in current literature: the standard row column paradigm and the checkerboard paradigm. An offline study consisting of nine healthy subjects is conducted which compares the proposed system to the standard method as well as the checkerboard paradigm with two different grid sizes.

In chapter 9, the hidden Markov model presented in chapter 3 is combined with the Baum-Welch algorithm to achieve unsupervised training of the P300 speller. The offline dataset from chapter 3 is used with hidden labels with the goal of accurately reproducing the target words without training data. The convergence of the study is measured in two cases: when no knowledge about the feature space is known beforehand, and when a general model is provided as a starting point.

Chapter 10 includes a discussion of the results from the previous chapters and as well as suggestions and preliminary work on future adaptations of the methods presented here.

Chapter 11 consists of concluding remarks and a brief discussion of future directions of the field.

PART I: ADVANCING BCI COMMUNICATION

The domain of natural language has been studied extensively in linguistics and has been used in the natural language processing (NLP) field in applications including information extraction, machine translation, and speech recognition. While the most common use of the P300 speller is to generate natural language, information about the output domain has largely been ignored in BCI systems. Although the movement to include this information began only recently, studies have already shown the potential of language integration in BCI communication and it has become a growing area in BCI research.

The earliest application of language information in BCI was to suggest word and phrase completion based on previous selections. After each character selection, dictionary lookups are conducted on the partially completed word and the most common completions are returned. The system then presents an option for the user to select one of these words rather than continuing to type. Ryan et al. (2011) conducted the first study using such a system. Their implementation ran a P300 speller concurrently with the WordQ2 word completion software (version 2.5, Quillsoft, Ltd., Toronto, ON). Middleware was developed that routed the P300 output as input to WordQ2, which then used dictionary lookups to find potential word completions. The ten number spaces in the P300 grid were remapped to WordQ2 commands such as selection of a completion or undoing a previous command. Accuracy using this system decreased due to the added complexity of the task, but typing speed increased drastically because of the ability to select multiple characters at once using word completion. Lee et al. (2011) presented a similar dictionary lookup scheme in a menu-based motor imagery system. Kaufmann et al. (2012)

integrated a dictionary lookup scheme for common German words into the P300 system, showing decreased time required for typing a given sentence across all subjects.

Early systems incorporating language information into a classifier used a simple character n-gram language model. In this type of model, the conditional probabilities of characters are determined based on the previous (n-1) selections. These probabilities are generally determined by finding the relative counts of character patterns in a general language corpus. Speier et al. (2012) presented the first such system, which incorporated trigram probabilities through a naïve Bayes classifier. In this system, the posterior probability distribution over the possible target characters was found by multiplying the probability of the observed signals by the prior probability based on trigram counts. After normalization, this probability was compared against a threshold value to determine whether a selection should be made or more data needed to be collected. Samizo et al. (2013) presented a similar system using Japanese characters. Their system tested the relative performance of their subjects using unigrams, bigrams and trigrams, finding that trigrams provided the fastest typing speed. Orhan (2014) integrated a 6-gram language model with Witten-Bell smoothing into the rapid serial visual presentation (RSVP) speller, a P300-based BCI system that presents single characters sequentially in the center of the user’s screen.

A more sophisticated approach to language integration treats spelling as a process model. In this case, there exists an underlying state model representing the user’s target character. While the subject is focusing on a character, the system is in one state. When the character is selected, the system then transitions to the state represented by the next character. In general, this model is unobserved, so inference of the current state must be made based on the observed EEG signal and the transition probabilities determined by the language model. Park and Kim (2012) created the first such model using a partially observable Markov decision process (POMDP) which models the system state by a set of variables and uses character bigrams to determine transition probabilities. Ulas and Cetin (2013) created an offline system that models BCI spelling as a hidden Markov model (HMM) with trigram transition probabilities. In HMMs, dynamic programming methods can be used to find the optimal sequence of states that generate a series of observations. Speier et al. (2014a) developed an online HMM system that incorporated automatic error correction as well as dynamic stopping.

Process models generally compute transitions by finding a sum or maximum over the state space. While this is possible in simple n-gram models, it quickly becomes intractable as language models increase in complexity. Sampling methods are necessary for estimating the probability distribution over such models so that high probability sequences can still be tracked without losing the ability to run analysis in real time. Speier et al. (2014c) applied sequential importance resampling, a standard particle filtering (PF) method to handle more complicated language models. In this system, a probabilistic automaton was used to represent word frequency in English text. Because the model contains over 200,000 states, maximizing over the entire state space is not possible in a real time system. PF methods estimate the distribution over the state space by projecting possible realizations of the system (called particles) through the model over time. Particles are resampled periodically based on the observed signal, so the existing particle distribution closely reflects the posterior probability of a given character. This method was tested online against simpler language models and showed significant improvements in both typing speed and accuracy.

A major contributor to the increase in performance using language models is based on the ability to dynamically set the number of stimuli presented for a given character based on the probability distribution. Termed dynamic classification, methods for adapting the number of stimuli presented have been presented before (Serby et al., 2005), but they have started with a uniform probability distribution and therefore took longer to reach the required threshold. The naïve Bayes method presented in Speier et al. (2012) was the first to incorporate dynamic stopping along with a language model as stimulus presentation continued until the posterior probability for any character exceeded a set threshold value. Even with a simple language model, this method was able to achieve a 50% increase in average bit rate across subjects. Several subsequent methods have since incorporated similar methods and it is quickly becoming the standard in P300 classification (Park et al., 2012; Samiz et al., 2013; Kindermans et al., 2013; Speier et al., 2014a; Speier et al., 2015).

As new systems are proposed, evaluation metrics play an important role in the direction of research as they are used to evaluate the improvement and relative value of systems' results. Several metrics have been developed for BCI communication output, but most were created with traditional classifiers in mind. They therefore treat all characters as equally probable and do not take interactions between subsequent selections into account. As a result, they generally overestimate the amount of information that is conveyed in a system's output. A metric has been proposed that incorporates some of this information to more accurately assess the true amount of information that is conveyed in a BCI output string (Speier et al., 2013c). It achieves this by measuring the mutual information between the target string and the actual output string with using a trigram language model to represent the interactions between subsequent selections.

This section consists of four chapters. In chapter 2, the integration of a trigram language model using a naïve Bayes classifier is described. Chapter 3 describes an online HMM method that incorporates the trigram model in a process model used to automatically correct errors. Chapter 4 models language using a probabilistic automaton and applies a particle filtering method to integrate it into an online system. In chapter 5, the existing BCI communication metrics are described and evaluated and a mutual information evaluation metric is proposed.

2. NAÏVE BAYES

While the P300 speller is designed to provide a means for communication, most attempts at system optimization have not taken advantage of existing knowledge about the language domain. Existing analyses treat character selections as independent elements chosen from a set with no prior information. In practice, we can use information about the domain of natural language to create a prior belief about the characters to be chosen. By adding a bias to the system based on this prior, we hypothesize that both system speed and accuracy can be improved.

Statistical natural language processing (NLP) is an engineering field dedicated to achieving computer understanding of natural language. One application often used in domains such as speech recognition is to apply a language model to create probabilities for different interpretations of an input audio signal (Jelinek, 1998). We can use this method in a BCI system by finding the probabilities of all possible continuations of the text entered in previous trials. This probability provides a prior for subsequent character selections so that text agreeing with the language model is more likely to be selected.

This study exploits prior information using NLP to improve the speed and accuracy of the P300 speller. The system will determine the confidence of a classification by weighting the output of the standard stepwise linear discriminant analysis (SWLDA) algorithm with prior probabilities provided by a trigram language model. The number of flashes used to classify a character is dynamically set based on the amount of time required for the system to reach a predetermined confidence threshold.

2.1. Stepwise Linear Discriminant Analysis

SWLDA is a classification algorithm that selects a set of signal features to include in a discriminant function (Draper and Smith, 1981). The signals in the training set are assigned labels based on two classes: those corresponding to flashes containing the attended character and those without the attended character. The algorithm uses ordinary least-squares regression to predict class labels for the training set. It then adds the features that are most significant in the forward stepwise analysis and removes the least significant features in the backward analysis step. These steps are repeated until either the target number of features is met or it reaches a state where no features are added or removed (Krusienski et al., 2006).

Each new signal is then reduced to a score that reflects how similar it is to the attended class. The score for each flash in the test set, y_t^i , is computed as the dot product of the feature weight vector, $\boldsymbol{w}$, with the features from that trial's signal, $\boldsymbol{z}_t^i$

$$y_t^i = \boldsymbol{w} \cdot \boldsymbol{z}_t^i$$

For the static classification, the score for each possible next character, x_t , is the sum of the individual scores for flashes that contain that character:

$$g(x_t) = \sum_{i:x_t \in A_t^i} y_t^i$$

where A_t^i is the set of characters illuminated for the i th flash for character t in the sequence.

The number of flashes is predetermined in the static method, so the classifier will choose $\operatorname{argmax}_{x_t} g(x_t)$ after the set number of flashes is reached. In order to optimize this method, the number of flashes was varied from 1 to 15 and the associated speeds, accuracies and bit rates were recorded. For each subject, the number of flashes was chosen that optimized the bit rate.

2.2. Dynamic Stopping

As in the static method, the dynamic classification method (DYN) uses nine-fold cross-validation to obtain a training set for SWLDA. Instead of summing the scores as in the static method, it converts scores into probabilities and selects characters once a probability threshold is met. The classifier is first trained as in section 2.1. Scores for each flash in the training set were then computed and the distributions for the attended and non-attended signals were found.

While it has been shown that consecutive flashes are not independent (Citi et al., 2010), we made the simplifying assumption that each flash's score was drawn independently from one of these distributions. We made the further assumption that the distributions were Gaussian, which was tested using Kolmogorov–Smirnov tests for normality (Massey, 1951). The probability density function (PDF) for the likelihood probability can then be computed,

$$f\left(y_t^i \middle| x_t\right) = \begin{cases} \dfrac{1}{\sqrt{2\pi\sigma_a^2}} e^{-\frac{1}{2\sigma_a^2}\left(y_t^i - \mu_a\right)^2} & if\ x_t \in A_t^i \\ \dfrac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{1}{2\sigma_n^2}\left(y_t^i - \mu_n\right)^2} & if\ x_t \notin A_t^i \end{cases}$$

where μ_a, σ_a^2 , μ_n and σ_n^2 are the means and variances of the distributions for the attended and non-attended flashes, respectively.

Bayes' theorem was used to determine the probability of each character given the flash scores and the previous decisions (Duda et al., 2001).

$$p(x_t \,|\boldsymbol{y}_t, x_{t-1}, \ldots, x_0) \;=\; \frac{p(x_t \mid x_{t-1}, \ldots, x_0) p(\boldsymbol{y}_t \mid x_t, \ldots, x_0)}{p(y_t \mid x_{t-1}, \ldots, x_0)}$$

If we assume that the individual flashes are conditionally independent given the current attended character, the posterior probability is

$$p(x_t \,|\boldsymbol{y}_t, x_{t-1}, \ldots, x_0) \propto \; p(x_t \mid x_{t-1}, \ldots, x_0) \prod_i f\left(y_t^i \mid x_t\right)$$

where $p(x_t \mid x_{t-1}, \ldots, x_0)$ is the prior probability of a character given the history and $f\left(y_t^i \mid x_t\right)$ are the PDFs for the likelihood probability. Because the dynamic method uses a uniform prior probability, the posterior simplifies to

$$p(x_t \,|\boldsymbol{y}_t, x_{t-1}, \ldots, x_0) \;\propto\; \prod_i f\left(y_t^i \mid x_t\right)$$

A threshold probability, p_{thresh}, is then set to determine when a decision should be made. The program flashes characters until either $\max_{x_t} p(x_t \,|\boldsymbol{y}_t, x_{t-1}, \ldots, x_0) \geq p_{thresh}$ or the number of sets of flashes reaches 15. The classifier then selects the character that satisfies

$\operatorname{argmax}_{x_t} p(x_t \,|\boldsymbol{y}_t, x_{t-1}, \ldots, x_0)$. The speeds, accuracies and bit rates were found for values of p_{thresh} between 0 and 1 in increments of 0.01. The threshold probability that maximized the bit rate was chosen for each subject.

2.3. Trigram Model

The naïve Bayes (NB) method builds on the dynamic methodology. While the dynamic method had uniform prior probabilities, here NLP is integrated to provide language-specific prior probabilities. Prior probabilities for characters were obtained from frequency statistics in an

English language corpus. This probability was simplified using the second-order Markov assumption to create a trigram model (Manning and Schütze, 1999). The prior probability that the next character is x_t given that the last two characters chosen were x_{t-1} and x_{t-2} is then equal to the number of times that all three characters occurred together in the corpus divided by the number of times the last two characters occurred together:

$$p(x_t|x_{t-1}, \dots, x_0) \approx \frac{c(x_{t-2}, x_{t-1}, x_t)}{c(x_{t-2}, x_{t-1})}$$

where $c(x_{t-2}, x_{t-1}, x_t)$ is the number of occurrences of the string '$x_{t-2}x_{t-1}x_t$' in the corpus.

For the first two characters in a word, x_{t-1} and x_{t-2} are not defined. In the case of the first character, the prior probability is the number of words that start with that character divided by the number of words in the corpus. Similarly, the probability for the second character in the word is the number of words that start with '$x_{t-1}x_t$' divided by the number of words that start with 'x_{t-1}':

$$p(x_t|x_{t-1}, \dots, x_0) \approx \begin{cases} \dfrac{c(start, x_t)}{c(start)} & if\ t = 0 \\ \dfrac{c(start, x_{t-1}, x_t)}{c(start, x_{t-1})} & if\ t = 1 \\ \dfrac{c(x_{t-2}, x_{t-1}, x_t)}{c(x_{t-2}, x_{t-1})} & otherwise \end{cases}$$

where $c(start, x_{t-1}, x_t)$ and $c(start, x_t)$ are the numbers of words that start with '$x_{t-1}x_t$' and 'x_t' respectively and $c(start)$ is the total number of words in the corpus.

Trigrams for the English language were obtained from the Brown corpus (Francis and Kucera, 1979). The Brown corpus contains over 2 million words compiled from various types of documents published in the United States in 1961.

2.4. Validation

2.4.1. Protocol

The subjects were six healthy male graduate students and faculty with normal or corrected to normal vision between the ages of 20 and 35. Only one subject (subject 2) had previous BCI experience. The system used a 6 × 6 character grid, row and column flashes, an ISI of 125 ms and a flash duration that varied between 31.25 and 62.5 ms. Each subject underwent nine trials consisting of spelling a five letter word (Table 2.1) with 15 sets of 12 flashes (six rows and six columns) for each letter. The choice of target words for this experiment was independent of the language model used in the naïve Bayes method. BCI2000 was used for data acquisition (Schalk et al., 2004) and analysis was performed offline using MATLAB (version 7.10.0, MathWorks, Inc, Natick, MA). Three analysis methods were compared: the static method where the number of flashes is predetermined, the dynamic method that uses a threshold probability and a uniform prior, and a naïve Bayes method that incorporates a character trigram language model.

Table 2.1 Target words for the nine trials

UNITS	MINUS	NOTED
DAILY	SCORE	GIANT
HOURS	SHOWN	PANEL

Training is performed using nine-fold cross-validation where the test set is one of the trial words and the other eight are the training set.

2.4.2. Evaluation

Evaluation of a BCI system must take into account two factors: the ability of the system to achieve the desired result and the amount of time required to reach that result. The efficacy of the system can be measured as the selection accuracy, which we evaluated by dividing the number of correct selections by the total number of trials. For each model we also calculated the selection rate (SR). First, the average amount of time for a selection is found by adding the gap between flashes (3.5 s) to the product of the amount of time required for a flash (0.125 s), the average number of sets of flashes ($\bar{s}$) and the number of flashes in each set (12). The selection rate measured in selections per minute is then the inverse of the average selection time:

$$SR = \frac{60}{3.5 + 0.125 * 12 * \bar{s}}$$

Because there is a tradeoff between speed and accuracy, we also use bit rate as a metric which takes both into account. The bits per symbol, B, is a measure of how much information is transmitted in a selection taking into account the accuracy and the number of possible selections (Pierce, 1980):

$$B = \log_2 N + P \log_2 P + (1 - P) \log_2 \frac{1 - P}{N - 1}$$

where N is the number of characters in the grid (36) and P is the selection accuracy. The bit rate (in bits min^{-1}) can then be found by multiplying the selection rate by the bits per symbol. Significance was tested using paired two-sample t-tests with five degrees of freedom.

Although the number of flashes was fixed for all offline trials, different selection rates were simulated by limiting the number of data available for the classification algorithm. For example, if the confidence threshold is reached after six sets of flashes, the classification algorithm only uses the data from the first six sets and omits the remaining nine.

2.4.3. SWLDA Results

Using the SWLDA method, all subjects were able to type with varying levels of performance. The best performer (subject 1) was able to achieve 95% accuracy after 3 sets of flashes, while the worst performer (subject 5) reached a maximum of 82% after 15 flashes. The accuracy increased with the number of flashes for all subjects and five out of six were able to exceed 90% accuracy within 15 sets of flashes (Figs. 2.1 and 2.2). The optimal number of sets of flashes varied from 3 to 8, which yielded bit rates from 12.86 to 35.10 (Table 2.2). In general, subjects that performed better achieved an optimal bit rate in fewer flashes (Fig. 2.3). On average, the subjects had 36% accuracy after a single set of flashes which increased to about 95% after 15 sets (Fig. 2.1). The average selection rate for the static method was 5.91, the average accuracy was 82.97% and the average bit rate was 22.07.

Table 2.2 Results for the SWLDA, Dynamic, and naïve Bayes methods optimized for bit rate.

	selections/min			Accuracy			bit rate		
Participant	static	dynamic	NB	static	dynamic	NB	static	dynamic	NB
1	7.50	10.29	10.91	95.56	88.89	93.33	35.10	44.03	48.81
2	6.32	7.06	8.07	86.67	91.11	97.78	24.76	30.22	39.57
3	5.45	6.95	8.33	97.78	100.00	100.00	26.74	35.93	43.08
4	7.50	5.16	5.81	66.67	93.33	95.56	19.07	23.08	27.17
5	4.80	5.22	4.92	68.89	68.89	84.44	12.86	13.98	18.43
6	3.87	4.04	5.83	82.22	95.56	88.89	13.87	18.89	21.83
average	5.91	6.45	7.31	82.97	89.63	93.33	22.07	27.69	33.15

2.4.4. Dynamic Results

Score distributions were found by taking the histograms of the scores from the attended and non-attended signals for each subject (Fig. 2.4). Kolmogorov–Smirnov tests for normality were performed to verify our Gaussian assumption and none were found to be significant after Šidák correction for multiple comparisons. While the shape of the distributions was similar for all subjects, some exhibited better separation between the attended and non-attended scores (Fig. 2.4(a)).

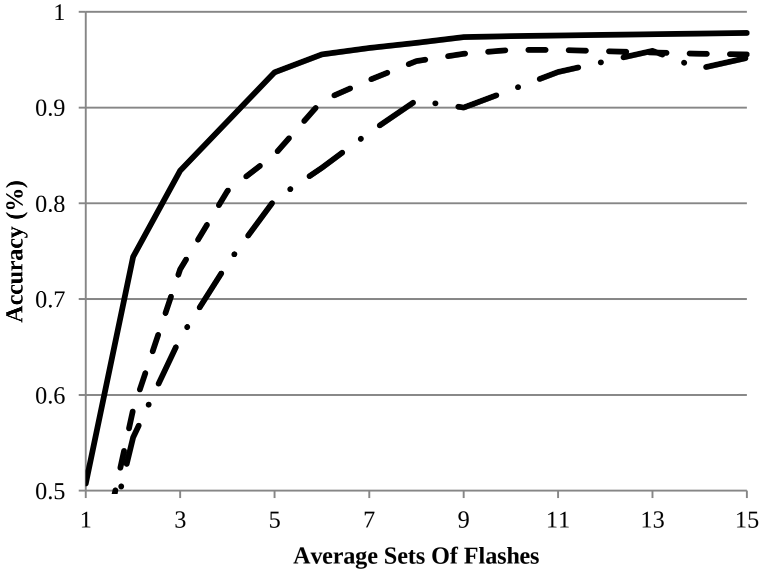
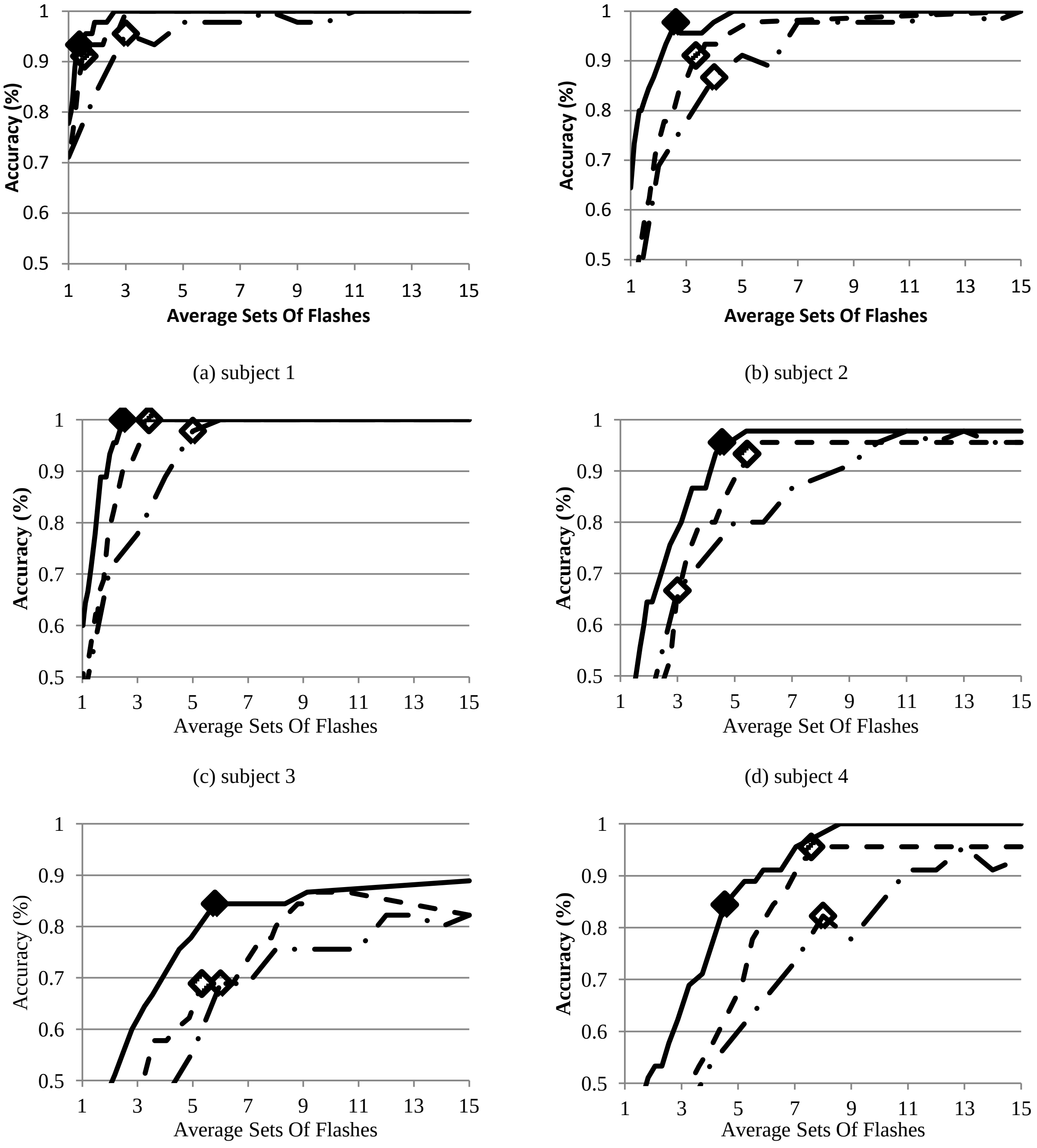


Figure 2.1 Average accuracies: average accuracy across subjects for the static (chain curve), dynamic (broken curve) and naïve Bayes (full curve) methods versus the average number of sets of flashes required to make a decision.

The maximum bit rates using dynamic classification improved by 25% overall (p = 0.003), ranging from 8% (subject 5) to 36% (subject 3) compared to the static method (Table 2.2). On average, the accuracy and selection rate trended upward, but were not statistically significant (p=0.11 and p=0.23, respectively). In some cases, however, a decreased accuracy (subject 1) or selection rate (subject 4) was reported for the dynamic method relative to the static method because of the optimization based on bit rate. In these cases, the optimal bit rate may occur at a lower accuracy but a much higher selection rate or vice versa.



(a) subject 1

(b) subject 2

(c) subject 3

(d) subject 4

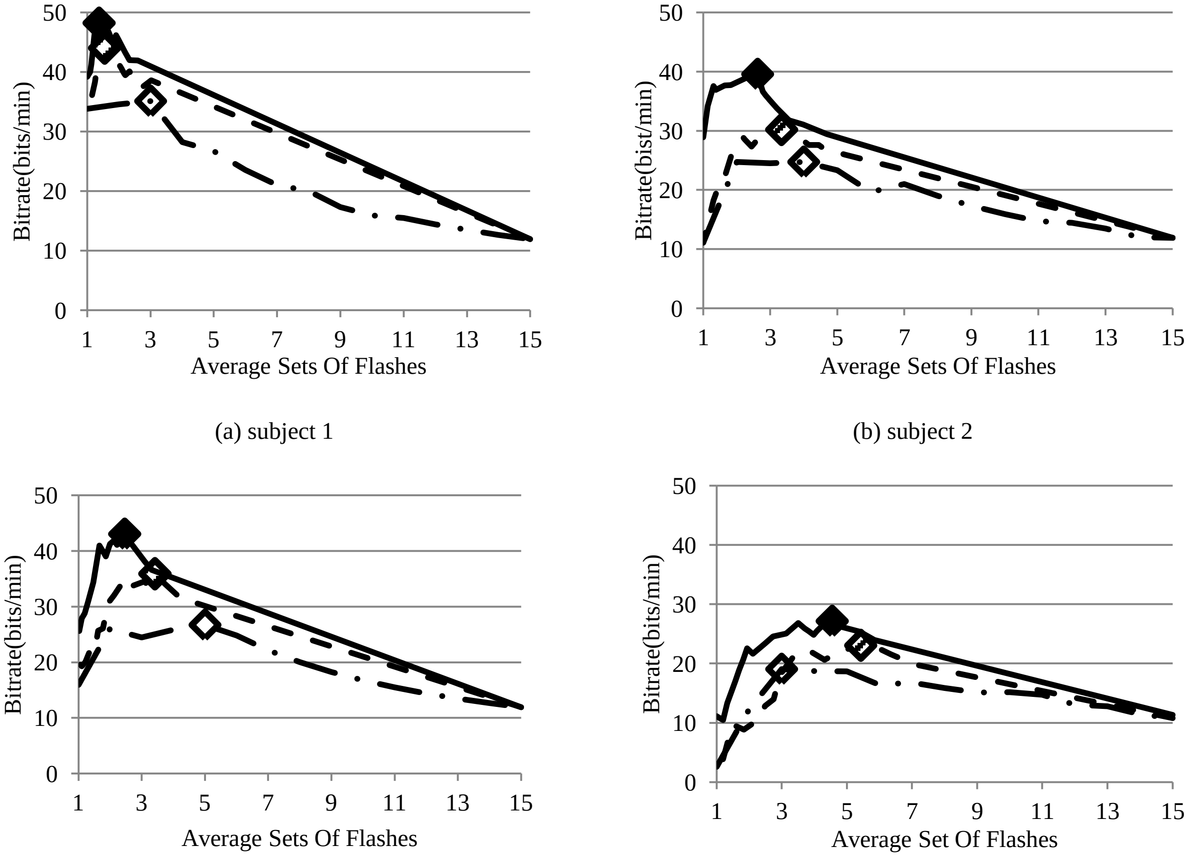
(e) subject 5

(f) subject 6

Figure 2.2 Subject accuracies: average accuracy for each subject using the static (chain curve), dynamic (broken curve), and naïve Bayes (full curve) methods versus the average number of sets of flashes required to make a decision. The markers represent the values on the curve that correspond to the optimal bit rate for each method.

2.4.5. Trigram Results

As the threshold probability varied for the naïve Bayes method, it achieved the best accuracies for any given selection rate (Fig. 2.2). Four of the subjects had 100% accuracies within nine sets of flashes and subject 4 had all characters but one correct within six sets of flashes. Only subject 5 failed to reach 90% accuracy, but an improvement to 84% accuracy was seen within six sets of flashes.

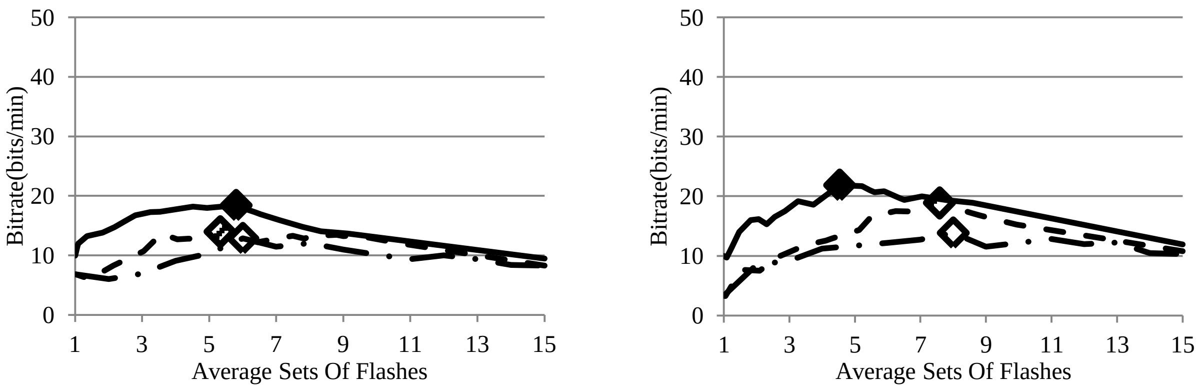


(a) subject 1

(b) subject 2

(c) subject 3

(d) subject 4

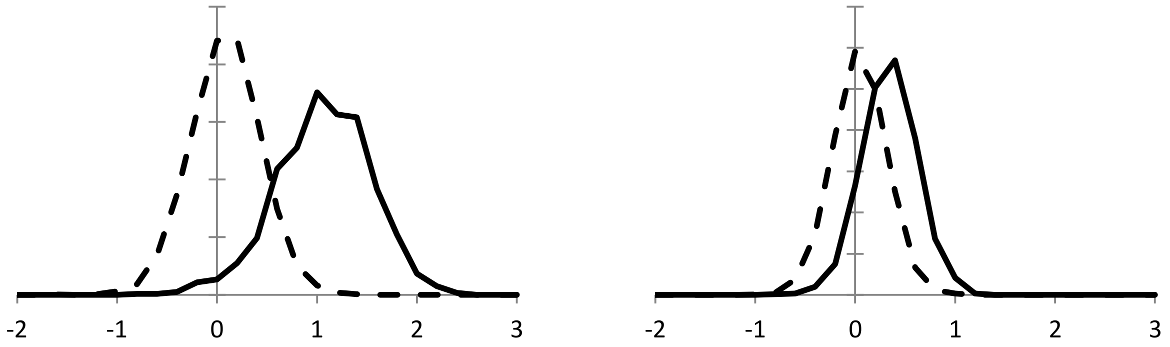


(e) subject 5

(f) subject 6

Figure 2.3 Subject bit rates: average bit rate for each subject using the SWLDA (chain curve), dynamic (broken curve), and naïve Bayes (full curve) methods versus the average number of sets of flashes required to make a decision. The markers represent the values on the curve that correspond to the optimal bit rates for each method.

The overall improvement from the static method to the naïve Bayes method was between 40% and 60% for each subject. The average bit rate across subjects improved by 50% from 22.07 to 33.15 (p = 0.0008). The accuracy increased from 82.97% to 93.33% (p = 0.03) and the selection rate trended up from 5.91 to 7.31, but was not statistically significant (p = 0.06).



(e) subject 1

(e) subject 5

Figure 2.4 SWLDA score distributions: histograms of the attended (solid curve) and non-attended (broken curve) scores from SWLDA.

2.5. Discussion

Current BCI communication systems ignore domain knowledge when processing natural language. Most systems also use static trial lengths so that all classifications are given the same amount of input information regardless of classification difficulty. Integration of dynamic classification and NLP addresses these shortcomings, improving performance in offline analysis.

2.5.1. Prior knowledge and dynamic classification

The dynamic method was able to improve speed by rendering a decision as soon as it became confident of a classification. At the same time, it increases accuracy by analyzing additional flashes to improve confidence in more challenging classifications. There is also the potential that faster feedback afforded by the dynamic method could improve user attentiveness, but this would require online analysis to observe.

The naïve Bayes method added a prior probability to the dynamic method based on the language model. This helped the system reach the threshold probability more quickly by adding additional probabilistic information from the linguistic domain rather than presuming equal a priori probabilities for all characters. It also improved accuracy as it increased the probability of selections that were consistent with natural language.

2.5.2. Significant advance

The static method achieved an average bit rate of 22.07 across subjects, which is consistent with previous studies (Townsed et al., 2010; Ryan et al., 2011). Our dynamic method improves the bit rate by 25% on average to 27.69. Using the language model for prior probabilities increased the bit rate by 40–60% for each subject in this study for an average of 33.15. To put this in context, Townsend et al. reported an average bit rate increase of 19.85–23.17 using their improved flashing paradigm (Townsend et al., 2010).

2.5.3. Similar work

Serby et al. (2005) implemented a maximum likelihood (ML) method which varies the number of flashes used to classify a character using a threshold as in our dynamic method. Their method differs in that it makes a decision when a target score is met rather than a confidence threshold. In situations where the classifier gives high scores to multiple characters, their method makes a decision if any of them exceed the target score. Because our method converts scores into a confidence probability, it continues flashing until there is enough information to confidently choose one of the characters.

Ryan et al. (2011) created a system that attempted to take advantage of the language domain by adding suggestions for word completion. Their system differs in that it does not use a language model, but instead performs dictionary lookups as characters are selected. This language information is incorporated into the user interface as several of the cells in their character matrix are reserved for word completions. Their approach has several limitations. Characters are removed from their interface to make room for word completions, resulting in a smaller possible output vocabulary and a reduced system bit rate. Also, their graphical interface was more complicated resulting in a lower reported accuracy.

Nevertheless, their study showed that word completion can improve the speed of a BCI system, so it could be beneficial to use it in conjunction with our method. Because our system integrates the language information into the signal processing method and their changes are exclusively in the user interface, they are essentially parallel tracks that can be integrated.

2.5.4. Limitations and future directions

More advanced models that better utilize knowledge of linguistic structure will likely provide even greater improvements than the work presented here. For example, a simple improvement would be to include a model with word probabilities. The corpus used in this study contains part of speech tags which could provide additional prior information. Discourse and context information can also be integrated into this system.

The corpus used in this model was chosen because it is large enough to give reliable trigram counts and because it contains text samples from a variety of domains. Clinical implementations of this system may prefer corpora that are more specific to the patients' needs.

This method is independent of system parameters, grid size and flashing paradigm, so it can be incorporated into most other systems as well. Also, the naïve Bayes method and language model prior can be combined with any classifier that returns a likelihood probability. Studying the effects of NLP in such systems remains as future work.

This study was performed to demonstrate a proof of concept for the use of a language model in BCI communication. While the results are encouraging, it remains to be seen if the improvement in bit rate translates into improved performance in a live communication system. The next step is to implement NLP in an online system and to measure the realized bit rate increase.

2.5.5. Conclusion

Natural language contains many well-studied structures and patterns. Understanding of this domain information can greatly improve the processing and creation of language. This study showed that utilizing natural language information can dramatically increase the speed and accuracy of a BCI communication system.

3. HIDDEN MARKOV MODELS

Automatic error correction is another technique that has been largely unexplored in BCI communication. Traditionally, users have been instructed to correct errors as they occur, but in some cases such corrections may made automatically. Many non-BCI typing methods employ automatic correction, including programs for word processing (Hart-Davis, 2011) and text messaging (Dunlop and Crossan, 2000). The field of BCI has witnessed some movement in this direction, as illustrated in a system by Ryan et al. (2011) whereby subjects can ignore errors when they are typed and correct them later with suggested words from a dictionary.

In this work, we build upon our previous system by modeling typing with the P300 speller as a hidden Markov model (HMM). An HMM treats typing as a sequential process where each character selection is influenced by previous selections. The model is hidden because we cannot observe user intent directly. Instead, we use the Viterbi algorithm to determine the optimal sequence of target characters given the observed EEG signal. This method was compared offline with a standard stepwise linear discriminant analysis (SWLDA) method with dynamic stopping as well as the naïve Bayes classifier (NB) from chapter 2 (Speier et al., 2012) on a set of 15 healthy subjects. These results were then verified through a five subject online pilot study.

3.1. Forward Backward Algorithm

Hidden Markov models are used to model Markov processes that cannot be directly observed, but can be indirectly estimated by state-dependent output. The goal of such systems is to determine the optimal sequence of states in the Markov process that could have produced an observed output sequence.

The HMM method treats typing as an nth order Markov process. States in the process consist of tuples representing the target character and the previous $n-1$ targets, $x_t = \langle x_t, \dots, x_{t-n+1} \rangle$. Transition probabilities correspond to the conditional probability of the next state, x_t, given the previous state, x'_{t-1},

$$p(\boldsymbol{x}_t | \boldsymbol{x}'_{t-1}) = \begin{cases} p(x_t | x_{t-1}, \dots, x_{t-n}) & x_i = x'_{i-1}, \forall i \\ 0 & otherwise \end{cases}$$

A typed word is then simply a sequence of states of the Markov process, $x = (x_0, \dots, x_n)$. Because we cannot directly inspect the states of the process, we observe indirectly through the EEG signals. The EEG response is dependent only on the current state and governed by the conditional probability, $p(y_t | x_t)$, which is defined as in the NB method. The goal is to determine x through observation of the EEG signals, $Y = (y_0, \dots, y_n)$.

At each time point, t, the probability of the current state is computed using the forward step of the forward-backward algorithm:

$$p(x_t | y_t, \dots, y_0\ ) \propto p(y_t | x_t) \sum_{x'_{t-1}} p(x_t | x'_{t-1}) p(x'_{t-1} | y_{t-1}, \dots, y_0)$$

As in the NB method, a selection occurs when the probability of a target character exceeds a threshold probability, p_{Thresh}. This is found by summing over all of the states that share the same character at time t.

$$\max_{x_t} p(x_t | y_t, \dots, y_0) = \max_{x_t} \sum_{x_{t-1}, \dots, x_{t-n+1}} p(x_t | y_t, \dots, y_0) \geq p_{Thresh}$$

3.2. Viterbi Algorithm

At each time step, the Viterbi algorithm is used to determine the path to each state with the highest probability.

$$V_t(x_t) = \max_{x'_{t-1}} p(y_t|x_t)p(x_t|x'_{t-1})V_{t-1}(x'_{t-1})$$

Back pointers are saved so that the optimal sequence ending in that state can be retrieved (Fig. 3.1). For each state, x_t, a pointer is created to the state which satisfies

$$\underset{x'_{t-1}}{argmax}\, p(y_t|x_t)p(x_t|x'_{t-1})V_{t-1}(x'_{t-1})$$

The optimal state for the current time step, x_t, is selected such that it satisfies

$$\underset{x_t}{argmax}\, p(x_t|y_t, \dots, y_0)$$

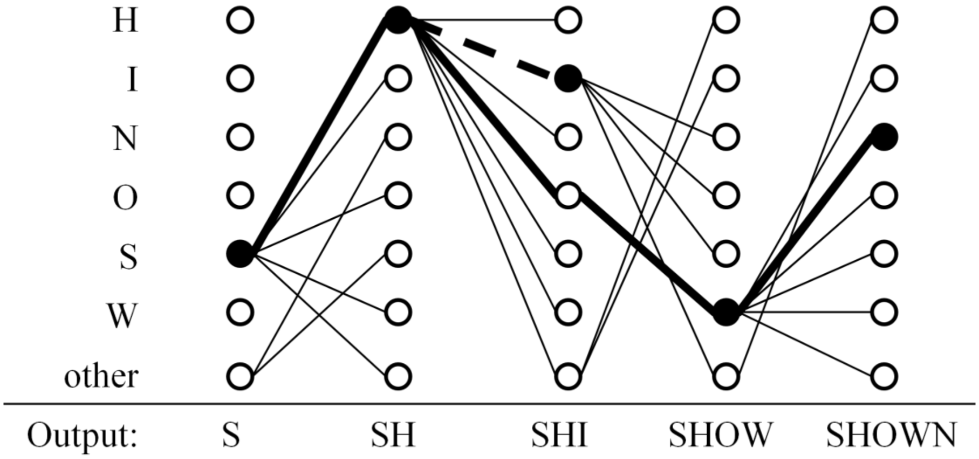


Figure 3.1 Simplified Viterbi trellis for subject B spelling the word “shown.” At time t=3, the character ’I’ has the highest probability, resulting in the output “shi” after following the back pointers (dotted lines). At time t=4, the character ‘W’ has the highest probability and the back pointers (bold lines) produce the output “show,” correcting the previous mistake.

The back pointers are then followed from the selected state to find the optimal sequence of states up to the current time step. The corresponding sequence of characters is considered the most probable string typed. Each time a selection is made, the entire sequence is overwritten by the current optimal string. In most cases, the sequence x_{t-1} has significant overlap with x_t, so the new string is simply the old string with an appended character and possibly some corrected errors. For example, at a given time t, the state x_t may have the highest probability. However, at time $t+1$, state x_{t+1} may have the highest probability with an optimal transition from x'_t. The system would then go back and change the previous character to x'_t (Fig. 3.1).

This study used a second order Markov process (i.e., n=2) for this method to stay consistent with the language model used by the NB method. The states of the model are then $\boldsymbol{x}_t = \langle x_t, x_{t-1} \rangle$ and the transition probabilities are the conditional probabilities $p(x_t|x_{t-1}, x_{t-2})$. As in the previous methods, the offline speeds, accuracies, and bit rates were found for values of p_{Thresh} between 0 and 1 in increments of 0.01 and the threshold probability that maximized the bit rate was chosen for each subject.

3.3. Validation

3.3.1. Offline Dataset

The offline dataset was an expansion of the set described in chapter 2 (Speier et al., 2012). The subjects in the offline dataset were 15 healthy graduate students and faculty with normal or corrected to normal vision between the ages of 20 and 35. Only one subject (subject F) had previous experience using a BCI for typing. The system used a 6 x 6 character grid, row and column flashes, and an interstimulus interval (ISI) of 125 ms. Each subject underwent between 8 and 10 trials consisting of spelling a five letter word with 15 sets of 12 flashes (six rows and six columns) for each letter. Three analysis methods were compared: SWLDA, NB, and HMM. The choice of target words for this experiment was independent of the trigram language model used in the NB and HMM methods.

3.3.2. Protocol

The subjects for the online study consisted of five healthy volunteers with normal or corrected to normal vision between the ages of 20 and 30. The training sessions for these subjects consisted of three sessions of copy spelling 10 character phrases. Each subject then chose a target phrase to spell in online sessions. In each session, the subject had five minutes to spell as much of the phrase as they could using one of the three analysis methods: SWLDA (without dynamic stopping), NB, or HMM. Subjects were instructed not to correct errors and to repeat the phrase if they completed it in under five minutes. Two different threshold values were used for each of the analysis methods (5 and 10 sets of flashes for SWLDA, and .95 and .98 for NB and HMM) for a total of six sessions. The threshold that yielded the best value for each method was chosen as the optimal value and the corresponding output was used to determine the subject's performance.

BCI2000 (Schalk et al., 2004) was used for data acquisition and online analysis. Offline analysis was performed using MATLAB (version 7.10.0, MathWorks, Inc, Natick, MA).

3.3.3. Offline Results

When using SWLDA in offline analysis, all subjects were able to type with varying levels of performance. The best performer (subject D) was able to achieve 89% accuracy at a rate of 9.96 selections per minute, while the worst performer (subject C) achieved an accuracy of 80% at a rate of only 3.96 selections per minute. The accuracy increased with the number of flashes for all subjects and 12 of the 15 were able to exceed 90% accuracy within 15 sets of flashes.

The optimal number of sets of flashes varied from 3 to 8, which yielded bit rates from 13.54 to 40.82 (Table 3.1). In general, subjects that performed better achieved an optimal bit rate in fewer flashes. On average, the subjects had a 29% accuracy after a single set of flashes which increased to about 95% after 15 sets. The average selection rate for the SWLDA method was 5.87, the average accuracy was 88.82%, and the average bit rate was 24.44.

Table 3.1 Optimal selection rates, accuracies, and information transfer rates for the 15 subjects after optimizing on ITR in offline analysis.

	SR (selections/minute)			ACC (%)			ITR (bits/minute)		
Subject	SWLDA	NB	HMM	SWLDA	NB	HMM	SWLDA	NB	HMM
A	8.07	8.33	9.80	93.33	100.00	95.56	36.13	43.08	45.87
B	5.33	5.81	7.42	88.89	95.56	88.89	21.82	27.17	30.38
C	3.96	4.09	8.45	80.00	95.00	67.50	13.54	18.92	21.91
D	9.96	10.91	10.83	88.89	93.33	97.78	40.82	48.81	53.08
E	3.98	5.83	5.20	95.56	84.44	97.78	18.64	21.83	25.47
F	7.21	8.07	8.99	95.56	97.78	95.56	33.76	39.57	42.09
G	5.33	6.03	6.56	94.00	96.00	92.00	24.18	28.48	28.60
H	8.66	9.06	10.17	88.00	90.00	90.00	34.86	37.96	42.59
I	5.08	6.36	9.38	78.00	86.00	78.00	16.68	24.58	30.77
J	4.78	6.32	5.86	90.00	84.00	94.00	20.03	23.46	26.57
K	3.98	8.24	6.68	84.00	68.00	80.00	14.80	21.63	22.85
L	5.62	8.08	7.36	92.00	86.00	90.00	24.47	31.23	30.80
M	4.13	6.61	6.19	82.00	82.00	84.00	14.72	23.57	22.98
N	7.60	8.01	8.98	94.00	96.00	90.00	34.48	37.83	37.62
O	4.40	7.53	6.28	88.00	78.00	84.00	17.71	24.70	23.33
average	5.87	7.28	7.88	88.82	88.81	88.34	24.44	30.19	32.33

The maximum bit rates using the naïve Bayes classifier improved by 50% on average (p < 10-8), ranging from 39% (subject E) to 65% (subject C) compared to the SWLDA results (Table 3.1). The selection rate rose significantly (p = 0.0002), while the accuracy remained relatively constant (p = 0.5). In some cases, a decreased accuracy (subject D) or selection rate (subject B) was reported for the NB method relative to the SWLDA method because of the optimization based on bit rate. In these cases, the optimal bit rate may occur at a lower accuracy, but a much higher selection rate, or vice versa.

Table 3.2 Optimal selection rates, accuracies, and information transfer rates for the five subjects in online trials.

	SR (selections/minute)			ACC (%)			ITR (bits/minute)		
Subject	SWLDA	NB	HMM	SWLDA	NB	HMM	SWLDA	NB	HMM
P	5.52	9.02	10.00	89.29	83.33	95.24	22.79	33.05	46.51
Q	5.52	7.07	8.11	75.00	82.86	90.24	16.98	25.65	34.12
R	3.28	8.10	6.96	94.12	85.37	88.57	14.91	30.93	28.35
S	5.52	11.09	10.90	100.00	83.64	94.44	28.53	40.90	49.86
T	5.52	11.50	11.80	100.00	78.95	93.22	28.53	38.49	52.69
average	5.07	9.35	9.55	91.68	82.83	92.34	22.35	33.80	42.31

Six of the subjects reached 100% accuracy within the 15 sets of flashes using the HMM method and subject D had all characters correct within two sets of flashes. The improvement in ITR from the SWLDA method to the HMM method ranged from 9% (subject N) to 85% (subject I). The average bit rate across subjects improved by 32% from 24.44 to 32.33 (p < 10-8). The selection rate rose from 5.87 to 7.88 (p < 10-5) and the accuracy increased stayed relatively constant (p = 0.35). The HMM method had a significantly higher ITR than the naïve Bayes method (p=0.001), although the optimal selection rates and accuracies were not significantly different (p=0.09 and p=0.43 respectively).

3.3.4. Online Results

In the online experiments, all five subjects were able to select characters with at least 75% accuracy using each of the methods (Table 3.2). Using the SWLDA method, four of the five subjects performed better using the five flash set, resulting in an average selection rate of 5.07, an average accuracy of 91.68%, and an average ITR of 22.35. Two of the five subjects achieved 100% accuracy with both thresholds tested. Using the NB classifier, the average accuracy dropped to 82.83%, but the selection rate rose significantly to 9.35 (p=0.003), resulting in a significantly higher ITR (33.80, p=0.0004).

Table 3.3 Example online output for each of the tested methods. Each is the result of subject S attempting to spell "Heroes in a half shell turtle power" for five minutes. While perfect accuracy was achieved using SWLDA, the user was able to type almost twice as many characters using the HMM algorithm in the same amount of time, resulting in a higher bit rate. HMM* would be the output of the HMM method if the errors were not corrected.

Method	Output
SWLDA	HEROES IN A HALF SHELL TURTLE
NB	HEROE **R**IN A HA**RE** SHELL TURTLE P**RE**ER HER**I**ES IN A**IN**ALF SH
HMM*	**T**EROES IN A HAL **6**HE**R**L TURTLE POWE**D** HEROES IN A HALF**LG**
HMM	HEROES IN A HAL**L** **T**HELL TURTLE POWE**D** HEROES IN A HALF S

Four of the five subjects performed best using the HMM classifier. The average selection rate was 9.55 characters/minute with an accuracy of 92.34%, resulting in an average bit rate of 42.31, which was significantly higher than those achieved using SWLDA (p=0.0003) and NB (p=0.02). On average, the HMM method was able to correct 3 errors per person, which accounted for 47% of the total errors made using this method (Table 3.3).

3.4. Discussion

Most current BCI communication systems do not utilize domain knowledge when processing natural language. Using a hidden Markov model to integrate this information can significantly improve the results of such a system.

In offline analysis, the SWLDA method achieved an average bit rate of 20.07 across subjects before implementing dynamic stopping, which is consistent with previous studies (Townsend et al., 2010; Ryan et al., 2011). Dynamic stopping improved this value 22% on average, which is close to previously reported values (Serby et al., 2005; Speier et al., 2012). Our HMM method improved the bit rate to 32.33 on average, which was 32% higher than when using SWLDA. The offline improvement over the naïve Bayes method was more modest (7.1%), but still statistically and likely clinically significant (p=0.001). It should be appreciated that any improvement in performance can be considered useful because these BCI are intended to be the primary modality of communication for affected patients.

In general, the offline improvements over naïve Bayes seen after optimization are in speed rather than accuracy, which may be counterintuitive. This occurs because the HMM method has a stronger prior probability due to its utilization of all past information. It is therefore able to reach a similar accuracy with a lower confidence threshold, resulting in faster typing speed. When the confidence threshold is held constant in online tests, the typing speed is almost identical, but the accuracy increases due to the improved classifier.

In online analysis, subjects achieved an average ITR of 22.35 when using the SWLDA method. Two of the subjects achieved perfect accuracy using both configurations, indicating the system was not optimized for them as the minimal number of flashes for consistent performance was not reached. Despite the lack of a systematic optimization, the average performance online was superior to the offline performance. Similarly, both the NB and HMM methods demonstrated large improvements in online performance over offline. This could have been due to motivation from receiving feedback or because of randomness due to a small sample size.

The online performance increase of HMM over NB was much larger than that observed in offline analysis (25% for online, 7% for offline). This is likely attributable to the fact that, without error correction, errors compound in the NB method. Drastic errors such as missing a space will change the prior for subsequent characters, resulting in additional errors (Table 3.3). In the HMM method, these errors are corrected, while less significant errors such as switching vowels are often missed. It may be possible to identify which errors are more likely to be caught by the HMM method, allowing users to trust the system to account for some errors while manually correcting those that might be missed. Allowing manual error correction could lead to improved results in one or both of these methods (see future directions).

The ability to retrospectively change prior miscategorized letters automatically could have significant benefits to users. However, some users could also be distracted or discouraged by the presence of an incorrect selection and feel compelled to manually correct it. Similar to other non-BCI typing methods, we believe that a user could learn to trust the system and adapt to a modified task that did not involve correcting errors, which is supported by the success of the subjects in our online study. We also note that the proposed HMM method remains compatible with a backspace option, and if a user is inclined to correct all errors manually, the system reduces to the NB method and is still an improvement over SWLDA.

3.4.1. Limitations and future directions

A standard system and classification method were used in this study, but NLP could be used to integrate domain knowledge into any BCI communication system. Also, the Viterbi method and language model prior can be combined with any classifier that returns a likelihood probability. Future studies can measure the effects of integrating NLP into other BCI systems in conjunction with different classifiers.

Because this method cannot correct all errors automatically, the potential exists for uncorrected errors to exist in the final string. Some of these errors could be fixed by combining this method with an autocomplete method similar to that proposed by Ryan et al. (2011). It is also possible that many errors do not need to be corrected in order for the user's intent to be conveyed as readers are able to understand text that contains errors. Depending on the application of the system, a certain error rate in the final output could be acceptable if it is associated with an increased typing speed. The relationship between error rate in BCI communication output and reader understanding remains to be studied.

The amount of data acquired by the online pilot experiments was limited because of the number of configurations that needed to be tested. The results were therefore subject to significant variability as single errors could result in large changes in bit rate. The optimal threshold values also could not be computed for the online system as only two candidate values could be tested for each method. Finally, the difference in performance when subjects are required to correct errors was not explored. Additional online tests, possibly including multiple sessions with the same user, could provide more optimal configurations of the system and a better evaluation of online performance.

3.4.2. Conclusion

Typing with a P300 system can be modeled as a Markov process that can be indirectly observed through EEG response signals. The Viterbi algorithm effectively incorporates domain information into signal classification, which greatly improves the user's ability to create language. This study shows that incorporating this natural language information significantly improves the performance of a BCI communication system.

4. PARTICLE FILTERS

Character n-gram models are easy to implement as they reduce the state space, allowing the use of dynamic programming algorithms to find optimal classifications. However, they provide a poor representation of natural language, as they ignore context and can give high probability to character strings that do not formulate words. Sentences generated using n-gram language models consist of common character patterns that do not generally make up valid words (Table 4.1). More sophisticated language models could improve accuracy by giving stronger prior probabilities to target characters. These models generally involve an unbounded state space, which makes the use of dynamic programming methods impractical. Stochastic methods such as particle filters can overcome this challenge by estimating prior distributions using sampling.

Table 4.1Examples of 50 character strings of generated text using four language models

Language Model	Example generated text
Uniform	kju705i6gs7nrur 3tpix7uu7c0xjz0o5ogt9hsygp05k7io2y
Unigram	roult ihves4nlcf tsietaakee9swd tst ed tisolpcfgeo
Trigram	whe ford poleselte of ourem becric whout quall ing
Probabilistic Automaton	in as how allowances group away or one besides wid

In this work, English words are modeled using a probabilistic automaton (Mohri, 1996) and probability distributions are estimated using sequential importance resampling, a common particle filtering (PF) algorithm (Gordon et al., 1993). These probability distributions are used as priors in a Bayesian process model to classify EEG signals in the P300 speller system. This method was compared offline with a standard stepwise linear discriminant analysis (SWLDA) method with dynamic stopping as well as a previously presented Hidden Markov Model classifier (HMM) on a previously published data set consisting of 15 healthy subjects (Speier et al., 2014a). Prospective evaluation was then performed online using the HMM and PF algorithms.

4.1. Probabilistic Automaton

The probabilistic automaton models the English language by creating states for every substring that starts a word in the corpus. Thus, the word “the” would result in three states: “t,” “th,” and “the.” The start state corresponds to a blank string. Each state then links to every state that has a string that is a superstring that is one character longer. Thus, the state “t” will link to the states “th” and “to.” (Fig. 4.1) States that represent complete words contain links back to the root node to begin a new word.

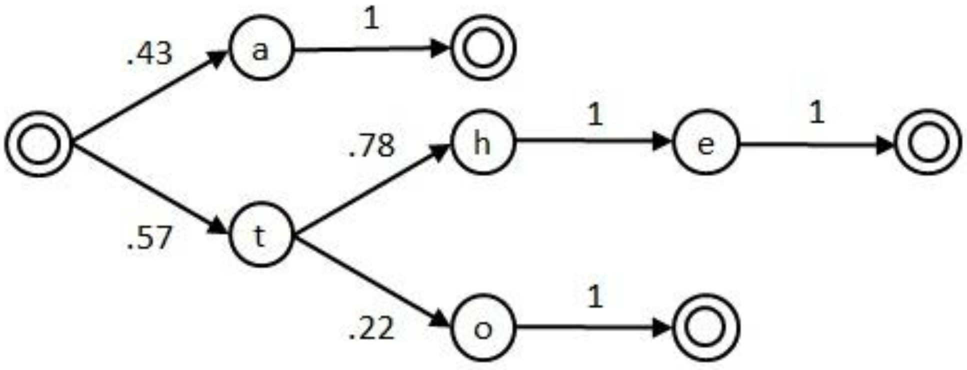


Figure 4.1 Example automaton for a reduced vocabulary consisting only of the words “a,” “the,” and “to.” Double circles represent possible termination states for a word. These states link back to the root node to represent the beginning of a subsequent word.

Similar to the trigram model, the transition probabilities are determined by the relative frequencies of words starting with the states' substrings in the Brown English language corpus (Francis and Kucera, 1979).

$$p(x_t|x_{0:t-1}) = \frac{c(x_0, \dots, x_{t-1}, x_t)}{c(x_0, \dots, x_{t-1})}$$

where $c('a','b')$ denotes the number of occurrences of a word that starts with the string “ab” in the corpus. Similarly, the probability that a word ends and the model transitions back to the root is the ratio of the number of occurrences of complete words consisting of a string to the total number of occurrences of words beginning with that string.

$$p(' '|x_{0:t-1}) = \frac{c(x_0, \dots, x_{t-1}, ' ')}{c(x_0, \dots, x_{t-1})}$$

where $c(a, b, ' ')$ is the number of occurrences of the word “ab” in the corpus.

4.2. Witten Bell Smoothing

Unlike trigram models, word models do not represent words that do not occur in the training corpus. As long as a word is comprised of character patterns that occur in the corpus, a trigram model will give it positive probability, regardless of whether the actual word occurred (Table 4.2). In the probabilistic automaton model, any string that does not occur in the corpus is given a 0 probability, regardless of whether it is a valid word. Thus, it is impossible for a system using this model to generate any word that does not occur in the corpus. In common communication, this is generally not a problem as a large corpus will usually contain all common words in a language. Problems can still occur, however, particularly in cases where a word or phrase may be common for a user, but not in general language, such as the names of people and places in the user's life.

Table 4.2 String counts in the Brown corpus. Component trigrams for the string "_viral_" all occur in the corpus, which results in a non-zero probability in the trigram model, despite the word never occurring.

String	Count	Probability
__v	6569	0.01
_vi	2063	0.31
vir	313	0.03
ira	264	0.02
ral	3018	0.12
al_	14099	0.38
viral	0	0

Witten Bell is a method for "smoothing" the probability distribution to give non-zero probabilities to words that have not occurred in the corpus. In this method, the system combines the model with a more general probability estimate (trigrams in this case) to give some probability to cases that do not appear in the corpus.

$$p(x_t|x_{0:t-1}) = \lambda p(x_t|x_{0:t-1}) + (1-\lambda)p(x_t|x_{t-1}, x_{t-2})$$

The assumption is made that the probability of a new observation given the observed history $x_{0:t-1}$ is higher if there have been many different previous observations relative to the frequency of observing that history. The value of λ is thus said to be:

$$\lambda = \frac{c(x_{0:t-1})}{c(x_{0:t-1}) + T(x_{0:t-1})}$$

where $T(x_{0:t-1})$ is the number of distinct characters that follow $x_{0:t-1}$ in the corpus and $c(x_{0:t-1})$ is the total number of occurrence of $x_{0:t-1}$ in the corpus. In the case where $x_{0:t-1}$ does not occur in the corpus, λ is undefined and the system falls back to the trigram probability. After smoothing, the transition probabilities are defined as:

$$p(x_t|x_{0:t-1}) = \begin{cases} \dfrac{c(x_{0:t}) + T(x_{0:t-1})p(x_t|x_{t-1}, x_{t-2})}{c(x_{0:t-1}) + T(x_{0:t-1})} & c(x_{0:t-1}) > 0 \\ p(x_t|x_{t-1}, x_{t-2}) & c(x_{0:t-1}) = 0 \end{cases}$$

where $T(x_{0:t-1})$ represents the number of distinct characters that occur in the corpus after the string "$x_0 x_1 \dots x_{t-1}$" and $p(x_t|x_{t-1}, x_{t-2})$ is the trigram probability as described in chapter 2. This distribution gives a small amount of probability to strings that did not occur in the corpus, allowing for typing of out of vocabulary words (Table 4.3). This probability is generally small, which is appropriate due to the fact that it represents a word that did not occur in this history observed text. The probability of an out of vocabulary word will increase with the rarity of the observed text and the number of different observed options for completing a string.

Table 4.3 Smoothed probability for the letter 'a' in the word "viral," which did not occur in the corpus.

Expression	Value
$c(\text{"\_vira"})$	0
$c(\text{"\_vir"})$	254
$T(\text{"\_vir"})$	6
$p(\text{"a"} \mid \text{"ir"})$	0.02
$p(\text{"a"} \mid \text{"\_vir"})$	0.0005

4.3. Sequential Importance Resampling

When the system begins, a set of P particles is generated and each is associated with the root node. At the start of a new character, samples ${x_t}^{(j)}$ are drawn from the proposal distribution defined by the transition probabilities from the previous state. When a particle reaches a terminal node for the current word, it transitions to the root node of the model to begin typing a new word. The history for each particle, ${x_{0:t}}^{(j)}$, is stored to represent the output sequence associated with that particle.

$$ {x_t}^{(j)} \sim \pi\left(x_t \middle| {x_{0:t-1}}^{(j)}\right) $$

In this case, the proposal distribution is defined as the prior probability of state x_t given the history, $p\left(x_t \middle| {x_{0:t-1}}^{(j)}\right)$. After each stimulus response, the probability weight is computed for each of the particles

$$ {w_t}^{(j)} \propto \frac{p\left({x_t}^{(j)} \middle| {x_{0:t-1}}^{(j)}\right) p\left(y_t \middle| {x_t}^{(j)}\right)}{\pi({x_t}^{(j)}|{x_{0:t-1}}^{(j)})} \propto \prod_i f\left(y_t^i \middle| {x_t}^{(j)}\right) $$

The weights are then normalized and the probability of the current character is found by summing the weights of all particles that end in that character.

$$ p(x_t|y_{0:t}) = \sum_k {w_t}^{(k)} \delta_{x_t}^{{x_t}^{(k)}} $$

where δ is the Kronecker delta. If the maximum probability is above a threshold, the particle with the maximum weight is selected and its history is used as the output text. New particles are then resampled from the weight distribution and the system moves on to the next character.

The main concern with using this method is the number of particles to use. Using more particles increases the processing necessary for estimating the distributions. However, a low number of particles could result in undersampling the distributions and missing important possible sequences. Sensitivity analysis is performed on the number of particles by determining the offline performance using 10, 100, 1,000, 10,000 and 100,000 particles.

4.4. Validation

4.4.1. Offline Dataset

The offline dataset for this project was previously described in chapter 3 (Speier et al., 2014a). The subjects in the offline dataset were 15 healthy graduate students and faculty with normal or corrected to normal vision between the ages of 20 and 35.Only one subject (subject F) had previous experience using a BCI for typing. The system used a 6 x 6 character grid, row and column flashes, and an interstimulus interval (ISI) of 125 ms. Each subject underwent between 8 and 10 trials consisting of spelling a five letter word with 15 sets of 12 flashes (six rows and six columns) for each letter. Three analysis methods were compared: SWLDA, HMM, and PF. The choice of target words for this experiment was independent of the trigram language model used in the HMM and PF methods.

4.4.2. Protocol

The subjects for the online study consisted of 15 healthy volunteers with normal or corrected to normal vision between the ages of 20 and 30. The electrode set consisted of a reduced set of four channels (PO8, PO7, POz, and CPz) (Speier et al., 2014b). The training sessions for these subjects consisted of three sessions of copy spelling 10 character phrases. Each subject then chose a target phrase to spell in online sessions. In each session, the subject had five minutes to spell as much of the phrase as they could using one of the three analysis methods: SWLDA (with dynamic stopping), HMM, or PF. Subjects were instructed not to correct errors and to repeat the phrase if they completed it in under five minutes.

BCI2000 (Schalk et al., 2004) was used for data acquisition and online analysis. Offline analysis was performed using MATLAB (version 7.10.0, MathWorks, Inc, Natick, MA).

4.4.3. Offline Performance

When using SWLDA in offline analysis, all subjects were able to type with varying levels of performance. The best performer (subject D) was able to achieve 89% accuracy at a rate of 9.96 selections per minute, while the worst performer (subject C) achieved an accuracy of 80% at a rate of only 3.96 selections per minute (Table 4.4, Fig. 4.3). The accuracy increased with the number of flashes for all subjects and 12 of the 15 were able to exceed 90% accuracy within 15 sets of flashes.

Table 4.4 Optimal selection rates, accuracies, and information transfer rates for the 15 subjects in offline trials.

	SR (selections/minute)			ACC (%)			ITR (bits/minute)		
Subject	SWLDA	HMM	PF	SWLDA	HMM	PF	SWLDA	HMM	PF
A	8.07	9.80	10.74	93.33	95.56	97.78	36.13	45.87	51.33
B	5.33	7.42	7.33	88.89	88.89	88.89	21.82	30.38	30.05
C	3.96	8.45	6.71	80.00	67.50	85.00	13.54	21.91	25.43
D	9.96	10.83	11.75	88.89	97.78	95.56	40.82	53.08	54.99
E	3.98	5.20	6.54	95.56	97.78	91.11	18.64	25.47	27.98
F	7.21	8.99	10.63	95.56	95.56	95.56	33.76	42.09	49.74
G	5.33	6.56	9.33	94.00	92.00	92.00	24.18	28.60	40.64
H	8.66	10.17	9.77	88.00	90.00	96.00	34.86	42.59	46.13
I	5.08	9.38	10.68	78.00	78.00	80.00	16.68	30.77	36.55
J	4.78	5.86	6.55	90.00	94.00	98.00	20.03	26.57	32.28
K	3.98	6.68	7.81	84.00	80.00	84.00	14.80	22.85	29.01
L	5.62	7.36	7.71	92.00	90.00	98.00	24.47	30.80	37.98
M	4.13	6.19	6.16	82.00	84.00	90.00	14.72	22.98	25.82
N	7.60	8.98	9.58	94.00	90.00	100.00	34.48	37.62	49.55
O	4.40	6.28	9.43	88.00	84.00	82.00	17.71	23.33	33.62
Average	5.87	7.88	8.70	88.82	88.34	91.59	24.44	32.33	38.07

Six of the subjects reached 100% accuracy within the 15 sets of flashes using the HMM method and subject D had all characters correct within two sets of flashes. The improvement in ITR from the static method to the HMM method ranged from 40% (subject N) to 100% (subject I). The average bit rate across subjects improved by 32% from 24.44 to 32.33 ($p < 10^{-8}$). The selection rate rose from 5.87 to 7.88 ($p < 10^{-5}$) and the accuracy increased stayed relatively constant (p = 0.35).

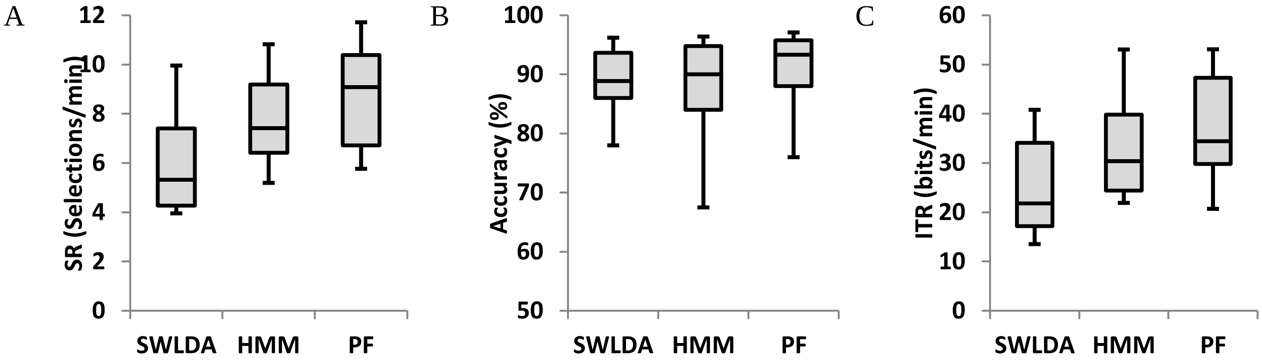


Figure 4.2 Boxplots of the selection rate, accuracy, and bit rate distributions among the 15 subjects using the SWLDA, HMM, and PF classification algorithms in offline analysis.

Using the particle filter, 11 of the 15 subjects reached 100% accuracy within the 15 sets of flashes. The average bit rate rose significantly using this method from 32.33 to 38.07 (p=0.00001), with increases of at least five bits/min for nine of the 15 subjects. The average selection rate rose significantly over the HMM method from 7.88 to 8.70 (p=0.01) and the accuracy showed significant improvement from 88.34% to 91.59% (p=0.02).

4.4.4. Sensitivity Analysis

Using 10 particles, the average ITR was 12.34 bits/minute as no subject achieved an accuracy above 50% (Fig. 4.2). Progressively changing the number of particles to 100, 1,000, and 10,000 resulted in significant improvements in average ITR value: 33.34 (p<10-8), 36.79 (p<10-4), and 38.07 (p=0.01), respectively. Increasing the number of particles to 100,000 did not result in a significant increase (ITR=38.21; p=0.29).

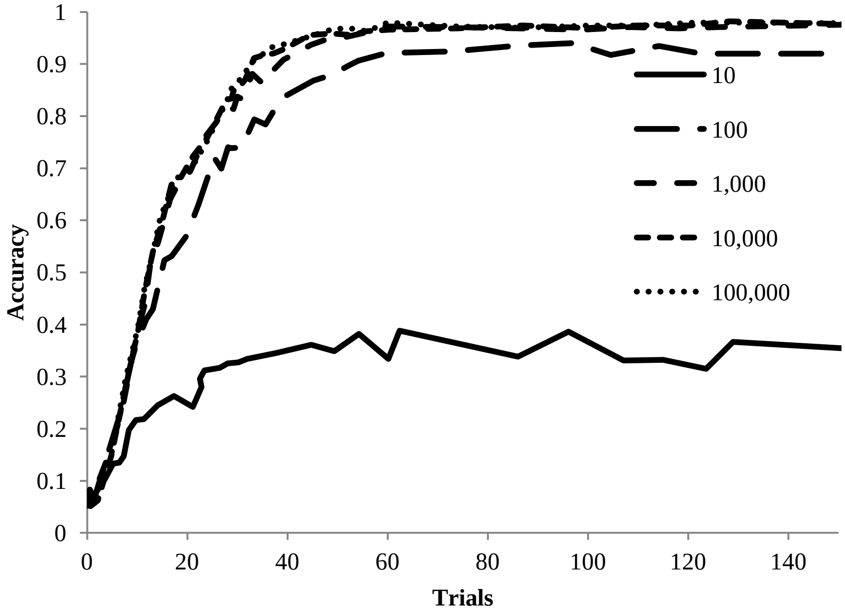


Figure 4.3 Analysis of the sensitivity of offline results to the number of particles used. Optimal classifier performance was achieved when using at least 10,000 particles.

4.4.5. Online Performance

In online experiments, all 15 subjects were able to type characters with at least 65% accuracy using each of the algorithms (Table 4.5). Using the HMM method, 11 of the 15 subjects achieved at least 80% accuracy and 6 characters per minute. All subjects selected characters with at least 74% accuracy using the PF method, with 13 of 15 subjects selecting over seven characters per minute on average. One subject (subject P) had substantially lower results than the rest of the data set, selecting fewer than three characters per minute on average with accuracy under 75% for both methods.

Table 4.5 Online selection rates, accuracies, and information transfer rates for each subject using the hidden Markov model and particle filtering algorithms.

	SR (selections/min)		ACC (%)		ITR (bits/min)	
Subject	HMM	PF	HMM	PF	HMM	PF
A	8.07	10.74	93.33	97.78	36.13	51.33
B	5.33	7.33	88.89	88.89	21.82	30.05

C	3.96	6.71	80.00	85.00	13.54	25.43
D	9.96	11.75	88.89	95.56	40.82	54.99
E	3.98	6.54	95.56	91.11	18.64	27.98
F	7.21	10.63	95.56	95.56	33.76	49.74
G	5.33	9.33	94.00	92.00	24.18	40.64
H	8.66	9.77	88.00	96.00	34.86	46.13
I	5.08	10.68	78.00	80.00	16.68	36.55
J	4.78	6.55	90.00	98.00	20.03	32.28
K	3.98	7.81	84.00	84.00	14.80	29.01
L	5.62	7.71	92.00	98.00	24.47	37.98
M	4.13	6.16	82.00	90.00	14.72	25.82
N	7.60	9.58	94.00	100.00	34.48	49.55
O	4.40	9.43	88.00	82.00	17.71	33.62
Average	8.05	8.64	83.74	89.70	30.69	37.31

In this study, 12 of 15 subjects achieved a higher bit rate when using the PF classifier than when using the HMM method. On average, subjects selected 8.05 characters per minute with 83.74% accuracy, resulting in an average bit rate of 30.69 bits/minute using the HMM algorithm (Fig. 4.4). When using the PF algorithm, subjects achieved significant improvements with an average selection rate of 8.64 characters/minute (p=0.001), an average accuracy of 89.70 (p=0.01), and an average bit rate of 37.31 (p=0.003).

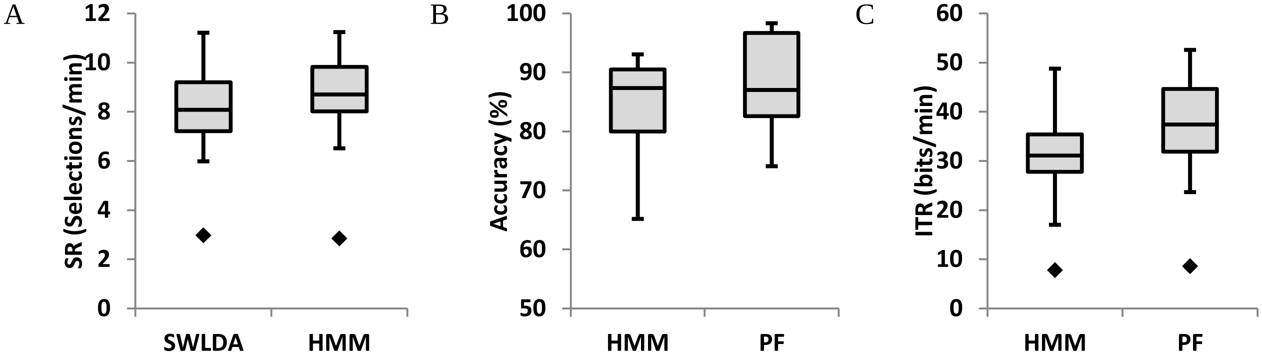


Figure 4.4 Boxplots of the selection rate, accuracy, and bit rate distributions among the 15 subjects using the HMM and PF classification algorithms in online analysis.

The particle filtering algorithm successfully corrected 6.8 errors on average for each subject (Table 4.6). These corrections accounted for 49% of the classification errors in the initial classifications. All subjects had at least two errors corrected with correction rates varying between 20% (subject AA) and 100% (subject V). Automatic error correction was responsible for increasing classifier accuracy from 82.90% to 89.70% and the average bit rate from 31.55 to 37.31.

Table 4.6 Example online output for each of the tested methods. Each row is the result of subject Q attempting to spell "I want to be the very best like no one ever was to catch them is my real test" for five minutes. HMM* and PF* are the outputs of the two algorithms without error correction.

Method	Output
TARGET	I WANT TO BE THE VERY BEST LIKE NO ONE EVER WAS TO CATCH THEM IS MY REAL TEST
HMM*	I WANT T**E** BE**R**THE VERY BE**GN** LIKE **HEL**ONE **Q**VER**E**WAS T**A** C
HMM	I WANT TO BE**R**THE VERY BE**GN** LIKE **HEL**ONE EVER**E**WAS TO C
PF*	**CF**WANT TO BE THE **W**ER**E** BEST**S**LIKE NO ON**CH**E**S**ER WAS T**P** CA
PF	I WANT TO BE THE **W**ER**E** BEST LIKE NO ONE EVER WAS TO CA

4.5. Discussion

The particle filter required fewer samples and made more accurate selections than the standard classification and HMM methods. This improvement is due to the improved language model that biases selections towards English words rather than simply common character patterns (Table 4.1). Because n-gram models only use a limited character history, they give high probability to strings that resemble correct patterns locally, which do not necessarily make sense in context (Table 4.7). Limitations in the language model result in incorrect prior distributions, which can mislead the classifier or cause the threshold probability to be met before sufficient observations were made.

Table 4.7 String counts in the Brown corpus. Component trigrams for the string "_ing_" are common in the corupus, which results in a high probability in the trigram model, despite the exact string rarely occurring.

String	Count	Probability
__i	68440	0.07
_in	33783	0.49
ing	30454	0.34
ng_	30035	0.78
ing	1	~0

Online performance was consistent with the results from offline analysis, with only a small decrease in average performance. This decrease is largely due to a single subject (subject P) as the average bit rates of the remaining subjects (32.32 bits/min for HMM and 39.37 bits/min for PF) are almost identical to the offline results (32.33 bits/min and 38.07 bits/min, respectively). A decrease is not surprising as online studies did not optimize the probability threshold and only used four electrodes, while the offline analysis contained a full set of 32 channels. These factors may have been offset by added user motivation resulting from feedback and free spelling as well as a reduction in fatigue afforded by the shorter setup time resulting from the reduced number of electrodes.

Several subjects saw modest or no improvement over the HMM method. In general, the errors that these subjects saw were consistent with the language model. For instance, a typo that changes “UNITS” into “UNITY” cannot be solved by a single word language model as both are valid words. In this case, context would need to be incorporated into the system to truly determine the target character. For instance, previous words can help to determine the most likely part of speech of the current target word. This information could be used to change the probabilities in the automaton, so that the model reflects the appropriate subset of the corpus.

Variance between subjects increased in online trials because allowing subjects to select the target sentence allowed for target strings that were not well represented by the training corpus for the language model. When implementing this system with “locked-in” patients, a targeted corpus could be developed that models likely words or phrases for a patient rather than a generic model of the English language. Such a model could also adapt based on context such as time of day or the subject’s environment. The model would then provide a stronger prior probability to the particle filtering algorithm, resulting in faster selections and more accurate automatic error correction. Developing dynamic and targeted language models for BCI application remains as future work.

The performance of the particle filter algorithm was shown to be reliant on a sufficient number of particles as the algorithm failed to accurately classify characters for any subject when using only 10 particles. This is not surprising as undersampling the posterior distribution will not accurately reflect the true distribution, which can have highly volatile results. However, the algorithm proved fairly robust, as it was able to achieve good classification accuracy with as few as 100 particles and its results stabilized after increasing the number to 10,000. While the complexity of the algorithm is linear in the number of particles, it is important to limit the number used because of the short duration of a time step (125 ms) in the online system. In this instance, 10,000 is a reasonable number for online computation, but more particles will be needed as the complexity of the language model increases. Future implementations should be wary of the number of particles needed for a sufficient representation of the posterior distribution and the effect that will have on the performance of the classifier in a real-time setting.

4.5.1. Limitations and future directions

Final output strings from the particle filtering algorithm could contain errors as it is not able to make all corrections automatically. Some errors are obvious to a reader and are unlikely to affect the ability of the user to convey intent. However, in some cases a small error can change the meaning of a sentence or make output incomprehensible. To handle these cases, the user can be given the option to make manual corrections in scenarios where the system is unlikely to be able to make a correction automatically. Combining this algorithm with an autocomplete method could also allow the user to fix some of these errors without sacrificing speed (Ryan et al., 2011). Studying the relationship between error rate in BCI output and reader understanding could provide insight into the impact of misclassifications and optimal strategies for correcting errors.

This study was conducted using healthy volunteers who did not have the same constraints as “locked-in” patients, such as restrictions to eye gaze. As a result, studies implemented in the target patient population are likely to yield lower bit rates and are also likely to be more variable as patients will have differences in severity of disease. The particle filter algorithm is expected to have a similar improvement for these subjects as it does not change the front end of the system that is presented to the user. Therefore, the signal quality is not affected and the classification improves by incorporating external language domain knowledge. Performance using the P300 speller system with language domain knowledge needs to be tested in the target patient population in order to measure the true effect on performance.

4.5.2. Conclusion

Typing with a P300 system can be modeled as a Bayesian process that can be indirectly observed through EEG response signals. Stochastic importance sampling effectively incorporates domain information into signal classification, which greatly improves a user's ability to create language. This study shows that incorporating this natural language information significantly improves the performance of a BCI communication system.

5. EVALUATION

Given the number and variety of approaches to optimizing the P300 speller, a reliable metric is important for evaluation and comparison across experimental paradigms and ultimately across studies, which to date is lacking. A useful metric must consider the amount of time taken, the accuracy of selections, and the tradeoff between the two. Increasing the amount of data and therefore time needed to make a decision can increase the accuracy of the selection at the expense of system speed. Perfect accuracy however is not always necessary as a BCI can integrate prior knowledge about the domain and common user behavior to understand output despite errors. In the case of typing natural language, for instance, text is often readable despite the presence of typos. In a non-typing context, errors may not be permissible, so errors must be corrected by subsequent “undo” selections, which would result in a perfect accuracy, but slower typing speed.

Information Transfer Rate (ITR) is a general evaluation metric devised for BCI systems that determines the amount of information that is conveyed by a system’s output (McFarland et al., 2003). The metric is appealing for several reasons: it is derived from information theory principles, it combines the competing statistics of speed and accuracy, and it reduces to an information transfer problem that can be compared across applications (Pierce, 1980). However, ITR is not appropriate for evaluating language systems because it makes two assumptions that are incorrect in general, particularly in the language domain: 1) that all possible selections are equally probable and 2) that systems are memoryless. Several methods have since been introduced in attempt to reduce the adverse attributes of ITR. Word Symbol Rate (WSR) normalizes ITR by its maximum value and then scales down based on error rate (Furdea et al., 2009). Practical Bit Rate (PBR) finds the theoretical bit rate if the user had corrected every selection error (Townsend et al., 2010). Characters per Minute (CPM) calculates the theoretical number of characters correctly typed after error correction (McFarland et al., 2011). Output Characters per Minute (OCM) is an online metric similar to CPM that requires all errors to be corrected (Ryan et al., 2011). In general, these metrics all depend on aspects that are system specific and therefore not generalizable (see methods).

A standard method for evaluating results does not exist, making it difficult to compare the relative value or the superiority of different experimental paradigms or approaches. We present an information rate metric (MI_n) based on mutual information designed to incorporate language domain knowledge to more accurately measure the utility of language-based BCI systems. Three versions of this metric are compared to five existing methods that are currently used for evaluation in P300 literature. We use each metric to optimize the dataset used by Speier et al. (2012) to show the difference in performance achieved. We then reevaluate the results of 11 published studies using the existing metrics used in the literature and compare the results to those determined using the proposed metrics. We cannot retroactively account for differences in system parameters and experimental paradigms, so it is impossible to make fair comparisons between studies. However, we show the effects of choosing various evaluation metrics on comparisons made within studies and the conclusions that result. Our analysis shows that the selection of a metric significantly affects system optimization as well as the evaluation of different approaches for BCI communication, leading to the necessity for adopting a consistent and reliable performance metric.

5.1. Previously Published Metrics

5.1.1. Information Transfer Rate

ITR finds the average bits of information contained in each selection and then divides by the time required to make the selection. The first assumption made by this method is that each selection is independent, so that no information for a selection can be acquired from previous selections. The bits per symbol, B, is then simply the mutual information between the chosen symbol, Y, and the desired symbol, X.

$$B = I(X;Y) = H(Y) - H(Y|X)$$

where $I(X;Y)$ is mutual information, $H(Y)$ is the marginal entropy of Y, and $H(Y|X)$ is the conditional entropy of Y given X. It is assumed that the conditional probability of the selection depends only on whether x is the correct symbol and errors are uniform over the remaining characters.

$$p(y|x) = \begin{cases} P & x = y \\ \dfrac{1-P}{N-1} & \text{x} \neq y \end{cases}$$

Using this assumption, the conditional entropy becomes

$$H(Y|X) = -\sum_{x} p(x) \sum_{y} p(y|x) \log p(y|x) = -P \log P - (1-P) \log \frac{1-P}{N-1}$$

The assumption is then made that the marginal probabilities $p(x)$ and $p(y)$ are uniform over the characters in the grid (i.e., $p(x) = p(y) = \frac{1}{N}$). The marginal entropy then becomes

$$H(Y) = -\sum_{y} p(y) \log p(y) = \log N$$

The value for bits per symbol is then

$$B = \log N + P \log P + (1 - P) \log \frac{1 - P}{N - 1}$$

ITR is then the bits per symbol divided by the average time required to select a single symbol, T.

$$ITR = B/T$$

The theory behind ITR was derived from the concept of a noisy channel with $1 - P$ representing the error frequency in the output string. Instead, BCI literature generally uses P as the selection accuracy. In some systems, this is equivalent, but it is not in cases where multiple steps are used for one selection or where backspaces can be used to correct errors. In these cases, counterintuitive results can occur where two users can type the same string without errors and one can have a slower speed, but a higher ITR (Fig. 5.1).

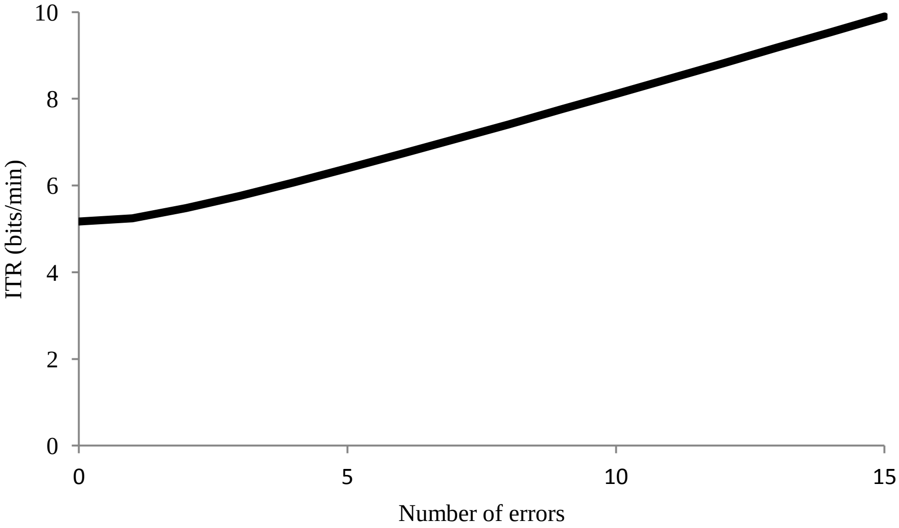


Figure 5.1 ITR calculation for hypothetical cases of typing a 10 character sequence with error correction in 10 minutes. For each error, two additional selections are required. As a result, the ITR increases because the increase in number of selections more than offsets the decrease in selection accuracy.

5.1.2. Word Symbol Rate

To calculate WSR, the bits per symbol from 5.1.1 are first scaled by their maximum possible value. The result is called symbol rate (SR).

$$SR = \frac{B}{\log N}$$

SR is treated as the probability of a correct selection. The average number of selections necessary to choose one character is then found by determining the number of additional selections required for correcting errors. Errors occur at a probability of (1-SR) and require an additional selection for correction. Each of these correcting selections also has a probability of (1-SR) of being incorrect, which then also need to be corrected. Averaging all of these correcting selections leads to the sum of a geometric series.

$$1 + 2(1 - SR) + 2(1 - SR) * 2(1 - SR) + \cdots = \sum_{i=0}^{\infty} \big(2(1 - SR)\big)^i = \frac{1}{2SR - 1}$$

The average number of characters typed in a single selection is then the reciprocal, $2\mathrm{SR} - 1$. The argument is made that if an average selection provides less than half the maximal amount of information, then there will be more errors than correct selections, so WSR becomes,

$$WSR = \begin{cases} \dfrac{2SR - 1}{T} & SR > 0.5 \\ 0 & SR \leq 0.5 \end{cases}$$

5.1.3. Practical Bit Rate

Similar to WSR, PBR assumes that the user will correct all typing errors. However, instead of using SR the actual typing accuracy is used. This yields the same geometric series

$$1 + 2(1 - P) + 2(1 - P) * 2(1 - P) + \cdots = \sum_{i=0}^{\infty} \big(2(1 - P)\big)^i = \frac{1}{2P - 1}$$

This metric then divides the bits of information contained in a single correct selection (still assuming all characters have the same probability of being chosen) by the average number of selections required to choose that character,

$$(2P - 1)\log N$$

Here, subjects with selection accuracy below 50% would make errors at a faster rate than they would be able to correct them, so the bit rate becomes,

$$PBR = \begin{cases} \frac{(2P - 1)\log N}{T} & P > 0.5 \\ 0 & P \leq 0.5 \end{cases}$$

Practical bit rate has also been computed substituting ITR for $\log N$ (Jin, 2010). Because both PBR and ITR include penalties for incorrect selections, this metric will double count errors, resulting in an overly conservative estimate of bit rate.

5.1.4. Characters per Minute

CPM builds off of PBR as it uses the same correction factor to account for additional selections required to correct errors. It differs in that it does not take matrix size into account and instead calculates the number of characters selected per minute.

$$CPM = \begin{cases} \frac{2P - 1}{T} & P > 0.5 \\ 0 & P \leq 0.5 \end{cases}$$

5.1.5. Output Characters per Minute

OCM is a similar metric to CPM, but is only possible in online implementations where all errors have been corrected. OCM is computed simply by dividing the total number of characters in the typed string by the time required to type it. Because all errors are corrected, the accuracy is necessarily 100%, so no correction is required.

5.2. Proposed Metrics

We propose an alternate mutual information-based metric that has similar benefits to ITR, but does not rely on the same assumptions. Three versions are included that progressively remove assumptions, resulting in increasingly accurate representations of the true bit rate. The first version, MI_0, removes the uniform character probability assumption and instead uses relative character frequency. The second version, MI_n, removes the assumption of independent selections by incorporating knowledge about the previous characters using an n-gram language model. The third version, MI_ne, uses an error model to incorporate additional information contained in incorrect selections.

5.2.1. MI_0

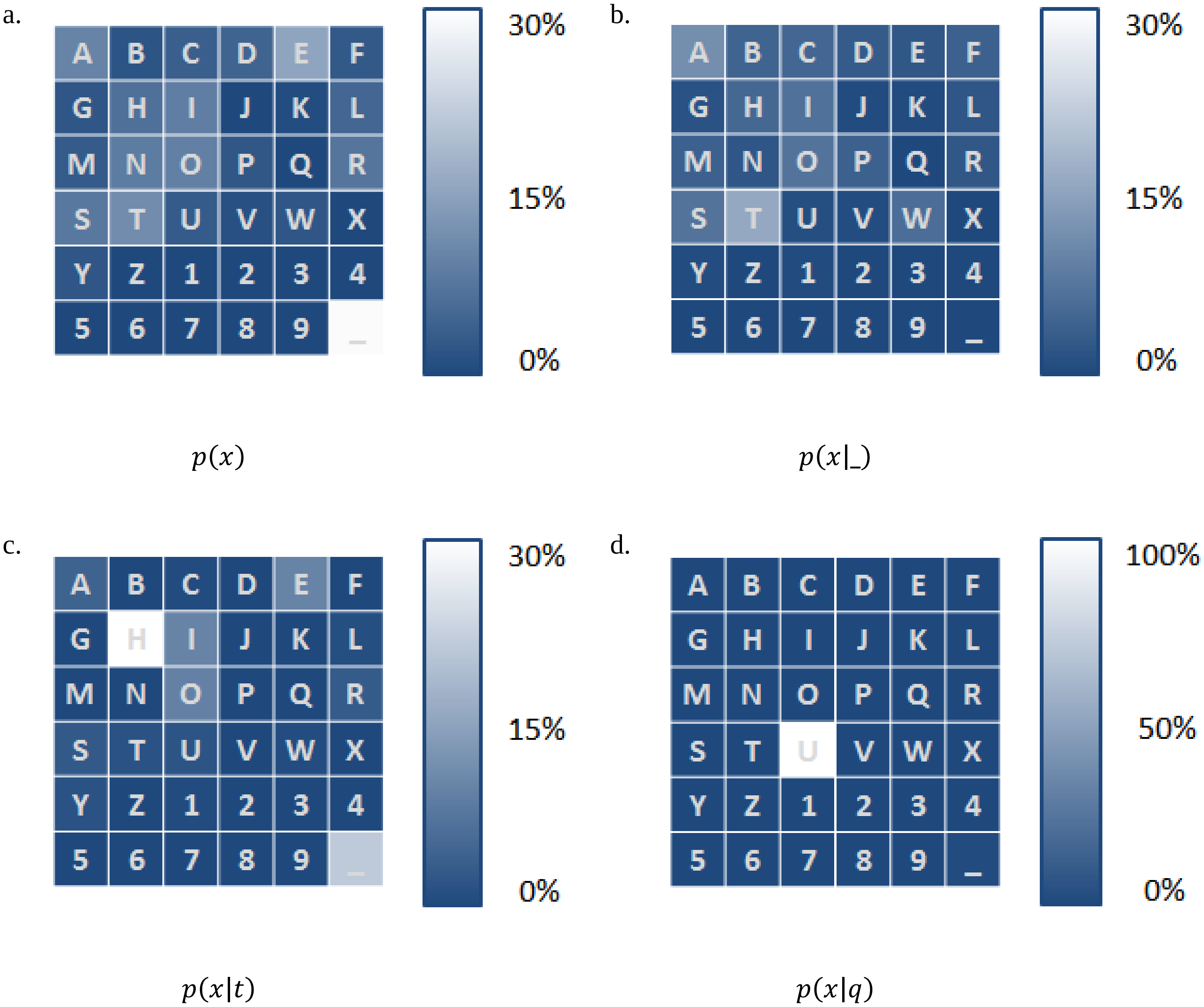
With this method, the system remains memoryless (i.e., all selections are assumed independent) and all errors are still assumed to be uniform over all incorrect characters. MI_0 is simply the mutual information between the target symbol, Y, and the selected symbol, X, as in ITR.

$$B_0 = I(X;Y) = H(Y) - H(Y|X)$$

We maintain the assumption that errors are uniform over all incorrect characters. The conditional entropy, $H(Y|X)$, is then unchanged from ITR. However, we remove the assumption that all characters are equally probable and instead determine their probabilities by their relative frequencies in the general purpose Brown corpus (Francis and Kucera, 1979) (Fig. 5.2) as

$$p(x) = \frac{c(x)}{c(*)}$$

where $c(x)$ is the number of occurrences of character x in the corpus and $c(*)$ is the number of characters in the corpus. The probability of selecting character y can then be found.



$p(x)$ $p(x|\_)$ $p(x|t)$ $p(x|q)$

Figure 5.2 Marginal probability of characters in the English language (a) and conditional probabilities of characters given previous characters of space (b), 't' (c), and 'q' (d).

$$p(y) = \sum_x p(y|x)p(x) = \frac{1-P}{N-1} + \frac{NP-1}{N-1} p(X=y)$$

The marginal entropy may then be computed.

$$H(Y) = -\sum_x \left(\frac{1-P}{N-1} + \frac{NP-1}{N-1} p(x)\right) \log\left(\frac{1-P}{N-1} + \frac{NP-1}{N-1} p(x)\right)$$

Combining these equations yields the bits per symbol, and dividing by the average time yields MI_0.

$$MI_0 = B_0/T$$

5.2.2. MI_n

MI_n builds on MI_0 by removing the assumption that all character selections are independent. We assume that selected characters are directly dependent on the respective target characters and that target characters are dependent on the previous n characters (Fig. 5.3).

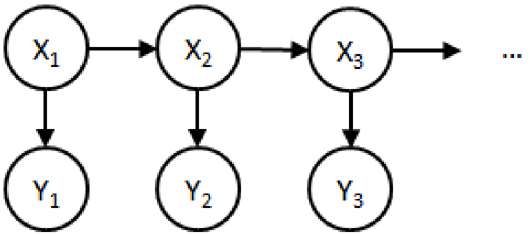


Figure 5.3 Causal graph for dependence between target characters x_i and selected characters y_i when using 1 step of history MI_1.

The bits per symbol is then the mutual information between the target symbol, Y, and the selected symbol, X conditioned on n steps of history. Under the Markov assumption, only one step of history needs to be factored into the calculation.

$$B_1 = I(X; Y|X_{-1}, \dots, X_{-n}) = H(Y|X_{-1}, \dots, X_{-n}) - H(Y|X, X_{-1}, \dots, X_{-n})$$

The conditional entropy $H(Y|X, X_{-1})$ is unaffected because Y is conditionally independent of past selections given X. The only change is then the marginal entropy.

$$H(Y|X_{-1}, \dots, X_{-n}) = -\sum_{x_{-1}, \dots, x_{-n}} p(x_{-1}, \dots, x_{-n}) \sum_{y} p(y|x_{-1}, \dots, x_{-n}) \log p(y|x_{-1}, \dots, x_{-n})$$

We can find the conditional probability $p(y|x_{-1})$ by marginalizing over x:

$$p(y|x_{-1},\dots,x_{-n}) = \sum_{x} p(y|x)p(x|x_{-1},\dots,x_{-n}) = \frac{1-P}{N-1} + \frac{NP-1}{N-1} p(X = y|x_{-1},\dots,x_{-n})$$

Resulting in a marginal entropy of

$$H(Y|X_{-1},\dots,X_{-n}) \\ = -\sum_{x_{-1},\dots,x_{-n}} p(x_{-1},\dots,x_{-n}) \sum_{x} \left(\frac{1-P}{N-1} \right. \\ \left. + \frac{NP-1}{N-1} p(x|x_{-1},\dots,x_{-n})\right) \log\left(\frac{1-P}{N-1} + \frac{NP-1}{N-1} p(x|x_{-1},\dots,x_{-n})\right)$$

The conditional probabilities $p(x|x_{-1},\dots,x_{-n})$ can be found by finding the fraction of occurrences of character x_{-1} that are followed by x in the corpus.

$$p(x|x_{-1},\dots,x_{-n}) = \frac{c(x_{-n},\dots,x_{-1},x)}{c(x_{-n},\dots,x_{-1})}$$

In general, the previous target character, x_{-1} is not known, but y_{-1} is, so

$$p(x|y_{-1},\dots,y_{-n}) = \sum_{x_{-1},\dots,x_{-n}} p(x|x_{-1},\dots,x_{-n}) \frac{p(x_{-1},\dots,x_{-n}) \prod_{i=1}^{n} p(y_{-i}|x_{-i})}{\sum_{\langle x'_{-1},\dots,x'_{-n}\rangle} p(x'_{-1},\dots,x'_{-n}) \prod_{i=1}^{n} p(y_{-i}|x'_{-i})}$$

would be more appropriate than $p(x|x_{-1},\dots,x_{-n})$. There are some cases where $p(x|x_{-1},\dots,x_{-n})$ is appropriate such as in the beginning of typing, but in general it is used for simplicity. As the classification accuracy of the system increases, $p(x|y_{-1},\dots,y_{-n})$ converge to $p(x|x_{-1},\dots,x_{-n})$, so this estimate is reliable when the error rate is low, but underestimates information contained in low accuracy systems.

These equations can be combined to determine the bits per symbol. Knowledge from additional steps can be factored into this equation by conditioning over previous targets and summing over their possible values.

$$B_n = I(X;Y|X_{-1},\dots,X_{-n}) = H(Y|X_{-1},\dots,X_{-n}) - H(Y|X,X_{-1},\dots,X_{-n})$$

Dividing by time yields the value for MI$_{\text{n}}$.

$$MI_n = B_n/T$$

5.2.3. MI$_{\text{ne}}$

Townsend et al. showed that errors in P300 systems are systematic, and therefore information about the types of errors that occur during spelling could be useful for an end user or post-processing method (2010). Below, we propose error models based on values determined in their analysis. First, errors in the checkerboard paradigm have been shown to occur more often within the same virtual matrix as the target character.

$$p(y|x) = \begin{cases} P & x = y \\ \dfrac{(1-P)P_1}{\frac{N}{2}-1} & \mathrm{x} \neq y, cb(x) = cb(y) \\ \dfrac{(1-P)P_2}{\frac{N}{2}} & cb(x) \neq cb(y) \end{cases}$$

Where $cb(x)$ refers to the virtual matrix that character x is assigned to, P_1 refers to the probability of an error occurring in the same virtual matrix as the target, and $P_2 = 1 - P_1$ refers to the probability of an error occurring in a different virtual matrix. These values were found to be .7414 and .2586 respectively by Townsend et al. (2010).

Next, there were three distinct types of errors found in the row/column paradigm. Adjacent characters were observed to be selected the most often, followed by characters that shared a row or column with the target character, both of which were more likely than erroneously selecting a distant character.

$$p(y|x) = \begin{cases} P & x = y \\ \frac{(1-P)P_1}{N_1} & |r(x) - r(y)| + |c(x) - c(y)| = 1 \\ \frac{(1-P)P_2}{N_2} & \mathrm{x} \neq y, \big(r(x) - r(y)\big)\big(c(x) - c(y)\big) = 0 \\ \frac{(1-P)P_3}{N_3} & otherwise \end{cases}$$

Here, $r(x)$ and $c(x)$ are the row and column of character x in the matrix. P_1, P_2, and P_3 are the probabilities of incorrectly picking characters that are adjacent, in the same row or column, or anywhere else relative to the target character. N_1, N_2, and N_3 are the numbers of characters that are adjacent, in the same row or column, or anywhere else relative to the target character. The error probabilities were found to be .4065, .4452, and .1483 respectively by Townsend et al. (2010).

Other flashing paradigms are more random so error patterns are less likely to occur. None of the papers included error analysis, so the uniform model was used. The bits per symbol is then

$$B_{ne} = \sum_{x, x_{-1}, \dots, x_{-n}} p(x, x_{-1}, \dots, x_{-n}) \sum_{y} p(y|x) \log \frac{p(y|x)}{\sum_{x'} p(y|x') p(x'|x_{-1}, \dots, x_{-n})}$$

The appropriate error model is used for $p(y|x)$, and $p(x|x_{-1}, \dots, x_{-n})$ and $p(x_{-1}, \dots, x_{-n})$ are determined as before. The information rate is then found by dividing the bits per symbol by the average time per symbol.

$$MI_{ne} = B_{ne}/T$$

5.3. Analysis

Data from published BCI communication studies are used to show the effects of evaluation (Table 5.1). Studies were included if they provided the accuracy and selection speed that were achieved by each study subject, which are the only two values necessary for calculating each evaluation metric, allowing us to evaluate each subject's performance separately using each metric. The average values were then taken for each study arm and the results were reanalyzed. The results of each of these studies were evaluated using both previously published as well as the proposed metrics.

Table 5.1 Parameter values, optimization metrics, and evaluation metrics used in each of the included data sets.

Study	Method	Subjects	N	Steps	ISI (ms)	Gap (s)	Opt	Eval
Kaper (2004)	all	8	36	1	140	2	ITR	ITR
Serby (2005)	all	6	36	1	150	2	ITR	SR, ACC, ITR
Blankertz (2006)		2	6	2	NA	NA	none	OCM
Sellers (2006)	3x3,175	5	9	1	175	5	none	ITR
	3x3,350	5	9	1	350	5	none	ITR
	6x6,175	5	36	1	175	5	none	ITR
	6x6,350	5	36	1	350	5	none	ITR
Furdea (2009)	auditory	13	25	1	625	3.75	WSR	ITR, WSR
	visual	13	25	1	287.5	8.75	WSR	ITR, WSR
Cecotti (2010)		8	5	≥3	NA	NA	none	ACC, ITR, OCM
Townsend (2010)	all	18	72	1	125	3.5	WSR	ITR, PBR
Jin (2011)	all	10	84	1	175	2	none	ACC, PBR
Ryan (2011)	all	24	72	1	125	6	WSR	ITR, OCM

Schreuder (2011)	S1	14	6	2	175	18.25	none	ACC, ITR, OCM
	S2a, S2b	14	6	2	175	12	none	ACC, ITR, OCM
Speier (2012)	all	6	36	1	125	3.5	ITR	ITR

5.3.1. Data

Eleven studies were chosen as a representative sample of existing BCI communication literature. Seven visual P300 studies were included: one study focused on optimizing system parameters (Sellers et al., 2006), two proposed new flashing paradigms (Townsend et al., 2010; Jin et al., 2011), two used novel classification techniques (Kaper et al., 2004; Serby et al., 2005), and two integrated language information (Ryan et al., 2011; Speier et al., 2012). The remaining four studies proposed systems based on alternative neurological signal paradigms including audio P300 (Furdea et al., 2009; Schreuder et al., 2011), motor imagery (Blankertz et al., 2006), and SSVEP (Cecotti, 2010). Nine of the studies (Townsend et al., 2010; Jin et al., 2011; Serby et al., 2005; Ryan et al., 2011; Speier et al., 2012; Furdea et al., 2009; Schreuder et al., 2011; Kaper et al., 2004) included comparisons between study arms to validate the proposed method. The remaining two (Blankertz et al., 2006; Cecotti, 2010) each demonstrated their system alone as a proof of concept.

The studies reviewed used a variety of system parameters (Table 5.1), all of which significantly influence system performance. Because these values vary widely, performance differences observed in a comparison across studies could be a result of the different parameter combinations, rather than a validation of the techniques used. Additionally, each study used a different subject population and sample sizes were small (between two and 24), making it difficult to find significant differences in results. Individual studies are usually self-controlled and use standardized systems, which alleviates these concerns. We therefore focus on reanalyzing the comparisons within studies instead of comparing results between studies. Comparison across studies becomes more appropriate in situations where a study builds directly upon a previous one, limiting the parameter and protocol variation.

Each aforementioned study was evaluated using the each of the existing and proposed evaluation metrics. Within each study, the results of the different groups were compared using paired t-tests. These results were then compared to the findings in the original paper.

5.3.2. Optimization

The first analysis performed considered a previously published dataset (Speier et al., 2012). In this study, the probability of each of the possible characters was computed after each stimulus and the most probable character was selected once a confidence threshold was reached. In the published results, the value for the threshold was determined by choosing the value that optimized the results using the ITR metric.

Analysis consisted of re-optimizing the results using each of the metrics detailed above. The new optimal threshold probability is reported for each optimization as well as the corresponding performance using the MI_{2e} metric. These values are then compared to the results from optimizing on the MI_{2e} metric and evaluated for significance using paired t-tests.

5.3.3. Optimization Results

The original optimization reported in Speier et al. used ITR and chose an optimal value of 0.86 for the threshold probability (Speier et al., 2012). Many of the existing metrics chose similar optimal values, with only studies optimizing based on sample rate and accuracy choosing significantly different values. Using MI_0 resulted in the same optimal values, and MI_2 resulted in values that were lower, but not significant (p=0.087) (Table 5.2). The threshold values chosen using MI_{2e} were significantly lower than those using all other metrics, with lower values for five of the six subjects (Fig. 5.4).

Table 5.2 Threshold values and average MI_{2e} values of the data set from Speier et al. (2012) when optimizing on different evaluation metrics. OCM was not computable because the system did not require all errors to be corrected. p-values are determined using paired t-tests between the given value and the results when optimizing on MI_{2e}.

Metric	Threshold	p-value	Bit rate	p-value
MI_{2e}	0.49		17.05	
MI_2	0.82	0.005	16.24	0.021
MI_0	0.86	0.006	16.05	0.020
ACC	0.98	0.001	13.14	0.007
SR	0.12	0.003	15.25	0.008
ITR	0.86	0.006	16.05	0.020
WSR	0.93	0.001	15.31	0.005
PBR	0.87	0.005	15.97	0.015
CPM	0.87	0.005	15.97	0.015
OCM	*	*	*	*

When optimizing on the existing metrics, the average confidence threshold values varied between 0.12 and 0.98, and the corresponding information rates varied between 13.14 and 16.05 bits per minute. The optimized values achieved using MI_0 were identical to those using ITR, and MI_2 achieved an insignificant increase in results (p=0.087). When optimizing on MI_{2e}, the average confidence threshold was significantly lower (0.49) and the derived bit rate (17.05) was significantly higher than those using any other metric (Table 5.2).

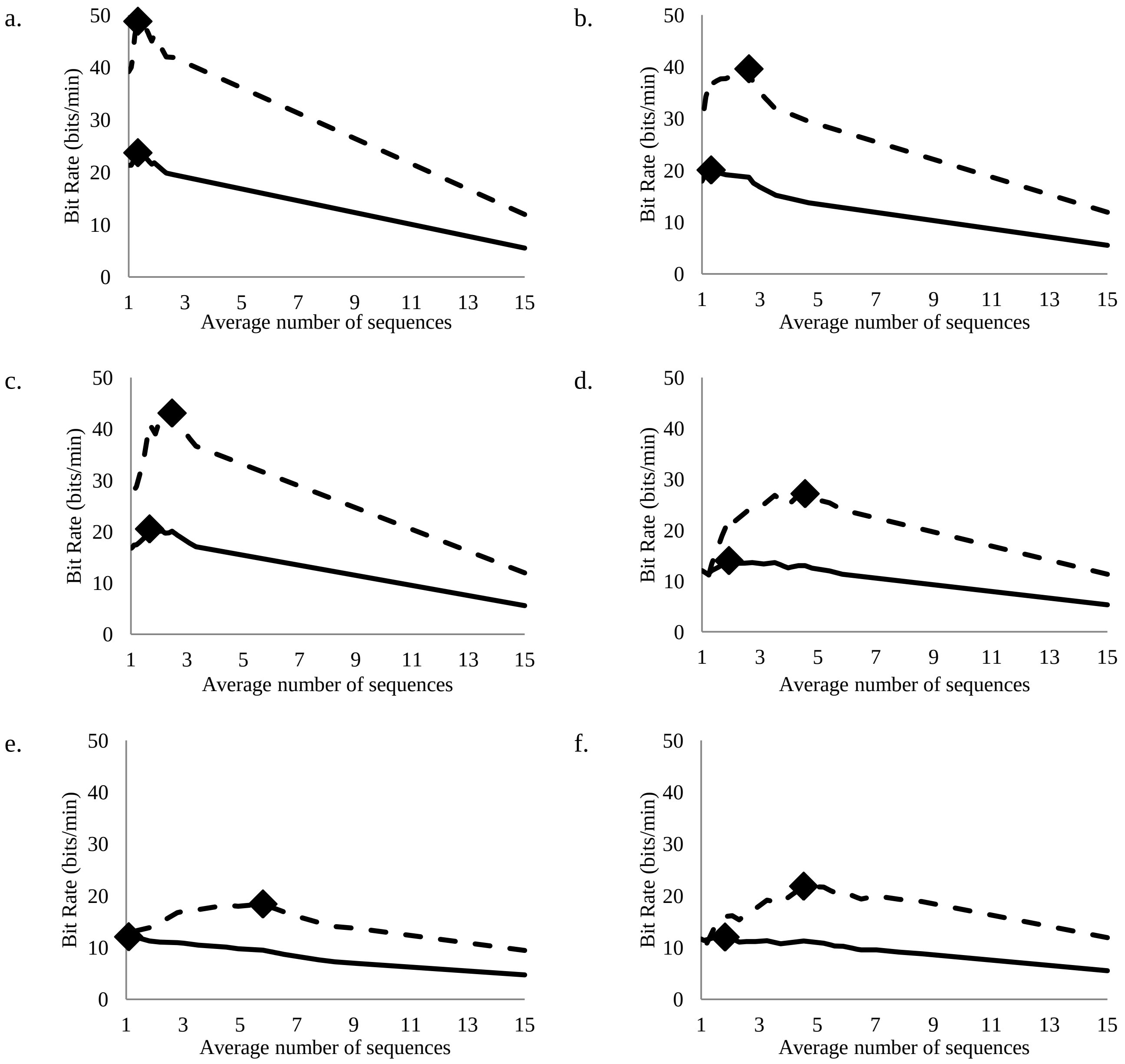


Figure 5.4 Values of ITR (broken curve) and MI_{2e} (full curve) versus the number of stimulus sequences used in classification for each subject in the Speier et al. (2012) data set. Optimal values are marked by diamonds.

5.3.4. Evaluation Results

In the Kaper (2004) study, all metrics other than WSR reflect better results using inner cross validation with significant differences between "inner" and "outer" noted using ITR, MI_0, MI_2, and MI_{2e} (p=0.00044, p=0.00033, p=0.00027, and p=0.00014, respectively), which is consistent with the published conclusions (Table 5.3).

In the Serby (2005) report, all metrics agreed with the original conclusion that independent component analysis achieved a higher bit rate than the maximum likelihood method. Each metric showed significant results other than accuracy (p=0.34) and WSR (p=0.09), with p values ranging from 0.023 (PBR) to 0.008 (MI$_{2e}$).

The Sellers (2006) paper showed varying results depending on the metric used. All existing metrics other than accuracy determined the 3 x 3 grid with an ISI of 175 ms to have the best performance, although none were significant. The three proposed metrics however identified the 6 x 6 grid with an ISI of 175 ms as the superior configuration with significant results (p=0.015, p=0.044, and p=0.002, respectively).

Table 5.3 Results from published P300 papers reevaluated using different metrics. Bold numbers refer to the leading method using that metric and bold method names refer to the results found in the original publication. Asterisks denote methods that cannot be computed for the target system.

Study	Method	ACC	SR	ITR	WSR	PBR	CPM	OCM	MI$_0$	MI$_2$	MI$_{2e}$
Kaper (2004a)	**inner**	**54.38**	**13.85**	**25.21**	0.13	**9.41**	**1.82**	*	**21.13**	**12.94**	**18.06**
	outer	47.88	9.70	14.47	**0.23**	3.10	0.60	*	12.21	7.51	11.59
Serby (2005)	ML	90.02	3.66	15.79	2.45	15.54	3.01	*	12.63	7.49	7.77
	ICA	**92.12**	**4.56**	**19.88**	**3.13**	**19.66**	**3.80**	*	**15.90**	**9.43**	**9.74**
	online	79.53	3.89	13.77	1.72	12.35	2.39	*	11.11	6.64	7.27
Blankertz (2006)		*	*	*	*	*	*	4.88	19.95	11.73	11.73
Sellers (2006)	**3x3,175**	61.25	**3.87**	**4.53**	**0.32**	**3.99**	**1.26**	*	2.43	1.58	1.82
	3x3,350	**69.38**	2.31	3.19	0.10	2.83	0.89	*	1.70	1.11	1.21
	6x6,175	53.75	2.31	4.50	0.26	2.68	0.52	*	**3.72**	**2.26**	**3.10**
	6x6,350	48.13	1.28	1.93	0.00	0.08	0.02	*	1.64	1.01	1.54
Furdea (2009)	auditory	88.08	1.15	4.66	0.91	4.65	1.00	*	3.59	2.22	2.26
	visual	**98.08**	**3.54**	**15.75**	**3.24**	**15.79**	**3.40**	*	**12.15**	**7.46**	**7.49**
Cecotti (2010)		92.25	19.64	35.34	*	*	*	5.51	22.54	13.25	13.25
Townsend (2010)	**cb72**	**91.52**	4.33	**23.01**	**3.12**	**22.45**	**3.64**	*	**15.67**	**9.25**	**9.28**
	rc72	77.34	**4.64**	19.70	2.07	16.51	2.68	*	13.68	8.12	8.59

Jin (2010)	9-P	87.33	**5.82**	**29.32**	3.35	27.14	4.25	*	**18.65**	**11.09**	**11.09**
	12-P	88.00	5.40	27.49	3.20	25.97	4.06	*	17.48	10.39	10.39
	14-P	93.26	3.77	20.93	2.78	20.55	3.21	*	13.34	7.90	7.90
	16-P	93.23	5.26	29.14	**3.85**	**28.36**	**4.44**	*	18.58	11.00	11.00
	19-P	**93.99**	4.70	26.39	3.56	25.86	4.05	*	16.83	9.96	9.96
Ryan (2011)	**PS**	84.92	**3.78**	17.85	2.03	16.46	2.67	**5.28**	**21.64**	**12.71**	**12.71**
	NS	**89.80**	3.74	**19.28**	**2.51**	**18.52**	**3.00**	3.12	12.79	7.51	7.51
Schreuder (2011)	S1	**87.99**	2.08	3.75	*	*	*	0.62	2.54	1.49	1.49
	S2a	86.16	**3.05**	**5.27**	*	*	*	0.91	3.71	2.18	2.18
	S2b	86.07	2.96	5.26	*	*	*	**0.94**	**3.83**	**2.25**	**2.25**
Speier (2012)	Static	82.97	5.91	22.06	2.65	20.24	3.91	*	17.78	10.60	11.42
	Dynamic	89.63	6.45	27.38	4.14	26.61	5.15	*	21.92	13.00	13.55
	NB	**92.59**	**7.31**	**33.15**	**5.51**	**32.91**	**6.37**	*	**26.44**	**15.65**	**16.05**

All metrics in the Furdea (2009) study determined that the visual P300 speller was superior to their audio version. All metrics other than accuracy (p=0.078) revealed significant differences between the two approaches with p values less than 10^{-6}.

In the Townsend (2010) study, the results from the checkerboard paradigm proved better than the row/column paradigm on a 9 x 8 grid by all metrics other than selection rate. The results were significant using ITR (p=0.035), MI_0 (p=0.044), and MI_2 (p=0.047), but not MI_{2e} (p=0.12).

There was variability in the results of the system presented in Jin et al. (2011). The original paper concluded that 19-P method was superior using PBR, which is consistent with the WSR and CPM metrics. However, selection rate, ITR, MI_0, MI_2, and MI_{2e} all indicated the 9-P method was superior. In all cases, the results were close and none were statistically significant.

In the Ryan (2011) paper, evaluation using accuracy, ITR, SWR, PBR, or CPM revealed significantly higher values using the nonpredictive speller with p values between 0.02 and 0.04.

OCM (the metric used in the original paper), MI_0, MI_2, and MI_{2e} all showed significantly higher rates for the predictive speller ($p<10^{-8}$).

In the Schreuder (2011) paper, all metrics other than accuracy showed significantly higher results for the S2a and S2b trials, including ITR (the metric used in the original paper) and the proposed metric ($p<0.0001$). There were no significant differences between the S2a and S2b trials using any metric.

The Speier (2012) paper showed superior results for the naïve Bayes method regardless of the evaluation metric used. All metrics showed significant results other than selection rate ($p=0.064$).

5.4. Discussion

5.4.1. Evaluation

In six of the 11 studies analyzed, changing the evaluation metric could have resulted in different conclusions from that originally published. Only two of the existing metrics, PBR and CPM, always agreed. This highlights the critical importance of identifying an appropriate metric for evaluation of P300 speller studies, and more generally all BCI studies.

The proposed metrics agreed with the published conclusion in nine of the 11 studies. In the Sellers et al. study, all existing metrics chose the 3 x 3 grid because they did not consider actual typing ability. Because they only have nine characters to choose from, their system would not be able to type most English words and is therefore less effective at communicating language. This shortcoming could be addressed by making selections in two steps, but the effectiveness would need to be reevaluated (Sellers et al., 2006). The proposed metrics also would have provided different conclusions in the study by Jin et al., although the results were close and the difference was not statistically significant (Jin et al., 2011).

Most existing metrics could not account for the predictive model used in the Ryan et al. (2011) study. The nonpredictive speller achieved a higher accuracy and similar selection speed, so it was found to be significantly better in most cases. The only existing metric that was able to account for the improvements in their system was the metric introduced in the same paper. The metrics proposed here are able to account for the predictive model and thus agree with the highly significant results found in the study.

Another critical advantage of the proposed metrics is their universal utility. Only the proposed metrics were consistently computable and intuitive across all studies. Some of the existing metrics could not be computed for all of the studies either because all errors were not corrected (OCM), the system involved multi-step decisions (WSR, PBR, and CPM), or rates and accuracies were not recorded for intermediate steps (ITR). While ITR could be computed if all intermediate results were recorded, it did not always reflect actual performance. In the Schreuder et al. (2011) study, subjects were able to type the target sentences faster in the S2b trial, but the S2a trial had a higher ITR value due to the multi-step nature of the system.

5.4.2. Optimization

The performance of BCI systems is influenced highly by system parameter values. These parameters are typically set by optimizing using some metric. Our analyses illustrate the impact of the optimization method on system performance. Optimization is designed to make a value achieve its optimal value, so it is trivial that optimizing on MI_{2e} achieves the highest information rate. The interesting aspect of this analysis is the disparity between the threshold value determined by MI_{2e} and the thresholds chosen using other metrics. The threshold is significantly lower than the values determined by other metrics. A lower threshold results in faster decisions, resulting in significantly higher bit rates when error information is taken into account.

Optimizing on MI_{2e} results in the adoption of lower threshold values in part because it takes the information contained in errors into account. This information may not be useful in all cases. If the reader is not aware of the error model, then this information would be ignored and the functional information transfer would be that described by MI_2. In this case, the optimization on MI_{2e} would be too aggressive, resulting in an error rate that might be too high for practical use. The end application should be considered when choosing the evaluation metric so that the system can be appropriately optimized.

In many BCI communication studies, optimization and evaluation are performed using different metrics. The papers referenced in this study used several different optimization procedures, resulting in incompatible results even after converting them to consistent metrics. Even within papers, various metrics are used for evaluation in order to compare with various different studies. Going forward, we suggest a standard metric should be chosen to standardize BCI results and allow for more consistent comparisons across studies, such as the one presented here.

5.4.3. Error Model

The improvements in results from including the error model varied from negligible amounts (Furdea et al., 2009) to over 50% improvement (Sellers et al., 2006), and were based mainly on the average accuracy achieved. Depending on the application, information considered by this error model might not actually be useful. If the output string is sent to a post-processing algorithm designed to correct errors using this error model, it could be translated into a real increase in overall accuracy. When a human is reading the user input, knowledge of the trends of errors could be useful in trying to determine the attempted output, but this could be a difficult task. Further studies could show a reader's ability to correct different types of errors (see future directions). It is the system designer's role to consider the end application when determining the correct metric to use, and it might be appropriate to omit an error model in some instances.

5.4.4. Limitations

The ideal error model used in MI_{ne} would include the actual probabilities $p(y \mid x)$ for all $\langle x, y \rangle$ pairs for each subject. However, it would be impractical to actually find all of these in a training step, so some simplifying assumptions need to be made. The probabilities of adjacent, same row or column, and same virtual matrix probabilities used in the $p(y \mid x)$ values used in section 5.2.3 would vary between subject and system, and therefore should be calculated during training rather than blindly using the values provided by Townsend et al. (2010). Unfortunately, studies rarely publish these numbers, so this was not possible in this study.

While adopting a standard evaluation metric makes information rates of BCI systems comparable, comparisons between studies can still be misleading. BCI systems are high-dimensional systems that can have many different parameters, electrode configurations, and hardware constraints. It is therefore difficult to determine whether an improved performance corresponds to a superior method or a better tuning of the system parameters. For this reason, researchers should be cautious when comparing between studies and limit these comparisons to situations where studies share similar configurations such as when a new study directly builds upon a previous one. Some work has been performed in parameter optimization (Sellers et al., 2006; McFarland et al., 2011; Lu et al., 2012), but several aspects such as the length of the pause between selections have not been addressed. Furthermore, most of these studies involved healthy subjects, so the translation of these results into the target patient population could vary between systems, irrespective of the evaluation metric used.

5.4.5. Future Directions

In this study, we focused on using BCI systems for communicating language information. In general, these systems are often extended to include various types of menu-based commands (Townsend et al., 2010). Probabilities for selections can still be computed similar to the n-gram language model, assuming a data set of sequences of selections is provided. In this case, the conditional probability of a selection sequence would be the relative frequency of that sequence in the selection history. To our knowledge, no such data set has been published. Furthermore, all studies that we know of were performed using a pure spelling task. Studies of alternate uses of these systems would allow us to create more general models of selection probabilities in order to further generalize this metric.

To date, no BCI communication systems use information about the types of errors to improve their selections. Current systems treat all errors as a wrong answer that is either ignored or deleted, rather than combining it with knowledge about common types of errors to acquire information about the target symbol. Applications can improve their output if they incorporate this information through either a post-processing program or integrate it into the classifier itself. When designing a BCI system, constraints of the target domain should be considered because they provide information that can improve overall performance when incorporated into the classifier. To this end, we have recently reported the benefits of integrating knowledge of language into P300 speller classification (Speier et al., 2012).

Ultimately, the goal of a communication system is to convey the intent of the user. It is clear that a lower error rate is preferable, but it is uncertain how low it needs to be in order for the output to be understood. In addition to the number of errors, the types of errors that occur can be important to reader comprehension. In English, for instance, replacing a vowel with another vowel will often result in another word, while replacing a vowel with a consonant will usually result in a string that is not a word, making the error more apparent and easier to correct. The relationship between language-based BCI output accuracy and reader understanding has not yet been studied.

5.4.6. Conclusion

The performance metric used is integral to the evaluation of BCI systems as it can influence optimization and comparison of different methods. Current methods for evaluating language-based BCI systems are largely misapplied and based on incorrect assumptions, leading to suboptimal implementations and misleading results. System designers should consider the inherent structure of the language domain and the ultimate goal of communication when developing and evaluating these systems. The mutual information metric presented here compensates for many of these shortcomings and provides a better way to compare and evaluate language-based BCI results.

PART II: IMPROVING SYSTEM PRACTICALITY

While significant research has been conducted on EEG signal classification, comparatively little time has been devoted towards system practicality for “locked-in” patients. Despite availability, adoption of assistive technology for communication has been impeded by several factors, including ease of use of the device, including time taken to program the device, reliability, limited vocabularies, and the time taken to generate a message (Baxter et al., 2012). A survey of “locked-in” patients shows that these subjects desire typing speeds of 15-19 characters per minute with 90% accuracy, with setup times less than 30 minutes and only 2-5 training sessions (Huggins et al., 2011). These benchmarks are largely unmet: many systems can meet the 90% accuracy requirement, but only with insufficient typing speeds; current systems require approximately 30 minutes for setup excluding training sessions that can take up to 20 additional minutes; and training sessions are required for every use of the system. Improving the usability of the system therefore remains a significant challenge for widespread adoption of BCI communication systems.

Current BCI systems require expensive hardware that is complicated and time consuming to set up and requires significant expertise for maintenance and debugging. Minimization of the required hardware is paramount for a practical BCI solution to limit the cost and maintenance as well as to reduce the time and difficulty of the system setup. While most research systems use a complete EEG system, several empirically defined configurations exist for healthy subjects (Kaper et sl., 2004; Krusienski et al., 2008; Hoffmann et al., 2008). These electrode montages address most of the hardware concerns for healthy subjects, but they have been shown to be suboptimal in cases where gaze is restricted. Methods for defining subject specific montages have been developed recently (Cecotti et al., 2011; Xu et al., 2013; Colwell et al., 2014), but they require an initial test using a larger system, so the translation into a more practical system for the target population is uncertain.

Rather than focusing on the classifier, P300 speller performance can be improved by increasing the quality of the signal. Approaches to improving the signal have focused either on changing the stimulus presentation or the signal acquisition modality. The traditional row/column stimulus presentation paradigm has been shown to be suboptimal both in speed (Jin et al., 2011) and its ability to produce consistent responses (Townsend et al., 2010). Townsend et al. (2010) presented a "checkerboard" flashing paradigm which produced better response signals despite slowing flashing speeds. Jin et al. (2011) created a flashing scheme that reduced the amount of time required, but did not address the response signal concerns.

While noninvasive approaches are generally preferable, "locked-in" patients have expressed interest in invasive approaches if the added performance is sufficient to justify the risk of invasive surgery. Recent studies have tested BCI systems using Electrocorticography (ECoG), an invasive neural signal modality that implants passive electrodes on the cortical surface (Brunner et al., 2011; Krusienski and Shih, 2011a; Speier et al., 2013a). These studies have shown the viability of invasive electrodes in BCI communication as some of the subjects in these studies have shown results exceeding those of the best EEG subjects (Brunner et al., 2011; Speier et al., 2013a). However, performance seems to vary depending on the location of the implanted electrodes and, because these studies use data from subjects with electrodes implanted for other neurological treatments, the electrode location could not be specified and the subject sample sizes were small.

Another critical factor with respect to "ease of use" is system training, which can be time consuming and distract from time during which the user could be using the system. Current P300 speller iterations require training sessions before each use during which the user copies a specified string of characters. Prior work has attempted to create a general classifier that would remove the necessity for this training session (Kaper et al., 2004). In this study, the results using the general classifier dropped significantly from those trained on the specific session. Kindermans et al. (2012) proposed a method using expectation maximization to train the system during an unsupervised session of free spelling. During use, the subject selects characters for a target word or phrase and, after each selection, the classifier attempts retraining. Initial selections will be wrong, but the system eventually learns the correct classification. Speier et al. (2013c) applied the HMM method from chapter 3 to create a similar system that retrospectively corrects the initial misclassified characters. In this system, characters are initially classified incorrectly as the system performs unsupervised learning. As it is trained, it overwrites the previous classifications with more accurate ones based on the trained system.

This section contains four chapters. In chapter 6, a Gibbs sampling method is presented for finding the optimal electrode montage across a population of users as well as a four electrode configuration which performs optimally for healthy subjects. Chapter 7 describes a study of two subjects using implanted ECoG electrodes to type using the P300 speller. Chapter 8 describes a novel stimulus presentation paradigm that increases system speed while optimizing the characteristics of the response signal. In chapter 9, an unsupervised approach to training the P300 speller using the Baum-Welch algorithm is described.

6. ELECTRODE PLACEMENT

While P300 speller system performance is an active area of research, there is comparatively little focus and investigation with respect to “ease of use.” Choosing the optimal placement and number of electrodes is essential in any EEG system as it balances the amount of available data against the set-up time, cost, and system complexity. There is minimal objective and quantitative analyses of the minimal EEG electrode set that can be used to achieve optimal system performance. Several studies have found similar offline performance in the P300 speller between using a 32 channel EEG and empirically chosen sets of six (Krusienski et al., 2008), eight (Hoffmann et al., 2008), and ten (Kaper et al., 2004) electrodes. These studies did not quantitatively deduce the described electrode sets nor consider other reduced configurations, so they present potential configurations, but do not show that they are minimal or optimal. They are also restricted to P300 experiments using healthy subjects and have been shown not to translate well to situations where user gaze is limited, which is often the case in the target population (Brunner et al., 2011).

Several recent studies have developed methods to rank electrodes based on their contribution to offline classification accuracy on an individual subject basis in P300 systems using EEG (Cecotti et al., 2011; Xu et al., 2013; Colwell et al., 2014) and electrocorticography (ECoG) (Speier et al., 2013a). These studies employed methods in which an initial testing phase included data from a complete electrode set to later determine a subject’s optimal configuration, only after which the number of channels could be reduced on a subject-specific basis. The configurations described in these studies varied in channel number and location between subjects, leading to the conclusion that subject-dependent configurations are necessary for optimal P300 performance. However, none of these studies attempt to optimize across subjects to provide a general configuration for comparison. Moreover, none of these studies validated their results with prospective trials using the reduced number of electrodes to demonstrate that they were robust across sessions. Finally, the methods employed in these studies do not improve “ease of use” for end users because the described approaches require a full set of recording electrodes and amplifiers for each subject before identifying an optimized reduced set. While some of this hardware would only need to be available during an initial configuration phase, changes in the user’s state such as loss of eye gaze control would result in a changing optimal configuration that would require repeated access to this entire system.

The goal of this project was to provide a method for optimizing EEG electrode placement for P300 studies across a subject population, which we demonstrate by providing a minimal set of electrodes for studies conducted on healthy subjects. In this study, we initially use a retrospective offline analysis approach using a previously published data set of 15 healthy subjects with 32 electrodes (Speier et al., 2014b). Gibbs Sampling was used to find sets of electrodes based on the joint distribution of the subjects’ EEG signals and the known labels. Offline testing with a naïve Bayes classifier (Speier et al., 2012) was performed using data from each of these electrode sets to show the relationship between the number of electrodes and system performance. The optimal four electrode set was then evaluated prospectively online against the full 32 electrode set as well as the six electrode set presented by Krusienski et al. (2008) to validate its viability in a real time BCI system.

Ultimately, these studies demonstrate an important method to generate a clinically and practically non-inferior reduced electrode set for a P300 speller that can and should be applied to target populations to determine if an optimal reduced electrode set can be identified in affected patient populations.

6.1. Generating Channel Sets

The retrospective study was performed using the 15 subject offline dataset described in chapter 3. This data was used to find electrode configuration that provided the best performance while also limiting the number of channels required. Preliminary offline analysis was then performed on this dataset to verify that the reduced channel set provided comparable results to the full electrode set as well as the reduced set proposed by Krusienski et al. (2008). An online study was then conducted which allowed a new set of 15 healthy subjects to spell a sentence of their choosing in order to simulate actual use of the system.

6.1.1. Feature selection

For each stimulus, the 32 channel EEG data for the next 600 ms was decimated by a factor of 12 and concatenated into a vector. The vector that corresponds to stimulus i for letter t in the sequence was considered the feature vector, z_t^i for use in classifying that stimulus. Stepwise linear discriminant analysis (SWLDA) used a stepwise method to separate the available features into two groups based on whether the feature was significant in classification. At each step, the most significant feature above a threshold in the non-significant group was added to the significant group. Similarly, the least significant feature below a threshold in the significant group was removed from use in classification. The probabilities of adding and removing features were 0.1 and 0.15 respectively. These steps were repeated until the number of significant features reached a threshold of 60 features or until the feature groups reached equilibrium. These significant features were then stored in a weight vector, w. (Krusienski et al. 2006)

During testing, the dot product between the feature vector for each stimulus and the feature weight vector was taken to determine a score for that stimulus:

$$y_t^i = w \cdot z_t^i$$

The means and standard deviations were then found for the scores for target and non-target stimuli. Assuming a normal distribution, the probability density function (PDF) for the likelihood probability were computed,

$$f\left(y_t^i | x_t\right) = \begin{cases} \frac{1}{\sqrt{2\pi\sigma_a^2}} e^{-\frac{1}{2\sigma_a^2}\left(y_t^i - \mu_a\right)^2} & if\ x_t \in \boldsymbol{A}_t^i \\ \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{1}{2\sigma_n^2}\left(y_t^i - \mu_n\right)^2} & if\ x_t \notin \boldsymbol{A}_t^i \end{cases}$$

where μ_a, σ_a^2, μ_n, and σ_n^2 are the means and variances of the distributions for the attended and non-attended scores, respectively, and $\boldsymbol{A}_t^i$ is the set of characters included in stimulus i for letter t.

6.1.2. Classification

Classification was performed using a naïve Bayes classifier which determined the conditional probability of a target character, x_t, given a set of flash scores and the history of previous decisions (Speier et al., 2012).

$$p(x_t | \boldsymbol{y}_t, x_{t-1}, \dots, x_0) \propto p(x_t | x_{t-1}, \dots, x_0) \prod_i f\left(y_t^i | x_t\right)$$

where $p(x_t | x_{t-1}, \dots, x_0)$ is the prior probability of a character given the previous selections and $f\left(y_t^i | x_t\right)$ are the PDFs for the likelihood probabilities.

This probability was simplified using the second-order Markov assumption to create a trigram model where the target character only depends on the previous two selections (Manning and Schütze, 1999). The prior probability a character, x_t, given the previous two character selections, x_{t-1} and x_{t-2}, can then be estimated by dividing the number of times that all three characters occur together in a training corpus by the number of times the last two characters occur together. In this study, prior probabilities for characters were obtained from frequency statistics in an English language corpus (Francis and Kucera, 1979):

$$p(x_t|x_{t-1},\dots,x_0) \approx p(x_t|x_{t-1},x_{t-2}) = \frac{c(x_{t-2},x_{t-1},x_t)}{\sum_{x_t} c(x_{t-2},x_{t-1},x_t)} = \frac{c(x_{t-2},x_{t-1},x_t)}{c(x_{t-2},x_{t-1})}$$

where $c(x_{t-2},x_{t-1},x_t)$ represents the number of times the string "$x_{t-2}x_{t-1}x_t$" occurs in the corpus.

After each stimulus, the probability $p(x_t|\boldsymbol{y}_t,x_{t-1},\dots,x_0)$ is computed for each character. If the probability for any character exceeds a set threshold value, that character is chosen and the system moves on to the next character in the sequence. In offline analysis, the threshold probability values between 0 and 1 were tested in increments of 0.01 and the value that maximized the bit rate was chosen for each subject. Although the number of flashes was fixed for all trials in the offline study, different selection rates were simulated by limiting the amount of data available for the classification algorithm. For example, if the confidence threshold was reached after 100 flashes, the corresponding data was used for classification and the rest was omitted. In online trials, the threshold was set to 0.95 for all subjects based on previous experiments (Speier et al., 2014a).

6.1.3. Gibbs Sampling

Channel sets were found using Gibbs sampling to optimize the joint probability of the EEG data and the known labels of the offline data set. Gibbs sampling is a Markov chain Monte Carlo (MCMC) method for estimating high dimensional distributions by generating a "random walk" through the state space (Liu, 2001). In MCMC methods, an initial configuration of the variables is set randomly along with a set of transition probabilities to new configurations based on their relative likelihood. The system then moves randomly through the state space and the frequency of a given state is proportional to its probability.

Here, the state of the system is the set of channels included in analysis. Channel inclusion was represented by a vector of binary variables, c, where $c_j = 1$ if channel j was used in classification. The feature weight vector when trained using the data in the channels indicated by c is represented by w_c. Scores are obtained as before by taking the dot product with the feature vector:

$$y_t^i(c) = w_c \cdot z_t^i$$

In Gibbs sampling, one variable is chosen, c_j, and the remaining, c_{-j}, are held constant. The chosen variable is then assigned a new value according to its probability distribution conditioned on the other variables, c_{-j}, and the known values for the signal, $z^{(s)}$, and the targets, $x^{(s)}$, for all subjects s:

$$p\left(c_j \middle| c_{-j}, x, z\right) \propto p\left(x \middle| y(c)\right) p\left(c_j \middle| c_{-j}\right) = p\left(c_j \middle| c_{-j}\right) \prod_s p\left(x^{(s)} \middle| y(c)^{(s)}\right)$$

When each stimulus response is treated as independent, the posterior probability of the known targets can be computed from the likelihood PDFs:

$$p(x|y(c)) = \prod_t \prod_i \frac{f\left(y_t^i(c)|x_t\right)p\left(x_t \in \boldsymbol{A}_t^i\right)}{\sum_{x_t'} f\left(y_t^i(c)|x_t'\right)p\left(x_t' \in \boldsymbol{A}_t^i\right)}$$

Once c_j is assigned, another variable is chosen and the process is repeated. The prior probability $p(c_j|c_{-j})$ defines a spatial bias based on expected features of an optimal configuration. Here, it is determined by an Ising model which gives reduced weight to configurations containing adjacent channels, which are likely to contain redundant information.

$$p(c_j|c_{-j}) \propto e^{-\beta \sum_{k \in n(j)} c_j c_k}$$

where $n(j)$ represents the indices of electrodes that are adjacent to electrode c_j in the initial 32 channel configuration.

For each electrode set size, the channel set that occurred most often in the Gibbs sampling was chosen as optimal. The offline data set was censored based on each of these configurations and then evaluated using a naïve Bayes classifier. Channel sets were evaluated based on the incremental improvement over the smaller sets as well as its performance relative to the results when using the Krusienski and full 32 electrode configurations.

6.2. Validation

6.2.1. Offline Dataset

The dataset from Chapter 3 was used in offline analysis, consisting of 15 healthy graduate students and faculty with normal or corrected to normal vision between the ages of 20 and 35. Only one subject (subject F) had previous experience using a BCI for typing. Data was acquired using g.tec amplifiers and electrode cap (Guger Technologies, Graz, Austria) with 32 channels in an established configuration (Lu et al., 2012, Sharbrough et al., 1991). The signals were sampled at 256 Hz, grounded to the left ear, referenced to AF_Z, and filtered using a band-pass filter of .1 to 60 Hz. The system used a 6 × 6 character grid, row and column flashes, and an interstimulus interval (ISI) of 125 ms. Subjects underwent between 8 and 10 trials, each consisting of spelling a five letter word with 15 sets of 12 flashes (six rows and six columns) for each letter. The choice of target words for this experiment was independent of the trigram language model used. Gaze was not fixed or tracked.

6.2.2. Protocol

The subjects for the online study consisted of 15 healthy volunteers with normal or corrected to normal vision between the ages of 20 and 30. The training sessions for these subjects consisted of three sessions of copy spelling 10 character phrases. Each subject then chose a target phrase to spell in online sessions. In each online session, the subject had either one five-minute session (subjects P-U) or two two-minute sessions (subjects V-AD) to spell as much of the phrase as they could. Subjects were instructed not to correct errors and to repeat the phrase if they completed it in under five minutes. One session was performed for each of three electrode configurations: the full 32 electrode configuration, the six electrode subset proposed by Krusienski et al., and the optimal four electrode set found during offline analysis. The order of the three online sessions was chosen for each subject using a random number generator. One volunteer could not participate in the study because connection in the occipital electrodes could not be obtained due to hair thickness.

BCI2000 was used for data acquisition and online analysis (Schalk et al., 2004). Offline analysis was performed using MATLAB (version 7.10.0, MathWorks, Inc, Natick, MA).

6.2.3. Evaluation

Evaluation of a BCI system must take into account two factors: the ability of the system to achieve the desired result and the amount of time required to reach that result. The efficacy of the system can be measured as the selection accuracy, which was defined as the proportion of correct characters in the final output string. The speed of the system was measured using selection rate (SR), the average number of selections per minute. In offline analysis, SR was found by taking the inverse of the average time required to make a selection. In online trials, the selection rate was computed by dividing the number of selections by the time required for those selections. For this analysis, the time was defined as the period between the start of the trial and the timestamp of the final character selected.

As there is a tradeoff between speed and accuracy, we also use information transfer rate (ITR) (in bits per minute) for evaluation, which takes both into account. The bits per symbol, B, is a measure of how much information is transmitted per selection on average (Pierce, 1980):

$$B \ = \ \log_2 N + P \log_2 P + (1-P) \log_2 \frac{1-P}{N-1}$$

where N is the number of possible characters (36) and P is the selection accuracy. ITR can then be found by multiplying the selection rate by the bits per symbol. Friedman tests were used to evaluate significant differences between electrode configurations and Wilcoxon signed-rank tests were used for pairwise comparisons and post-hoc analysis.

6.2.4. Channel Sets

In general, the average bit rate increased with the number of channels used for classification. Sets of one, two, or three electrodes produced average offline ITR values of 12.1, 20.1, and 26.4 bits per minute respectively (Table 6.1). These values were all significantly lower than the average performance using a set of four electrodes (PO_8, PO_z, PO_7, CP_z; Table 6.1). The average bit rate plateaued with incremental increases after four channels (Fig. 6.1). Because increasing the number of electrodes did not provide a significant improvement in bit rate (p=0.072) beyond four electrodes, the optimal four electrode set was selected for comparison against the six electrode set proposed by Krusienski et al. (2008) and the full 32 electrode configuration (Fig. 6.2).

Table 6.1 Channels and average bit rates for the first six electrode sets, the Krusienski set, and the full 32 electrode configuration.

Channel Set	Included Channels	Average Bit Rate	Icremental p Value
1	PO8	12.10	<0.001
2	PO8, POz	20.13	<0.001
3	PO8, POz, PO7	26.42	<0.001
4	PO8, POz, PO7, CPz	28.93	0.003
5	PO8, POz, PO7, CP1, CP2	29.34	0.079
6	PO8, POz, PO7, CP1, CP2, FCz	29.83	0.56
Krusienski	PO8, PO7, Oz, Pz, Cz, Fz	29.46	N/A
32	All	31.80	N/A

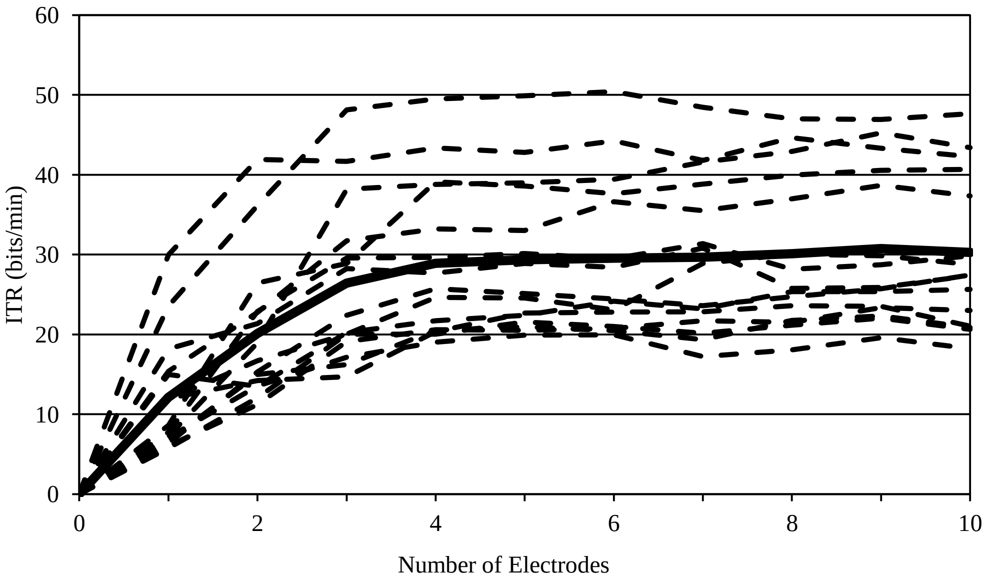


Figure 6.1 Individual (dashed) and overall average (solid) bit rates for the 15 subjects in offline analysis versus the size of the electrode set used.

The four electrodes in the reduced set all showed strong excitatory response potentials (ERP) in response to target stimuli (Fig. 6.3). The PO_8 and PO_7 channels showed a pronounced negative inflection after a delay of about 200 ms and a smaller positive inflection after 300 ms. The PO_Z and CP_Z waveforms did not include the first negative inflection, but showed a higher positive response after the 300 ms delay. All of the channels showed an oscillatory component at the stimulus frequency (8 Hz), but it was generally larger for the occipital electrodes.

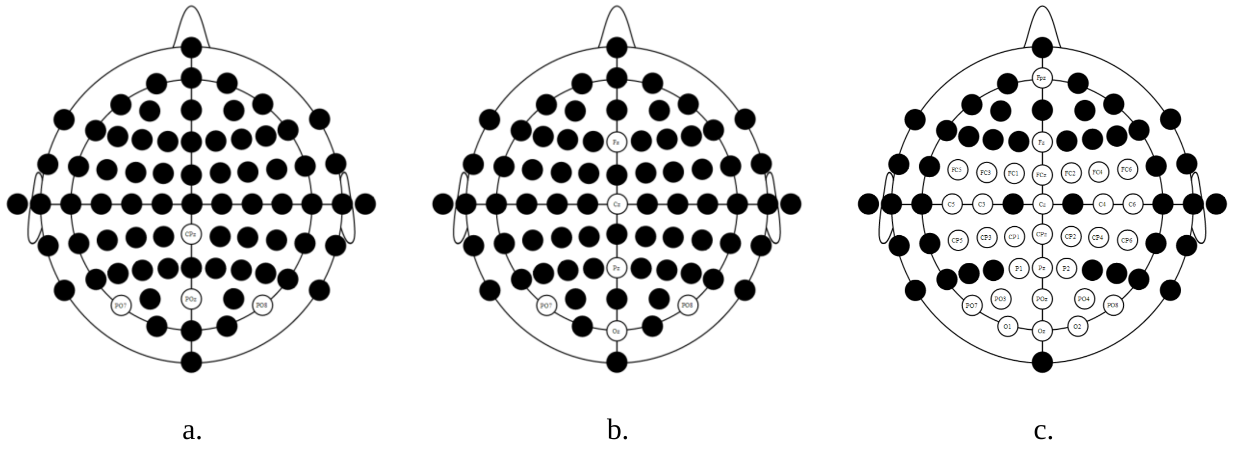


Figure 6.2 Electrode configurations used for online and offline analysis: the optimal four electrode set found in this study (a), the six electrode set proposed by Krusienski et al. (2008) (b), and the full 32 electrode configuration (c).

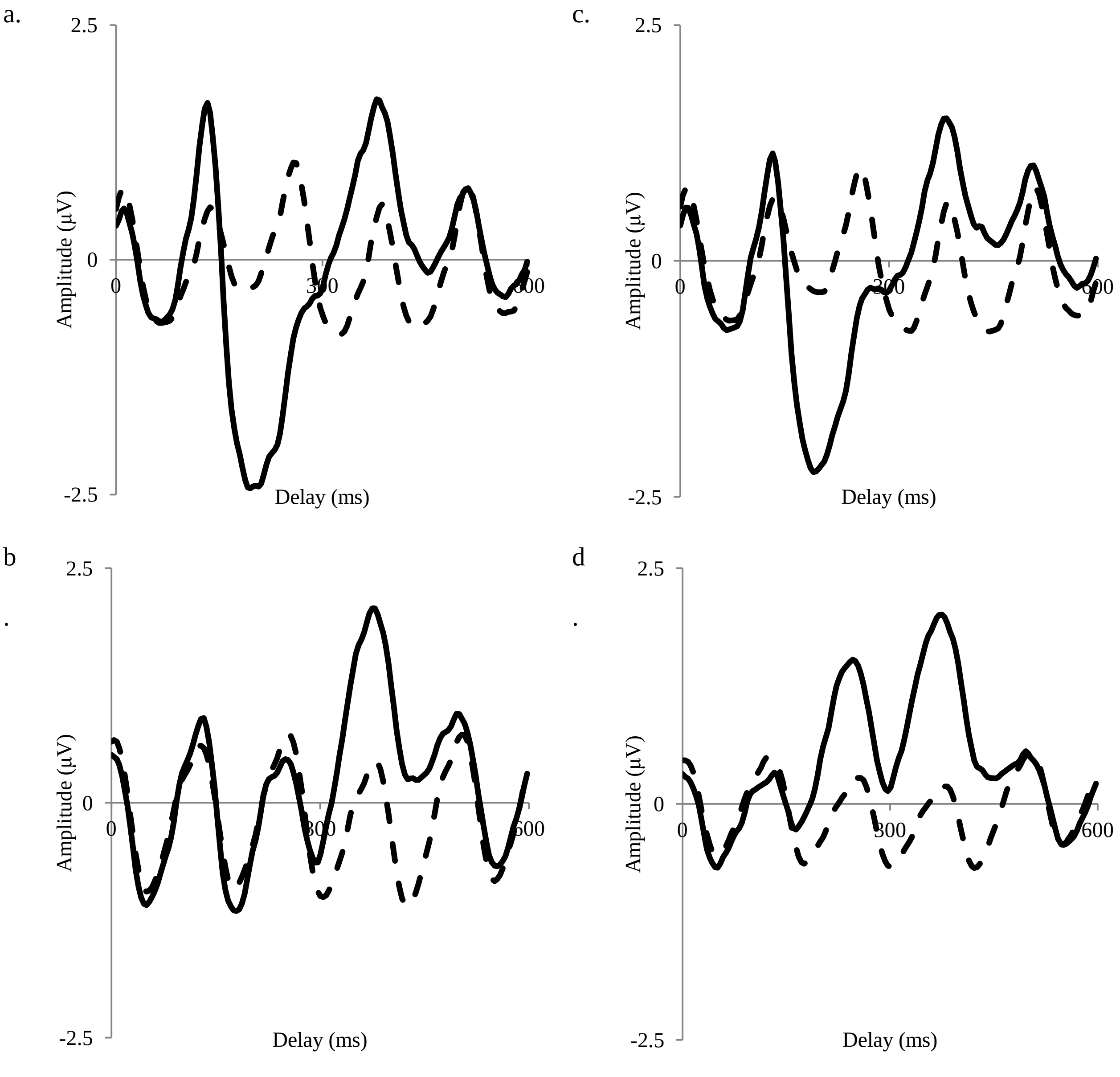


Figure 6.3 Grand average EEG responses for attended (solid) and nonatended (dashed) stimuli for channels PO_8 (a), PO_7 (b), PO_Z (c), and CP_Z (d).

6.2.5. Offline Analysis

The average bit rates for the four channel, six channel (Krusienski et al., 2008), and 32 channel electrode configurations in offline analysis were 28.93, 29.46, and 31.80 bits per minute respectively (Table 6.2). While the Krusienski configuration had a higher average bit rate, results for the four and six channel configurations did not differ significantly in our within subject analysis (p=0.19); seven of the 15 subjects performed better using the four electrode configuration. Both the four and six electrode configurations' bit rates were statistically significantly lower than the full 32 electrode configuration (p=0.004 and p=0.021 respectively). The differences were relatively small as the difference between the 32 electrode results and the four and six electrode sets were 9.0% and 7.4% respectively.

Table 6.2 Optimal selection rates, accuracies, and bit rates for the 15 subjects after optimizing on ITR in offline analysis.

	SR (sel/min)			ACC (%)			ITR (bits/min)		
Subject	4	6	32	4	6	32	4	6	32
A	8.44	7.74	9.26	93.33	100.00	95.56	40.29	39.99	43.33
B	5.90	6.46	7.42	91.11	80.00	82.22	26.39	22.12	26.58
C	4.15	6.43	6.09	72.50	72.50	82.50	19.43	18.70	21.96
D	8.96	10.84	10.94	100.00	95.56	93.33	49.34	50.74	48.94
E	4.56	8.73	6.22	93.33	68.89	86.67	21.55	23.41	24.38
F	7.62	8.16	9.96	91.11	95.56	88.89	34.49	38.20	40.80
G	5.52	6.12	8.19	92.00	98.00	84.00	25.98	30.14	30.42
H	8.71	9.99	10.19	96.00	90.00	92.00	42.86	41.83	44.39
I	4.93	6.17	6.72	94.00	94.00	88.00	27.77	27.97	27.06
J	4.48	6.68	5.74	86.00	82.00	96.00	19.33	23.82	27.09
K	4.78	8.07	6.50	80.00	66.00	88.00	20.77	20.19	26.16
L	6.52	7.32	8.07	90.00	88.00	84.00	29.18	29.48	29.98
M	5.10	7.54	8.59	78.00	72.00	70.00	19.18	21.71	23.63
N	7.56	8.57	9.17	98.00	94.00	90.00	37.24	38.87	38.39
O	4.21	7.30	7.12	90.00	74.00	82.00	20.08	21.96	25.38
average	6.10	7.74	8.01	89.69	84.70	86.88	28.92	29.94	31.90

6.2.6. Online Performance

In online testing, subjects achieved average bit rates of 20.83, 20.91, and 21.67 using the four, six, and 32 electrode configurations respectively (Table 6.3). No significant difference was found between subjects' results when using the three electrode configurations (p=0.92). Eight of the 15 subjects performed better using the four electrode configuration than the full set and eight performed better using the four electrode set than with the Krusienski set. Six subjects experienced a bit rate below 10 for at least one of the configurations, which did not occur for any subject in offline analysis.

Table 6.3 Online selection rates, accuracies, and bit rates for the 15 subjects using the four, six, and 32 electrode configurations.

	SR (sel/min)			ACC (%)			ITR (Bits/min)		
Subject	4	6	32	4	6	32	4	6	32
P	6.84	7.44	9.90	41.18	56.76	68.75	8.03	14.62	26.45
Q	8.80	8.59	8.13	87.36	82.93	75.00	34.98	31.23	25.01
R	7.32	5.08	10.33	100.00	88.00	91.11	37.82	20.44	44.23
S	6.88	8.76	8.32	70.59	62.79	43.90	19.17	20.23	10.84
T	3.35	2.66	5.95	75.00	15.38	72.41	10.31	0.56	17.30
U	3.21	4.51	6.67	62.50	68.18	18.18	7.35	11.90	1.93
V	3.46	3.52	3.87	46.15	30.77	40.00	4.88	2.56	4.35
W	4.83	5.21	6.69	41.18	57.89	68.18	5.67	10.57	17.62
X	5.49	6.76	6.04	100.00	88.00	75.00	28.40	27.19	18.59
Y	4.46	3.99	4.80	58.82	60.00	61.11	9.28	8.57	10.61
Z	5.46	8.20	6.65	80.95	87.50	64.00	19.05	32.70	15.83
AA	10.21	9.98	8.66	89.74	81.58	94.12	42.55	35.30	39.37
AB	8.98	9.94	10.26	75.00	86.96	85.42	27.64	39.19	39.21
AC	9.53	11.03	10.33	88.89	76.74	75.61	39.05	35.22	32.20
AD	5.27	4.99	6.17	80.77	95.65	80.77	18.33	23.39	21.44
Average	6.27	6.71	7.52	73.21	69.28	67.57	20.83	20.91	21.67

6.3. Discussion

Using Gibbs sampling, the number of electrodes required for an EEG-based BCI system can be reduced without significantly compromising information transfer. The application of this novel methodology to identify and validate a reduced electrode set within a subject population provides a potentially important resource for optimizing system design and performance in target populations. While the methodology is evaluated using a specific application (P300 speller) in a specific population (able-bodied subjects), the current methodology is agnostic to application or population.

6.3.1. Methodological Validation

In the population studied, users achieved comparable bit rates using the four electrode configuration as with the six electrode Krusienski set as well as the full set of 32 electrodes in both on and offline analysis. In the retrospective offline exploratory study, the difference between the four electrode set and the full set was statistically significant, but the results using the four electrode set were sufficient to make the system clinically viable. In the prospective online validation analyses, no significant difference was found between the results using these configurations, further indicating the potential power of this new methodology to identify a reduced electrode set without loss of information.

The optimal four electrode configuration found in our test population was consistent with previously published channel sets for healthy subjects. Methods that chose subject-specific sets of electrodes showed a high variability, but generally favored occipital and parietal electrodes, which was consistent with the coverage of our reduced set. The three most common electrodes chosen by Colwell et al. (2014), PO_8, PO_Z, and PO_7, completely overlapped with our four electrode set. Cecotti et al. (2011) and Xu et al. (2013) commonly included P_8, P_Z, and P_7, which were the closest available locations to those that we found most effective. The empirically derived configurations proposed by Kaper et al. (2004), Krusienski et al. (2008), and Hoffmann et al. (2008) all included either PO_7 and PO_8 or P_7 and P_8. All three empirical sets also included four midline electrodes: F_Z, C_Z, P_Z, and O_Z, which, while disjoint from the reduced set proposed here, are in close proximity to two of the electrodes (PO_Z and CP_Z) are are likely to capture similar information.

In general, the true optimal electrode configuration may vary between subjects. However, it is also likely that it varies between sessions based on the state of the user and his or her environment. Variability can also arise from the setup as electrodes need to be reconnected for each use and the exact location and strength of the connection will not be completely constant. As a result, any study that attempts to find a subject-specific configuration must use multiple session in order to verify that they are truly finding variation between subjects rather than sessions. Also, prospective tests using a fixed configuration are necessary to show that results using the configuration are reproducible.

In general, subjects realized a decrease in accuracy when using the system in online trials using each montage, resulting in average accuracies (73.21%, 69.28%, and 67.57%) that were lower than those published by several previous studies. The low accuracy in this study is largely a result of the speed/accuracy tradeoff inherent in the P300 speller as increasing the number of stimuli presented to the user will generally increase accuracy and reduce system speed. For instance, a study by Guger et al. (2009) showed an average accuracy of 91%, but provided stimuli for a single character for 28.8 seconds, resulting in an average typing speed of 2.1 characters per minute and an average bit rate of 8.9 bits per minute. Using the naïve Bayes algorithm, subjects in this study were typing at speeds around 6-7 characters per minute, resulting in an average bit rate over twice that of the Guger study. While the accuracies that subjects achieved may not be practical for patients, using a higher confidence threshold can increase the accuracy at the expense of speed. It was not practical to optimize this threshold for all subjects and all channel configurations in this study, so a constant value of 0.95 was used across all trials.

Another factor contributing to decreased performance during online trials is likely that subjects were not allowed to correct errors. When an incorrect selection is made using the naïve Bayes method, the wrong characters are used for computing the prior probability for subsequent selections, resulting in additional errors. As a result, the selection rate is relatively unaffected, but the accuracy decreases. A similar decrease in accuracy was previously shown in online implementations of the naïve Bayes algorithm without error correction, which could be addressed either by allowing the user to make corrections, or by implementing an algorithm that can automatically correct errors (Speier et al., 2014a). Systems with automatic correction capabilities are currently being developed (Ryan et al., 2011; Speier et al., 2014a; Speier et al., 2015).

While the current analysis is done in healthy subjects, the validation of this method provides an important opportunity to reduce the number of channels and therefore the usability of the system in a target patient population. Fewer electrodes translates into faster setup time, which addresses one concern expressed in a survey of “locked-in” patients (Huggins et al., 2011). Reducing the number of channels also makes the system more cost effective by requiring fewer amplifier channels, which provides not only hardware cost savings, but also decreased configuration and maintenance demands. Because patients with fixed gaze have trouble with the traditional P300 speller system (McCane et al., 2014), reducing the number of electrodes can also improve accuracy by allowing for more complex analysis methods. Unsupervised training and adaptive classifiers, for instance, could also allow for automatic adaptation to disease progression such as the loss of eye gaze control (Kindermans et al., 2012; Speier et al., 2013b).

6.3.2. Limitations and future directions

While the optimization method is general, the results are dependent upon the software, equipment, classifier, and system configuration used to collect the dataset. The electrode sets found using this method are therefore not necessarily robust across sites, implementations, or populations. For example, this study has used the row-column flashing paradigm, while the checkerboard paradigm has recently become widely used (Townsend et al., 2010). While both systems rely on the same neurological paradigm, it is possible that the distribution of features is not identical and therefore would have a different optimal electrode configuration. Also, when selecting the best electrodes locations, only locations that lie within the initial configuration can be candidates. Thus, any location that was not in the initial set of 32 channels will not appear in the final set, regardless of its value in classification.

Existing electrode montages have been shown to translate poorly into situations where a user's eye gaze is limited (Brunner et al., 2011). The application presented here on a population of healthy subjects is likely to have some of the same issues, as all subjects had gaze control. The Gibbs sampling method, however, is agnostic to the patient state and could be applied to find an electrode montage that works optimally in patient populations with fixed gaze. In long term implementations, it would likely be superior to existing patient-specific optimization methods as its goal is to optimize across patient state and would therefore me more robust to disease progression such as the loss of eye gaze control. This method can also be applied to other BCI systems as well as systems using other signal acquisition paradigms such as subdural electrodes for invasive P300 systems (Speier et al., 2013a).

6.3.3. Conclusion

This work presents a methodology for finding optimal electrode contages across a user population. Using this method in a population of healthy subjects, a four electrode configuration (PO_8, PO_Z, PO_7, CP_Z) is proposed which is shown to produce comparable results to a traditional 32 electrode configuration in online testing for healthy subjects. Reducing the number of channels will reduce the system's set-up time, hardware requirements for end users, and computation requirements for classification. These improvements can help to make the P300 speller system a more viable solution for "locked-in" patients.

7. ACQUISITION MODALITY

Recent studies have suggested that electrocorticography (ECoG) signals could be used in BCI communication due to its increased signal-to-noise ratio, high spatial resolution, and superior spectral content (Leuthardt et al., 2004, Wilson et al., 2006, Brunner et al., 2009, Miller et al., 2010). The obvious drawback of using implanted electrodes is that it requires invasive surgery. Still, a recent survey suggests that this surgery might be acceptable to some ALS patients if it yields sufficiently better performance (Huggins 2011). Sixty-one people suffering from ALS were interviewed about BCI and 72% said that they would consider outpatient surgery and 41% would accept a short hospital stay in order to have an implanted BCI, but that current BCI systems do not yet achieve desired performance levels.

Two groups have already shown promising results in using ECoG in P300 speller studies. Brunner et al. conducted a study consisting of a single patient with a temporal grid and a six electrode strip in the occipital region (Brunner et al., 2011). Their analysis focused on the occipital strip and they reported accuracies approaching 100% after only three sets of flashes. Krusienski and Shih reported results from six subjects with varying electrode placement, mainly in the temporal and parietal cortices (Krusienski and Shih, 2011a). The results of their subjects varied, but they generally showed a marginal improvement over EEG-based performance. Perhaps more interestingly, Krusienski and Shih also reported a significant correlation between the stimulus and spectral components of the response signal in ECoG (Krusienski and Shih, 2011b). They did not, however, go as far as using spectral features in classification to determine if they can supplement temporal features and improve accuracy. Given that the detection and classification of P300 speller signals have traditionally focused on cortical signals in the time domain, further work characterizing spectral domain features is critical in understanding their potential value in target classification.

The goal of this study is to investigate factors that may modulate system performance in an ECoG-based BCI system, including spatial, temporal, and spectral factors. Two subjects with subdural implanted electrodes were tested using the P300 speller. Neither of these patients had electrodes implanted in the parietal cortex, so the actual P300 signal could not be detected. Instead, more general event-related potentials (ERP) including visually evoked potentials (VEP) were observed. Offline analysis was performed to determine the accuracy and bit rate that these subjects can achieve using this system to spell common five letter words both with and without integration of spectral features and natural language processing (NLP) algorithms that could potentially further augment observed performance (Speier et al., 2012). Their results were compared to a dataset of 6 EEG subjects used in a previous publication (Speier et al., 2012). Spatial analysis was also performed to determine the cortical locations with the best classification ability for this task, showing possible optimal electrode placement for an invasive "P300" speller system.

7.1. Electrocorticography

7.1.1. Subjects

The subjects in this study had temporary subdural electrode arrays implanted to localize seizure foci prior to surgical resection. Both had corrected to normal vision and consented through protocol approved by the UCLA Institutional Review Board.

The first subject was an 18 year old left-handed male with seizures localized to the left occipital region. The patient had a 20-electrode grid implanted in the left lateral occipital region as well as subdural electrode strips implanted in the temporal (8 electrodes), anterior basal occipital (4 electrodes), mid basal occipital (4 electrodes), posterior basal occipital (6 electrodes), and occipital interhemispheric (6 electrodes) regions for a total of 48 electrodes (Fig. 7.1a-b).

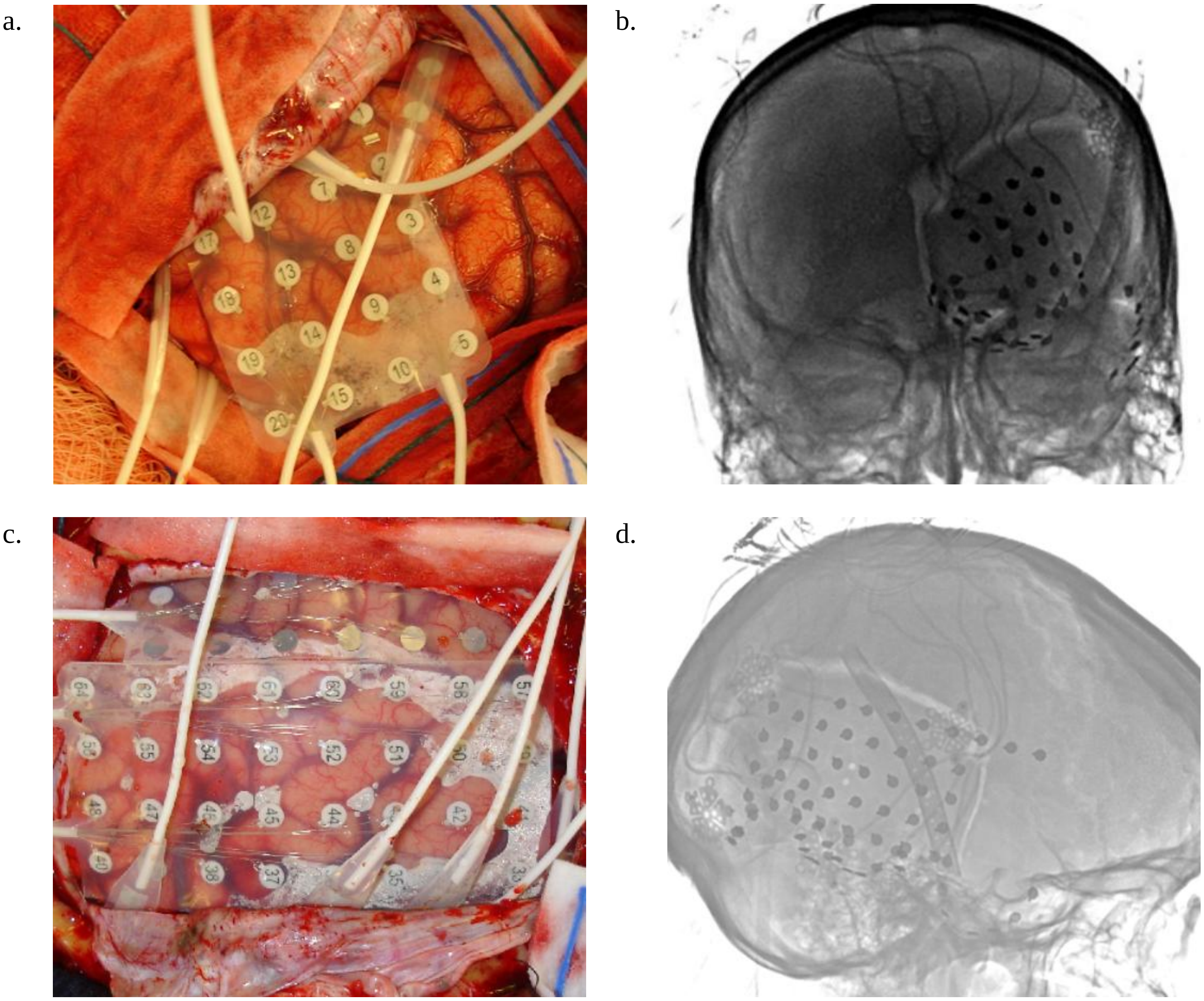


Figure 7.1 Images of implanted electrodes. (a) Photograph of the craniotomy and implanted electrodes for subject 1. (b) Intensity-inverted anteroposterior x-ray of subject 1 with implanted electrodes. (c) Photograph of the craniotomy and implanted electrodes for subject 2. (d) Intensity-inverted lateral x-ray of subject 2 with implanted electrodes.

The second subject was a 45 year old right-handed female with seizures that were not localizable, but predominant in the right temporal region. The patient had a 32-electrode temporal parietal occipital grid. There were also electrode strips implanted in the suprasylvian (7 electrodes), anterior temporal (7 electrodes), middle temporal basal (4 electrodes), middle posterior temporal basal (7 electrodes), posterior temporal basal (7 electrodes), and occipital basal (8 electrodes) regions of the right hemisphere for a total of 72 electrodes. The anterior temporal and middle temporal basal strips were omitted because our amplifier was limited to 64 channels, resulting in a total of 61 electrodes used in our analysis (Fig. 7.1c-d).

7.1.2. Data Collection

ECoG signals were recorded using four 16-channel g.USBamp amplifier systems (g.tec, Graz, Austria). Data was collected simultaneously with clinical monitoring by connecting the ECoG cables to a splitter at the time of implant. We could then connect our system to the splitter to acquire ECoG signals without interrupting the clinical monitoring system. Data was collected at 2400 Hz for the first subject and 512 Hz for the second and both were downsampled to 40 Hz for time-domain analysis. A notch filter at 60 Hz was used for both subjects. BCI2000 was used for data acquisition and experimental design (Schalk et al., 2004).

7.2. Methods

7.2.1. Classification

Analysis was performed offline using MATLAB (version 7.10.0, MathWorks, Inc, Natick, MA). Stepwise linear discriminant analysis (SWLDA) was used to determine the features for classification. Cross-validation was used for training, resulting in one of the trial words being used as the test set with the other two as the training set. The training set signals were then assigned labels based on whether the associated stimulus contained the target character.

For each stimulus, the ECoG data for the next 500 ms was treated as the feature vector, z_t^i for stimulus i for letter t in the sequence. SWLDA used a stepwise method to separate the available features into two groups based on whether the feature was significant in classification. At each step, the most significant feature above a threshold in the non-significant group was added to the significant group. Similarly, the least significant feature below a threshold in the significant group was removed from use in classification. The probabilities of adding and removing features were 0.1 and 0.15 respectively. These steps were repeated until the number of significant features reached a threshold of 60 features or until the feature groups reached equilibrium. These significant features were then stored in a weight vector, w. (Krusienski et al. 2006)

During testing, the dot product between the feature vector for each stimulus and the feature weight vector was taken to determine a score for that stimulus:

$$y_t^i = w \cdot z_t^i$$

The static method of classification was then performed by finding the character that had the highest total score for all of the stimuli it was associated with:

$$\arg\max_{x_t} \sum_{i:x_t \in \boldsymbol{A}_t^i} y_t^i$$

where $\boldsymbol{A}_t^i$ is the set of characters illumined for the ith flash for character t in the sequence.

The number of stimuli is predetermined on a per subject basis so that a decision will be made after a set number of flashes. We simulated this by varying the number of sets of flashes from 1 to 10. After the required number of flashes was reached, we made the decision and discarded the remaining data.

Discarding data in this way overemphasizes earlier trials and underutilizes our data in earlier decisions. We overcame this by taking advantage of the fact that each set of flashes was independent, so the order did not matter. For each letter, we created 1000 random permutations of the sets of stimuli and analyzed each independently, effectively bootstrapping the data. This gave us extra examples of the earlier decisions and used each stimulus equally.

7.2.2. Spectral Features

We computed spectral features of the ECoG signal following stimulus onset and adding them to the feature vector. The spectrogram was computed using the Chronux toolbox (Bokil et al. 2010) with a moving window of 100 ms and a step size of 50 ms. For each stimulus, the log power of the spectrogram values over the following 500 ms were used as features for classification. To reduce the total number of features, the values in the low gamma (30-70 Hz) and high gamma (70-200 Hz) were averaged.

The resulting features for each channel were then appended to the temporal feature to create a new feature vector, $\boldsymbol{z_t^i}$**.** SWLDA was then retrained on these new features to obtain a new weight vector that contains weights for each of the significant temporal and spectral features. As before, classification was performed by taking the dot product of this weight vector with the new feature vectors obtained for each stimulus.

7.2.3. Naïve Bayes

As a final step, NLP was integrated into the system to test the additional speedup from using knowledge about the language domain as described in (Speier et al. 2012). As in the previous analysis, cross-validation was used to obtain a training set for SWLDA. The means and standard deviations were then found for the scores for target and non-target stimuli. Assuming a normal distribution, the probability density function (PDF) for the likelihood probability were computed,

$$f\left(y_t^i|x_t\right) = \begin{cases} \dfrac{1}{\sqrt{2\pi\sigma_a^2}} e^{-\frac{1}{2\sigma_a^2}\left(y_t^i-\mu_a\right)^2} & if\ x_t \in \boldsymbol{A}_t^i \\ \dfrac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{1}{2\sigma_n^2}\left(y_t^i-\mu_n\right)^2} & if\ x_t \notin \boldsymbol{A}_t^i \end{cases}$$

where μ_a, σ_a^2, μ_n, and σ_n^2 are the means and variances of the distributions for the attended and non-attended scores, respectively.

A naïve Bayes classifier was then used to determine the conditional probability of a target character given a set of flash scores and the history of previous decisions.

$$P(x_t|\boldsymbol{y}_t, x_{t-1}, \dots, x_0) = \frac{1}{Z} P(x_t|x_{t-1}) \prod_i f\left(y_t^i|x_t\right)$$

where $P(x_t|x_{t-1})$ is the character bigram probability computed from the Brown corpus (Francis and Kucera, 1979), $f\left(y_t^i|x_t\right)$ are the PDFs for the likelihood probabilities and Z is a normalizing constant. Once the probability for a character reaches a threshold, that character is chosen and the system moves on to the next character in the sequence. The threshold probability was varied between 0 and 1 in increments of 0.01 and the value that maximized the bit rate was chosen for each subject.

7.3. Validation

7.3.1. Protocol

The system used a 6 x 6 character grid with row and column flashes, 50 ms flash duration, and 3.5 s pauses between selections. The first subject had an inter-stimulus interval (ISI) of 150 ms while the second had an ISI of 140 ms (as constrained by BCI2000 based on sampling rates). Each subject underwent three trials consisting of spelling a five letter word (subject 1: “HOUSE,” “BATCH,” and “ALOHA”; subject 2: “AVOID,” “BEING,” and “MIXED”) with 10 sets of 12 flashes (six rows and six columns) for each letter. The choice of target words for this experiment was independent of the trigram language model.

7.3.2. EEG Data

The dataset for this project was previously described in chapter 3 (Speier et al., 2014a). The subjects were six healthy male graduate students and faculty with normal or corrected to normal vision between the ages of 20 and 35. Only one subject (subject 2) had previous BCI experience. The system used a 6 × 6 character grid, row and column flashes, an ISI of 125 ms and a flash duration that varied between 31.25 and 62.5 ms. Each subject underwent nine trials consisting of spelling a five letter word with 15 sets of 12 flashes (six rows and six columns) for each letter. The choice of target words for this experiment was independent of the language model used in the naïve Bayes method. BCI2000 was used for data acquisition and analysis was performed offline using MATLAB (version 7.10.0, MathWorks, Inc, Natick, MA). Two analysis methods were compared: a static method where the number of flashes is predetermined and an naïve Bayes method that incorporates a character trigram language model and uses spectral features in response classification.

7.3.3. Evoked Response

In general, an evoked response was seen in electrodes starting at 100 ms after stimulus presentation (Fig. 7.2). This response was usually characterized by an initial negative response at 100 ms followed by a positive inflection at approximately 200 ms. This response is most pronounced in the electrodes near the occipital pole in the electrode grid in subject 1 (Fig. 7.2a-b). It can also be seen in the electrodes in the occipital basal strip in subject 2 (Fig. 7.2d). The electrodes in subject 2's temporal basal strip showed a small negative inflection in response to stimuli (Fig. 7.2e).

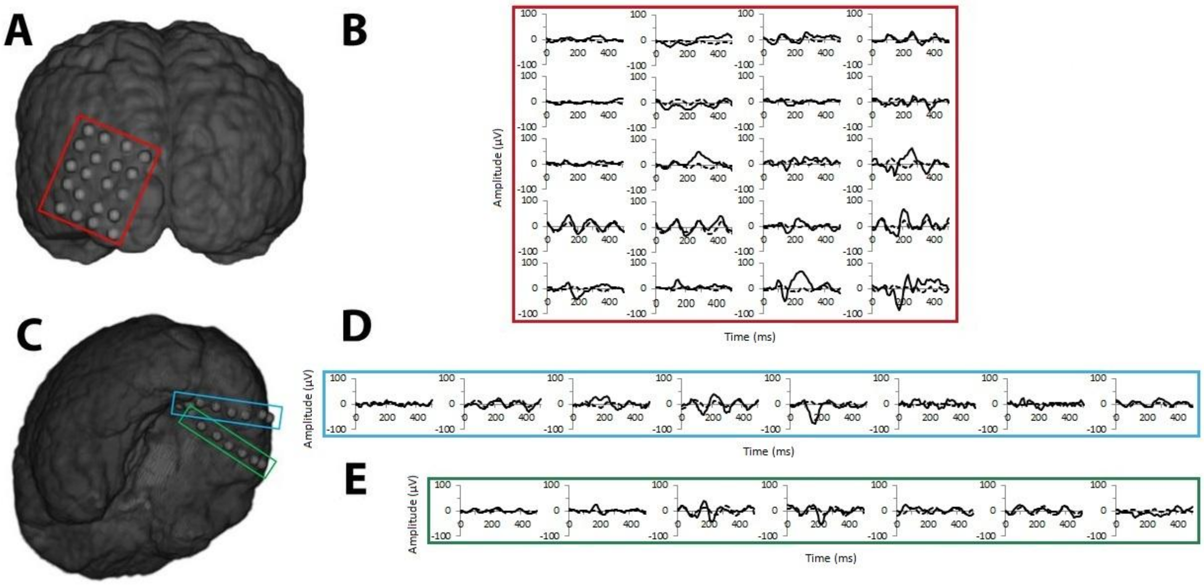


Figure 7.2 Average signal amplitude (in µV) over time (in ms) for response to attended (solid) and non-attended (dashed) stimulus for selected electrodes. For subject 1 (a), signals for the electrodes in the occipital grid are plotted (b). For subject 2 (c), signals for electrodes in the posterior temporal basal (d) and occipital basal € strips are plotted. Note that differentiation occurs between attended and non-attended signals in electrodes close to the occipital pole.

7.3.4. Spectral Analysis

Both subjects showed similar frequency responses to stimuli. Electrodes close to the occipital pole generally exhibited an increase in power about 100 ms after stimulus onset followed by a decrease in power after about 200 ms (Fig. 7.3). The first subject initially saw an increase in beta band power followed by an increase in low and high gamma power with a peak around 130 Hz (Fig. 7.3a). Subject 2 saw a peak in the alpha band after about 150 ms along with a general increase in low and high gamma power. In general, the first subject had a larger frequency response as his spectrogram contained a maximum of 8.6 dB compared to 5.8 dB for subject 2.

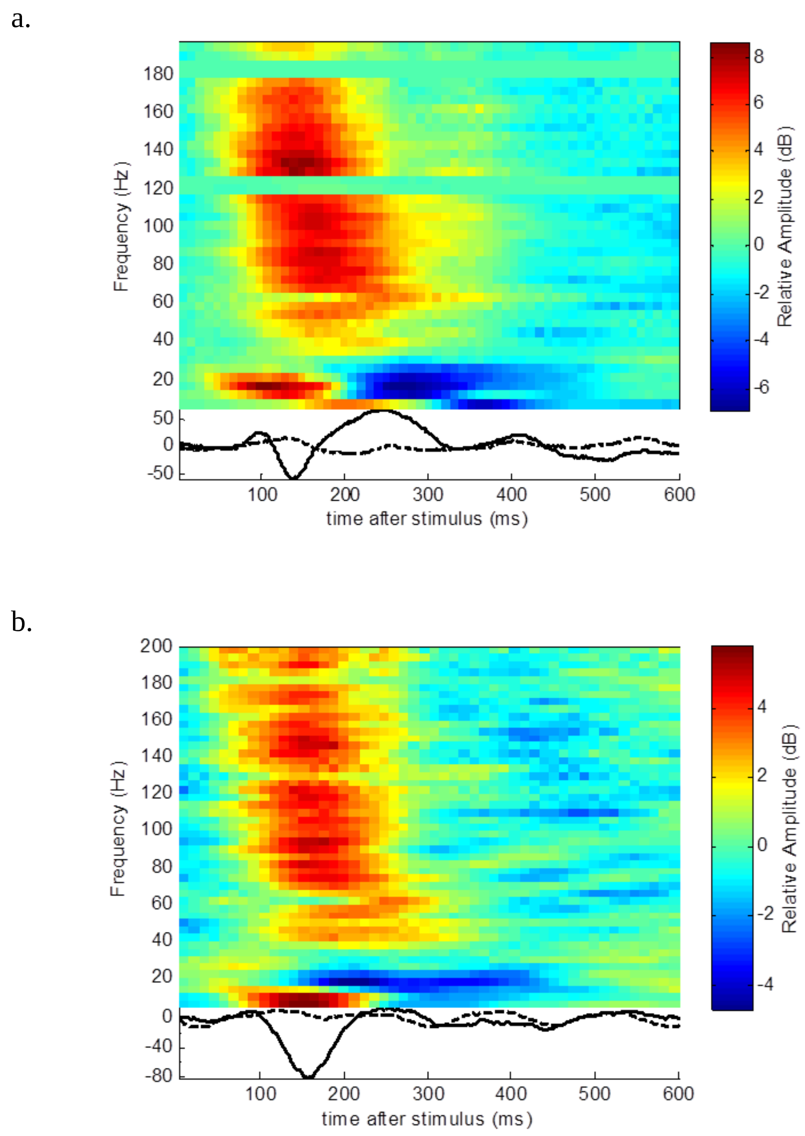


Figure 7.3 Log ratio of spectral response between target and non-target stimuli after onset along with average temporal response for a single electrode from each subject. The electrode from the occipital pole was used from subject 1 (a) and the middle electrode in the occipital basal strip from subject 2 (b), along with the average temporal response for attended (solid) and non-attended (dashed) stimuli.

7.3.5. BCI Performance

In our offline analysis, we found that the first subject was able to achieve an accuracy of almost 95% after two sets of flashes and over 98% after three sets using the standard classification method (Fig. 7.4a). His maximum bit rate of 41.02 was achieved after one set of flashes when he had a selection rate of 11.32 and an accuracy of 82.77% (Fig. 7.5a). The accuracy of the second subject was lower, reaching 73.33% after all 10 sets of flashes (Fig. 7.4b). After two sets of flashes, the subject achieved an accuracy of 59.03% and a selection rate of 8.72 characters per minute for a maximum bit rate of 18.26 (Fig. 7.5b). (Table 7.1)

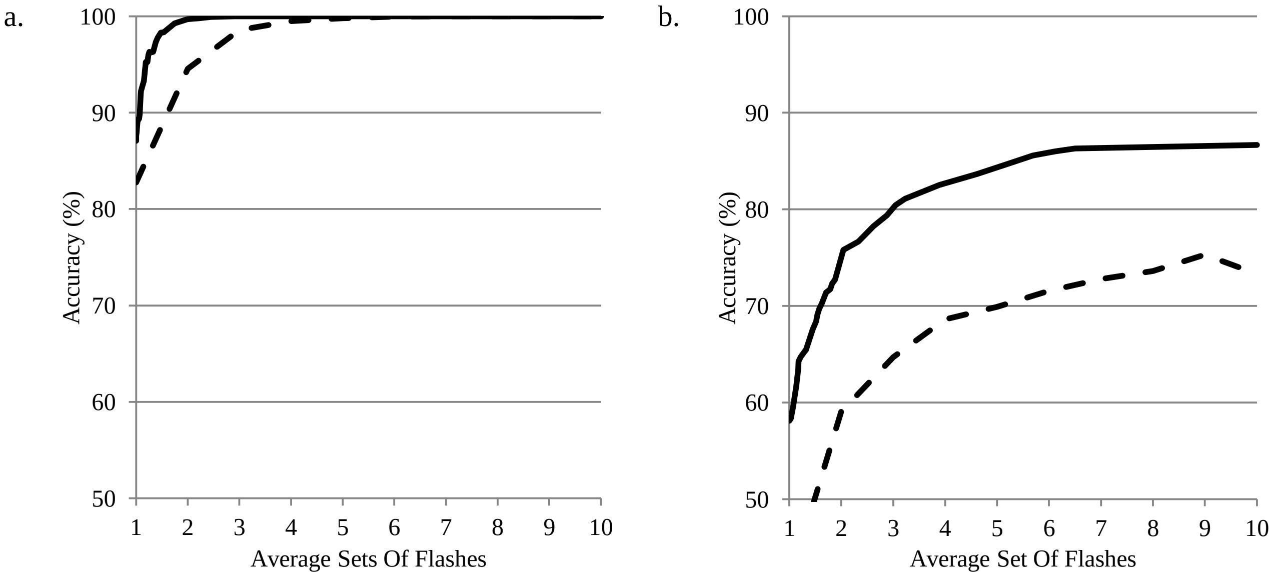


Figure 7.4 Subject accuracies. The average accuracies for each subject are plotted using the standard (broken curve) and naïve Bayes (full curve) methods versus the average number of sets of flashes required to make a decision.

Using spectral features alone for classification, the two subjects achieved maximum bit rates of 33.55 and 6.04 respectively, which were both lower than the values computed from temporal features alone. When the spectral features were combined with the temporal features for classification, the first subject's accuracy started slightly lower but rose faster, reaching 95.41% after two sets of flashes. The resulting maximum bit rate was 41.97, which was lower because it occurred after one set of flashes. The second subject's maximum bit rate increased slightly as she achieved a selection rate of 8.73 and an accuracy of 60.84 for a bit rate of 19.16. Overall, the average maximum bit rate rose slightly from 29.64 to 29.75.

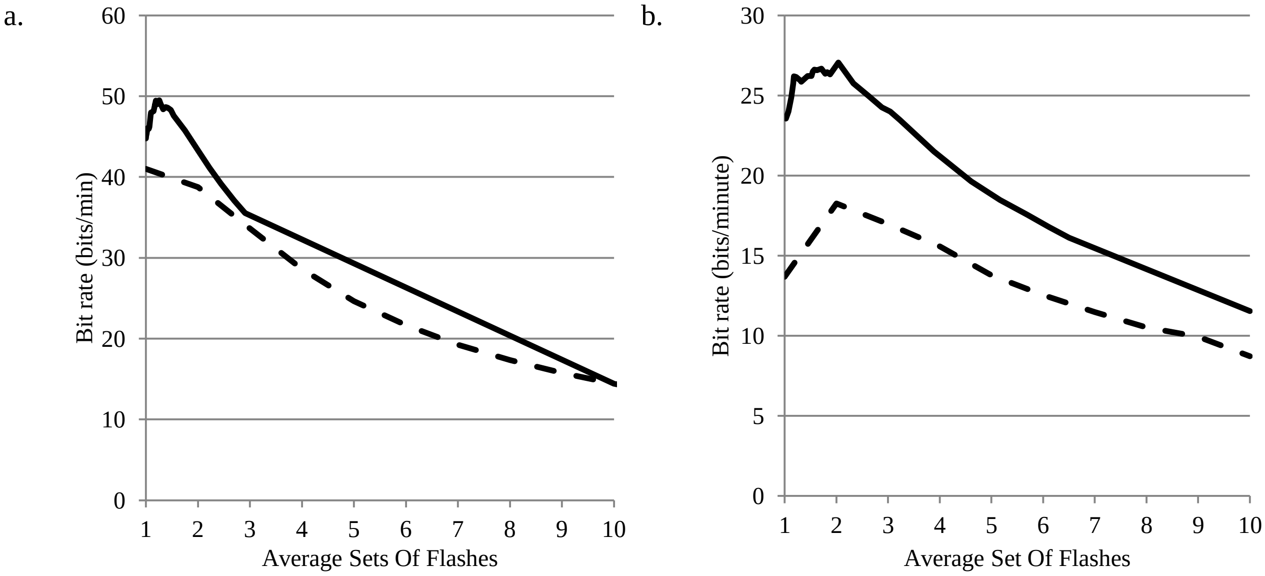


Figure 7.5 Subject bit rates. The average bit rates for each subject are plotted using the standard (broken curve) and naïve Bayes (cull curve) methods versus the average number of sets of flashes required to make a decision.

After integrating NLP in the temporal classifier, the first subject achieved a maximum bit rate of 48.81 with a selection rate of 10.53 and an accuracy of 95.09%. When spectral features were added to this classification, the selection rate became 10.41 and the accuracy rose to 96.31 for a maximum bit rate of 49.47 (Table 7.1). The second subject's maximum bit rate rose to 24.24 with a selection rate of 11.17 and 75.81% accuracy. Adding spectral features increased the bit rate to 27.05 with a selection rate of 8.64 and accuracy of 64.87 (Table 7.1).

Table 7.1 Results for the standard classification as well as the naïve Bayes method with spectral features optimized for bit rate.

	SR (selections/min)		Accuracy (%)		Bit Rate (bits/min)	
Subject	Standard	NB + Spec	Standard	NB + Spec	Standard	NB + Spec
1	11.32	10.41	82.77	96.31	41.02	49.47
2	8.73	8.64	59.03	75.81	18.26	27.05
AVG	10.02	9.52	70.9	86.06	29.64	38.26

Overall, the bit rate increased significantly with the inclusion of NLP and spectral features, as the average increased from 28.56 to 38.26 (p=0.006 using a paired t-test with 1 degree of freedom).

The average accuracy also increased significantly (p=0.03), while the selection rate had a non-significant decrease (p=.22).

7.3.6. Spatial Analysis

In order to find the cortical locations most useful in classification, spatial analysis was performed on the ECoG electrodes. For each patient, individual electrodes were used to classify each of the 150 sets of flashes, so that every electrode had an associated score based on its single flash classification accuracy. The electrodes were then plotted on the cortical surface and convolved with a Gaussian kernel to obtain a spatial classification accuracy map.

Individual electrode classification accuracies were tested for significance using a Pearson's chi-squared test with one degree of freedom, an alpha level of .05, Yates's correction for continuity, and Šidák correction for multiple comparisons (Fig. 7.6). The distribution of the total weight assigned by SWLDA across electrodes was also evaluated.

The channels for the first subject yielded accuracies that ranged from 0 to 59% with 16 exhibiting significant classification accuracies (Fig. 7.6a). Two of the electrodes on the posterior basal occipital strip, four of the six midline electrodes, and ten of the electrodes in the occipital grid were found to be significant, focused mainly around the occipital pole.

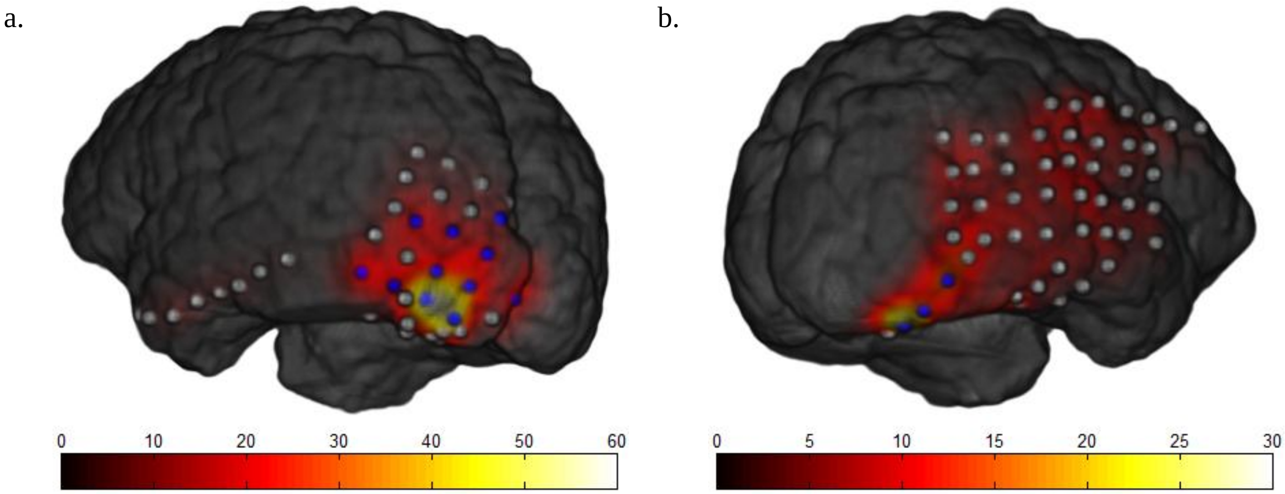


Figure 7.6 Event-related potential localization. The white dots show the locations of the electrodes and the cortical surface is color-coded based on the single-flash accuracy of the nearby electrodes. Electrodes that are statistically significant are highlighted in blue.

The channels for the second subject yielded accuracies that ranged from 0 to 27% with five exhibiting significant classification accuracies (Fig. 7.6b). In general, the temporal grid and the suprasylvian and middle posterior temporal basal strips were not useful as none of the electrodes had significant accuracy. The posterior temporal basal strip had one significant electrode with a maximum of 15% accuracy. The occipital basal strip had the best classification accuracy as it contained four significant electrodes, including the best performing electrode with an accuracy of 27%.

7.3.7. EEG comparison

To make the comparison fair, we reanalyzed the EEG dataset using the bootstrapping method and integrating spectral features (Table 7.2). Overall, the subjects in this study performed better than those in the EEG study. On average, the subjects in this study achieved a selection rate before integration of NLP of 10.02 selections/minute with an accuracy of 70.90%, resulting in a bit rate of 29.64. This was higher than the EEG averages bit rate of 21.69. After adding NLP to the analysis, the average selection was 9.52 selections/minute and the accuracy rose to 86.06%, resulting in an average bit rate of 38.26, which was higher than that for the EEG study (32.06).

Table 7.2 Results for the standard classification as well as the naïve Bayes method with spectral features optimized for bit rate for the EEG data set included for comparison.

	SR (selections/min)		Accuracy (%)		Bit Rate (bits/min)	
Subject	Standard	NB + Spec	Standard	NB + Spec	Standard	NB + Spec
1	9.23	10.51	86.22	94.4	35.86	48.05
2	7.5	8.78	73.54	87.46	22.34	34.95
3	9.23	9.68	79.17	93.84	31.05	43.76
4	5.45	7.54	74.76	81.56	16.69	26.67
5	5.45	6.79	54.83	66.05	10.14	16.99
6	4.29	6.86	72.58	76.84	12.5	21.95
AVG	6.86	8.36	73.52	83.36	21.43	32.06

The average improvement seen in this study is primarily due to the performance of the first subject. Using the standard analysis, the first subject had a significantly higher bit rate (p=0.004 using Wilcoxon rank-sum test) as the selection rate and accuracy were both better than the averages in the EEG data set. Using NLP, the first subject's bit rate was still higher than the average for the data set, but the difference was only marginally significant (p=0.017). In both cases, the first subject had a higher bit rate than any of the subjects in the EEG data set, although one subject was close after including NLP (subject 1).

The second subject’s bit rate was below the average for the EEG data set, but the difference was not statistically significant (p=0.38), and she performed better than three of the subjects from the EEG study. Her bit rate was still below the average for the EEG average after integrating NLP, but was still higher than the three subjects from the data set. Potential sources for variability and inferior performance were explored further (see Spatial Analysis below).

7.4. Discussion

7.4.1. System performance

ECoG-based P300 spellers can improve system performance relative to EEG-based systems, both with respect to accuracy and bit rate. This improved performance, however, is sensitive to electrode localization, as is evident in comparing the performance of the two subjects who had electrodes in different areas. Even without any modifications to the classification algorithm, the first subject achieved a higher accuracy with fewer stimuli than any of the subjects in the EEG data set or noted otherwise in the literature. The improved performance (bit rate) was despite using a higher ISI (150 ms) than the EEG study (125 ms), which meant that the selection rates and bit rates in this study would otherwise be lower than those for a similar subject in the EEG study. It is well known from the literature that stimulus timing parameters, particularly ISI, can significantly impact classification accuracy (Sellers et al., 2006; Lu et al., 2012). Optimizing system parameters for this system could lead to better improvements (see limitations and future directions).

The sensitivity to electrode localization is evident in the results of the second patient, as she performed worse than the average for the EEG data set. The spatial analyses for these two patients show one possible explanation for the difference between the two patients. The single flash accuracies of electrodes generally increased the closer the electrodes were to the occipital pole. While the first patient has electrodes through the occipital region, the second patient has only sparse coverage, consisting mainly of strips in the basal region. The difficulties shown by the second patient suggest that performance in an ECoG based BCI is sensitive to electrode placement. While this subject had more electrodes available and wider coverage than the first subject, her performance was significantly worse in both accuracy and selection rate. While we cannot test the locations outside of the ECoG coverage for this subject, the electrodes within the overlapping regions for the two patients had similar accuracies. This suggests that there could have been a strong ERP in the occipital region of the second subject that was not detected due to the electrode placement, accounting for some of the difference in performance. This is consistent with previous work which showed a drastic drop in performance when occipital electrodes were not used in analysis (Brunner et al., 2011).

The timing and localization of the signals detected in our analyses suggest that this system is based on visually evoked potentials (VEP). This is significant because VEP is highly dependent upon eye gaze, which severe locked-in patients may not be able to control (Riccio et al., 2012). It has been shown that performance drops drastically in EEG-based systems when the eye gaze is constrained (Brunner et al., 2010). The effect of eye gaze in an ECoG based system remains to be studied (see Future Work).

7.4.2. Spectral Features

Using spectral features alone, the classifier bit rates were lower than when using temporal features, but they were close on average to the EEG bit rate. Including spectral features with temporal features in the analysis increased the bit rates for both subjects when using the naïve Bayes analysis method. Further analysis using a larger sample size is required to definitively evaluate the value of incorporating spectral features in ECoG P300 analysis, but was unavailable at this time due to the unique and rare opportunities available to study such subjects.

The EEG dataset bit rate stayed relatively constant (31.97 to 32.06, p=.30) with the inclusion of spectral features. This could be because of the lower signal to noise ratio of EEG or because the lower sampling rate used in this dataset was too low to detect high frequency components.

7.4.3. Natural Language Processing

Integrating NLP into the classification resulted in significant increases in bit rate and therefore overall system performance. The magnitude of improvement (22% in the first subject), however, was lower than the 40-60% improvements reported in (Speier et al., 2012). We posit that this is due to a ceiling effect. Because the subject's performance was already at such a high level, NLP was not always needed. Because the subject was already achieving perfect accuracy the only possible improvement would be from reducing the time required. This is in contrast to the second subject who in fact experienced a 41% improvement with the inclusion of NLP as the bit rate improved from 19.16 to 27.05, consistent with those shown in the previous study.

7.4.4. Similar work

Brunner et al. also tested the P300 speller using ECoG signals (Brunner et al., 2011). Their study consisted of one patient and their analysis focused on a 6-electrode strip over the occipital cortex. Their study reported a substantially higher bit rate than either of the patients in this study. Our first subject achieved similar accuracy to theirs after the same number of stimuli, but our stimulus presentation rate was slower resulting in a lower bit rate. This would suggest that our presentation rate could have been increased without much loss in accuracy, although this was not tested. We leave finding the relationship between presentation rate and accuracy, as well as optimizing system parameters in an ECoG system, as future work.

Krusienski and Shih performed a P300 speller study on six epilepsy patients with ECoG electrodes implanted in the frontal, temporal, and parietal regions (Krusienski and Shih, 2011a). While five of the six patients were able to perform reasonably well, the average accuracy was shown to be close to that achieved by their EEG study and none of the subjects performed as well as subject 1 in this study or the subject in the Brunner study. This is likely due to the electrode placement as none of the subjects in their study had coverage of the occipital cortex.

Our experiment serves to verify the results of the previous studies showing the viability of using ECoG for the P300 speller as well as expanding on the spatial analysis by showing results in a broad cortical area. We also show additional improvements that can be made on these results using NLP and potentially spectral features to achieve even better P300 speller performance.

7.4.5. Limitations and future directions

While the performance of the first subject suggests that an ECoG based BCI system could yield better performance than current systems, more subjects need to be tested. This is in fact challenging as the placement of occipital grids and strips for epilepsy monitoring (which is the only model available for investigating this phenomenon) is relatively rare. Most such strips and grids are placed over frontal and temporal cortices. Also, showing the difference in performance within subjects would be a better indicator for the improvement from using this modality. In an ideal situation, we would be able to test the EEG system on a subject before the ECoG electrode placement surgery. Unfortunately, this was not possible in the current set of subjects studied.

Because the ECoG grids in our subjects were placed according to their seizure localization, the locations that we could use for our testing were limited. In particular, we were not able to test the parietal lobe, where the P300 signal occurs. We also were not able to test the system using data from the occipital pole of the second patient. Time constraints imposed by clinical parameters also prevented further system parameter optimization, such as stimulus length and ISI. Future studies will test patients with different electrode placements to more completely explore the electrode configuration space.

One of the major drawbacks for the approach described in this paper is the invasive surgery required to implant the electrodes. Even though the many ALS patients may be willing to undergo invasive surgery in order to obtain a better BCI (Huggins et al., 2011), it would still be preferable to limit the risks involved in surgery. While the added speed and accuracy of this system could make the procedure worthwhile to patients, we still want to minimize the extent of the surgery required. For this reason, a future direction for this work is to explore the use of epidural implants as a compromise between risk and benefit.

Finally, the subjects used in this study were not from the target patient population of "locked-in" patients. Because our subjects are not suffering from the same deficiencies, these results might not translate into identical improvement in the target population. For example, the spatial analysis in this study showed that the occipital region yielded the best results. Some severe “locked-in” patients are not able to move their eyes, however, so a system depending on occipital signals might not be appropriate. It remains to be seen if this system leads to actual improvements within our target population.

7.4.6. Conclusion

This study demonstrates that ECoG could potentially increase the accuracy and bit rate of a BCI system, but that the system is sensitive to the placement of the electrodes. Integrating NLP significantly improved the system and integrating spectral features showed an increase in bit rate that suggests that they could be useful as a supplement to traditional temporal features. Testing additional subjects, including those from our target patient population, could help verify that this improvement is enough to justify the required surgery and ultimately provide a better option for ALS patients.

8. Flashing Paradigm

In an EEG keyboard system, the user is presented with a flashing grid of characters $\underline{x} = \{x_1, x_2, \dots, x_n\}$ and the task is to decide the desired character based on whether a recognition signal occurred. In order to make a decision about which character was desired, each character needs to have a unique signature in the signal domain. When designing the system, there is a tradeoff between the amount of time required for deciding a character (dependent on the number of flashes) and the required accuracy in signal processing (dependent on the number of decisions that need to be made and the required precision in these decisions).

The simplest scheme would consist of flashing each character individually. This would only require one decision (which flash resulted in the "best" response), but would require n flashes (one for each character). The opposite end of the spectrum would have binary encoding. This method would require only $\log_2(n)$ flashes, but would require $\log_2(n)$ decisions, each of which would need to be correct in order to find the correct character.

Some middle ground between these two methods is preferable. The generally accepted compromise consists of flashing each character twice and triangulating the desired character based on the two positive responses. Depending on the flashing scheme used, the number of flashes and decisions varies. The only requirements on a scheme of this type are that each element x_i corresponds to two distinct flashes and that no two elements correspond to the same two flashes.

8.1. Row Column Paradigm

The row/column paradigm (RCP) puts the characters into a rectangular array and then flashes them in groups based on their rows and columns

Table 8.1 Character groups in the traditional row column flashing paradigm with n_c columns and n_r rows.

Group	Elements			
1:	x_1	x_2	$\cdots$	x_{n_c}
2:	x_{n_c+1}	x_{n_c+2}	$\cdots$	x_{2n_c}
$\vdots$	$\vdots$	$\vdots$		$\vdots$
n_r:	$x_{(n_r-1)n_c+1}$	$x_{(n_r-1)n_c+2}$	$\cdots$	$x_{n_r n_c}$
n_r+1:	x_1	x_{n_c+1}	$\cdots$	$x_{(n_r-1)n_c+1}$
n_r+2:	x_2	x_{n_c+2}	$\cdots$	$x_{(n_r-1)n_c+2}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
n_r+n_c:	x_{n_c}	x_{2n_c}	$\cdots$	$x_{n_c n_r}$

The RCP accomplishes the requirement because no two elements share both a row and a column. The number of flashes required is then $n_r + n_c$ which is optimized in a square grid where $n_r = n_c = \sqrt{n}$ where n is the number of elements in the character matrix, so the number of flashes required is $f_{rc} = \lceil 2\sqrt{n} \rceil$.

For example, if we want to flash a 36 character matrix consisting of the 26 letters and 10 digits, we need to create 12 groups:

Table 8.2 Character groups in a 6 x 6 grid containing the 26 English letters and 10 digits.

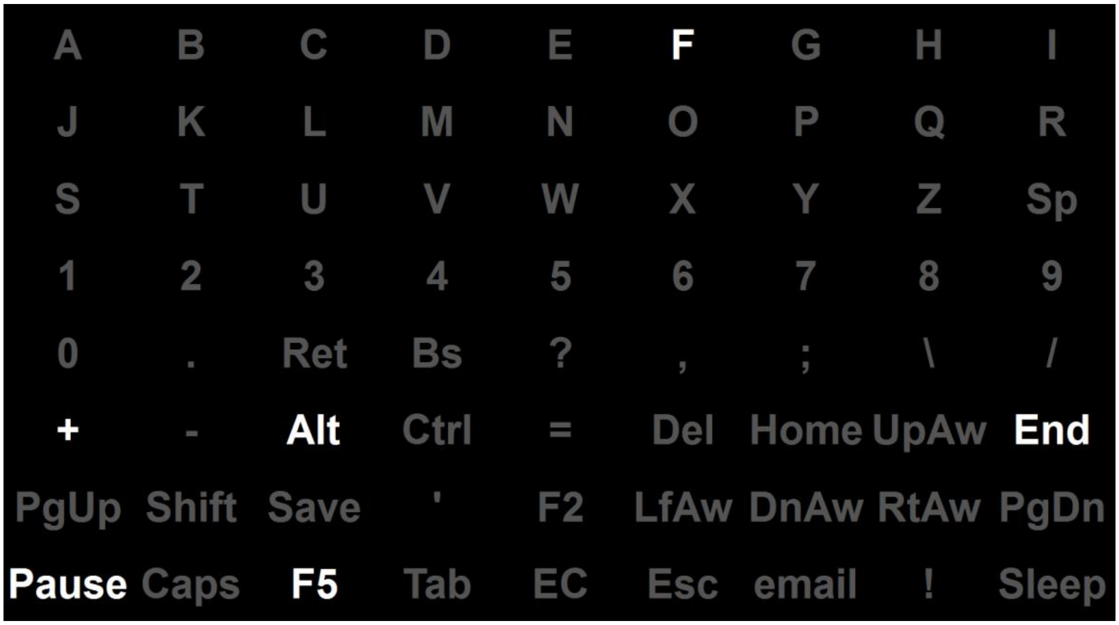
ABCDEF	GHIJKL	MNOPQR	STUVWX	YZ0123	456789
AGMSY4	BHNTZ5	CIOU06	DJPV17	EKQW28	FLRX39

Note that each character appears in two groups and no two characters appear together twice. Two decisions are then required to find the optimal row and column group.

There are two problems with this schema. First, it is not economical in the number of flashes required. Second, it allows elements to be flashed twice in succession, which has been shown to decrease the amplitude of the response signal (Citi et al., 2010).

8.2. Checkerboard Paradigm

Townsend and colleagues (2010) modified the standard RCP and suggested a novel checkerboard paradigm (CBP). This method splits the matrix into two alternating groups, similar to a checkerboard. The paradigm creates two randomized squares from these groups and stimuli consist of flashes of rows and columns of these squares. The system alternates between the groups, creating pseudo-random, sparse sets of characters rather than the row and column grouping of the RCP.



A	B	C	D	E	**F**	G	H	I
J	K	L	M	N	O	P	Q	R
S	T	U	V	W	X	Y	Z	Sp
1	2	3	4	5	6	7	8	9
0	.	Ret	Bs	?	,	;	\	/
+	-	**Alt**	Ctrl	=	Del	Home	UpAw	**End**
PgUp	Shift	Save	'	F2	LfAw	DnAw	RtAw	PgDn
Pause	Caps	**F5**	Tab	EC	Esc	email	!	Sleep

Figure 8.1 Example stimulus for the checkerboard flashing paradigm on a 72 character grid.

This method eliminates repetitions, but increases the number of flashes required in order to make a decision. The number of required flashes in this paradigm is

$$f_{cb} = 2\left\lceil 2\sqrt{\left\lceil \frac{n}{2} \right\rceil} \right\rceil$$

Using a 72 character grid and a standard classifier with five complete repetitions, which included ten target flashes, Townsend et al. (2010) showed a mean online accuracy improvement from 77% using the RCP to 92% using the CBP. Mean bit rate increased modestly, but significantly from 19 bits/min for the RCP up to 23 bits/min for the CBP.

8.3. Combinatorial Flashing

In order to find the minimal flashing paradigm, we need to first find the minimum number of flashes required to give each element in the matrix a unique flash signature. Given a number of flashes f, the number of elements that can correspond uniquely the a set of two flashes n is equal to the number of possible ways of choosing two unique flashes

$$n = \binom{f_{min}}{2} = \frac{f_{min}(f_{min} + 1)}{2}$$

Using the quadratic formula, we can then find the minimum number of flashes required for a given n

$$f_{min} = \frac{1 \pm \sqrt{1 + 8n}}{2}$$

Because f is necessarily a positive integer, this becomes:

$$f_{min} = \left\lceil \frac{1 + \sqrt{1 + 8n}}{2} \right\rceil$$

We can show that this is strictly less than the RCP in any realistic system (i.e. as long as $n > 4$) because single character flashing is at least as fast for $n \leq 4$. Here we assume $\sqrt{n}$ is integer valued so that f_{rc} is optimized.

$$f_{min} = \left\lceil \frac{1 + \sqrt{1 + 8n}}{2} \right\rceil < \left\lceil \frac{1 + \sqrt{9n}}{2} \right\rceil = \left\lceil \frac{1}{2} + \frac{3}{2}\sqrt{n} \right\rceil \leq 1 + \frac{3}{2}\sqrt{n} < 2\sqrt{n} = f_{rc}$$

Both methods are $O(\sqrt{n})$, but the difference in the constant can help with system speed. In the example of a square matrix of 36 characters, the minimum number of required flashes is 9 as opposed to the required 12 flashes for the RCP.

We can realize this minimum number of flashes by creating groups of size $f_{min} - 1 = \left\lceil \frac{\sqrt{1+8n}}{2} \right\rceil$. We create these groups by choosing $f_{min} - 1$ elements for the first group and then distributing them across the remaining $f_{min} - 1$ groups. We then complete the second group with the next $f_{min} - 2$ elements and distribute them across the remaining $f_{min} - 2$ groups. We repeat this process until all groups are filled. If $\frac{1+\sqrt{1+8n}}{2}$ is not integer valued, the number of elements will not work out completely. We can handle this instance by leaving some of the groups with fewer elements. This can be done either by carrying out the above process until there are no remaining characters, or by adding placeholders in either a random or systematic manner.

Table 8.3 Groups of characters in the minimal configuration for $\boldsymbol{n} = \binom{\boldsymbol{f_{min}}}{\boldsymbol{2}}$ characters.

Group	Elements				
1:	x_1	x_2	x_3	$\cdots$	$x_{f_{min}-1}$
2:	x_1	$x_{f_{min}}$	$x_{f_{min}+1}$	$\cdots$	$x_{2f_{min}-3}$
3:	x_2	$x_{f_{min}}$	$x_{2f_{min}-2}$	$\cdots$	$x_{3f_{min}-6}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

$f_{min}-1$:	$x_{f_{min}-2}$	$x_{2f_{min}-4}$	$x_{3f_{min}-7}$	$\cdots$	$x_{\binom{f_{min}}{2}}$
f_{min}:	$x_{f_{min}-1}$	$x_{2f_{min}-3}$	$x_{3f_{min}-6}$	$\cdots$	$x_{\binom{f_{min}}{2}}$

In the example of a 36 character matrix, the groups become

Table 8.4 Example minimal character groups for a 6 x 6 grid containing the 26 English letters and 10 digits.

Group	Elements							
1:	A	B	C	D	E	F	G	H
2:	A	I	J	K	L	M	N	O
3:	B	I	P	Q	R	S	T	U
4:	C	J	P	V	W	X	Y	Z
5:	D	K	Q	V	0	1	2	3
6:	E	L	R	W	0	4	5	6
7:	F	M	S	X	1	4	7	8
8:	G	N	T	Y	2	5	7	9
9:	H	O	U	Z	3	6	8	9

Note again that each character occurs in exactly two rows and that no two characters occur in the same two rows.

Decision making in this schema is more complicated. There are still two required decisions, but because we need to choose any two of the groups, the decisions are between all groups instead of a subset like in the RCP.

A variation on this method would be to randomize the groups. For this method, we create a set of placeholder variables $\underline{y} = \left\{y_1, y_2, \dots, y_{\binom{f_{min}}{2}}\right\}$ and create a mapping between $\underline{x}$ and $\underline{y}$ for every repetition. In the 36 character example from before, the flashing groups would be

Table 8.5 Randomized character groups for the first two sets of repetitions in the minimal case.

Repetition	Group	Elements							
1	1	0	T	S	X	2	6	E	B
1	2	0	F	8	N	4	1	D	I
1	3	T	F	Q	Z	H	5	U	L
1	4	S	8	Q	R	Y	K	7	G
1	5	X	N	Z	R	9	P	O	V
1	6	2	4	H	Y	9	3	A	J
1	7	6	1	5	K	P	3	M	C
1	8	E	D	U	7	O	A	M	W
1	9	B	I	L	G	V	J	C	W
2	1	O	K	D	S	M	1	5	0
2	2	O	G	Y	E	U	Z	8	T
2	3	K	G	W	L	C	9	I	4
2	4	D	Y	W	3	7	A	B	P
2	5	S	E	L	3	V	Q	H	6
2	6	M	U	C	7	V	R	2	X
2	7	1	Z	9	A	Q	R	F	N
2	8	5	8	I	B	H	2	F	J
2	9	0	T	4	P	6	X	N	J

This method requires each character to be analyzed individually. Only one decision is then required, but it is between all n characters. The correct character should correspond to a recognition signal every time, while incorrect characters will correspond to recognition signals anywhere between 0 and 50 percent of the time and less than $\frac{2}{f_{min}}$ on average. Instead of random mappings, systematic mappings between $\underline{x}$ and $\underline{y}$ can be used to ensure a more uniform distribution between incorrect characters, more even spatial distribution on the character matrix, or even to distinguish between characters with high scores.

This method does not address the issue of consecutive flashes of a single character, but rather increases the number of occurrences. Each stimulus in this method includes a character from the previous stimulus with the exception of the first stimulus in a repetition which can share anywhere from 0 to $f_{min} - 1$ characters with the previous stimulus depending on the mapping. We can solve the problem of the interstimulus repetitions through modifying the mapping algorithm.

8.4. Repetition Removal

We can remove all intrastimulus repetitions by shifting the second occurrence of each character back by one stimulus. This will increase the number of flashes by one and will ensure that no character flashes twice in succession.

$$f^* = f_{min} + 1 = \left\lceil \frac{3 + \sqrt{1 + 8n}}{2} \right\rceil$$

We can show that f^* is strictly less than the RCP in any system where $n > 16$ (again, $\sqrt{n}$ is assumed to be integer valued).

$$f^* = \left\lceil \frac{3 + \sqrt{1 + 8n}}{2} \right\rceil < \left\lceil \frac{3 + \sqrt{9n}}{2} \right\rceil = \left\lceil \frac{3}{2} + \frac{3}{2}\sqrt{n} \right\rceil \leq 2 + \frac{3}{2}\sqrt{n} < 2\sqrt{n} = f_{rc}$$

A plot of the number of flashes required versus the number of characters in the matrix shows that f^* grows slowly compared to f_{rc}.

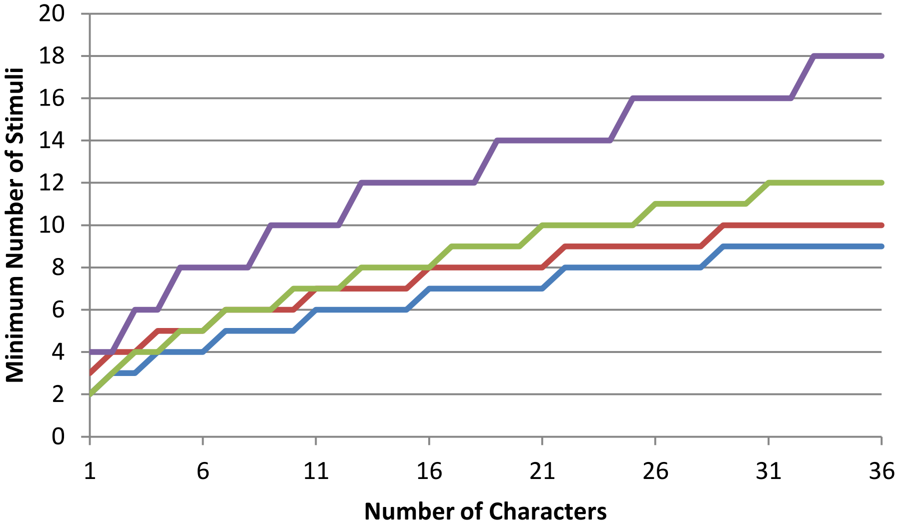


Figure 8.2 Minimal number of stimuli required for the three flashing schemes: checkerboard (purple), row column (green), minimal (blue), and minimal without repetition (red). Note that $\boldsymbol{f^*}$ (red) and $\boldsymbol{f_{min}}$ (blue) grow more slowly than $\boldsymbol{f_{rc}}$ (green) and the $\boldsymbol{f_{cb}}$ grows significantly faster.

This method requires the creation of a set of $\underline{y}$ variables and creating the groups:

Table 8.6 Groups of placeholders for the characters in the minimal configuration without repetitions for $\boldsymbol{n} = \binom{\boldsymbol{f_{min}}}{\boldsymbol{2}}$ characters.

Group	Elements					
1:	y_1	y_2	y_3	$\cdots$	y_{f^*-3}	y_{f^*-2}
2:	y_{f^*-1}	y_{f^*}	y_{f^*+1}	$\cdots$	y_{2f^*-6}	y_{2f^*-5}
3:	y_1	y_{2f^*-4}	y_{2f^*-3}	$\cdots$	y_{3f^*-10}	y_{3f^*-9}
4:	y_2	y_{f^*-1}	y_{3f^*-8}	$\cdots$	y_{4f^*-15}	y_{4f^*-14}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
f^*-2:	y_{f^*-4}	y_{f^*-7}	y_{f^*-11}	$\cdots$	$y_{\binom{f^*-1}{2}-5}$	$y_{\binom{f^*-1}{2}}$
f^*-1:	y_{f^*-3}	y_{2f^*-6}	y_{3f^*-10}	$\cdots$	$y_{\binom{f^*-1}{2}-4}$	$y_{\binom{f^*-1}{2}-2}$
f^*:	y_{f^*-2}	y_{2f^*-5}	y_{3f^*-9}	$\cdots$	$y_{\binom{f^*-1}{2}-1}$	$y_{\binom{f^*-1}{2}}$

Note that every variable occurs twice, no two variables occur in the same two groups, and no variable occurs in consecutive groups. It is possible for a single character to flash twice consecutively, however. This occurs when a character is assigned to one of the variables in group f^* in one repetition and then in group 1 in the following repetition. This can be solved by constraining the mapping so that this instance cannot occur.

This constraint is simple to enforce. We only need to keep track of the characters in group 10 during each repetition and changing the distribution for their assignments in the next repetition to make sure that they are not assigned to any of the variables in group 1. In practice, this means that we should assign the variables from group 10 first and either explicitly change their possible assignments or use rejection sampling where only valid mappings are accepted.

Table 8.7 Randomized character groups for the first two sets of repetitions in the minimal case without repetitions.

Repetition	Group	Elements							
1	1	J	F	D	9	4	7	B	5
1	2	P	Y	G	V	N	O	6	
1	3	J	M	Q	R	X	A	1	
1	4	F	P	3	2	H	U	E	
1	5	D	Y	M	C	L	I	T	
1	6	9	G	Q	3	W	S	8	
1	7	4	V	R	2	C	0	K	
1	8	7	N	X	H	L	W	Z	
1	9	B	O	A	U	I	S	0	
1	10	5	6	1	E	T	8	K	Z
2	1	L	M	0	F	W	2	C	D
2	2	1	3	B	N	V	Q	E	
2	3	L	Z	A	4	6	H	8	
2	4	M	1	5	O	U	G	X	
2	5	0	3	Z	K	P	I	J	
2	6	F	B	A	5	S	Y	R	

2	7	W	N	4	O	K	7	9		
2	8	2	V	6	U	P	S	T		
2	9	C	Q	H	G	I	Y	7		
2	10	D	E	8	X	J	R	9	T	

The resulting system addresses the shortcomings of the RCP because it only requires 10 flashes per repetition and does not allow repeat flashes of the same character. The inter-stimulus interval (ISI) could possibly be reduced in this case because there is more time between successive flashes of the same character. This could allow an even greater speedup than that which arises from the reduced number of stimuli.

Because the groups are created randomly, no characters will be more strongly linked to each other, so it might be easier to distinguish between the correct and incorrect characters. Reduced linkage could also be detrimental, however, because row/column information could be useful in signal processing even when an incorrect character is chosen. In a system that incorporates natural language processing, for instance, we might be able to combine language information with a spatial filter.

8.5. Validation

8.5.1. Protocol

An offline dataset was collected consisting of nine healthy graduate students and faculty with normal or corrected to normal vision between the ages of 20 and 35. Four stimulus paradigms were tested: a 6 x 6 character grid using the RCP, a 6 x 6 character grid using the combinatorial flashing without repetition paradigm, a 32 character grid using the checkerboard paradigm, and a 72 character grid using the checkerboard paradigm (Townsend et al., 2011). Each trial used an interstimulus interval (ISI) of 125 ms and a 32 electrode montage (Lu et al., 2013). Each subject underwent five trials of spelling a single five letter word for each paradigm. Ten sets of stimuli was presented to the subject for each letter. Offline analysis was performed using a hidden Markov model (Speier et al., 2014a). Results were evaluated using Information transfer rate (ITR) (McFarland et al., 2003) and mutual information given a two character history (MI_{2e}) (Speier et al., 2013c).

8.5.2. Results

Using the RCP, subjects selected 6.29 characters per minute on average at an accuracy of 84.44% (Table 8.9). When the checkerboard paradigm was used with a 32 character grid, the average accuracy was 76.89% and the selection rate was 6.59 characters per minute. The 72 character checkerboard paradigm produced an average selection rate of 7.13 characters per minute and an average accuracy of 80.44%. Using the combinatorial flashing paradigm, subjects achieved an average selection rate of 7.37 with an accuracy of 80.89.

Table 8.8 Selection rates, accuracies, and information transfer rates for the four stimulus presentation paradigms. Note that the 72 character checkerboard paradigm has a lower selection rate and accuracy than the combinatorial paradigm, but still has a higher bit rate.

	SR (sel/min)				ACC (%)				ITR (bits/min)			
Subject	RCP	CB32	CB72	COMB	RCP	CB32	CB72	COMB	RCP	CB32	CB72	COMB
A	5.35	6.41	5.73	5.26	96.00	80.00	100.00	96.00	22.29	29.70	25.10	25.45
B	7.62	6.65	6.12	7.94	96.00	92.00	100.00	92.00	36.39	29.01	34.88	34.22
C	5.36	2.93	6.43	8.93	52.00	20.00	48.00	36.00	9.89	2.94	11.92	3.62
D	4.51	5.57	3.95	6.31	84.00	88.00	88.00	88.00	21.50	26.99	20.73	19.66
E	6.05	8.01	9.87	5.63	76.00	64.00	60.00	72.00	23.12	20.96	19.94	16.27
F	8.33	10.10	10.10	10.12	92.00	88.00	100.00	100.00	31.54	42.01	64.93	54.56
G	6.12	2.69	5.79	6.20	72.00	76.00	72.00	64.00	23.90	12.82	27.56	14.01
H	7.03	8.43	6.32	6.61	92.00	96.00	88.00	100.00	37.67	36.72	26.13	36.88
I	6.24	8.55	9.88	9.32	100.00	88.00	68.00	80.00	34.93	29.43	37.72	31.65
Average	6.29	6.59	7.13	7.37	84.44	76.89	80.44	80.89	26.80	25.62	29.88	26.26

Using the combinatorial paradigm, subjects had higher selection rates and accuracies than when using the 72 character checkerboard paradigm, but the checkerboard paradigm had a significantly higher ITR (p=0.04). This is due to the reliance of ITR on the size of the grid of characters. Using the mutual information metric, the combinatorial method yielded a higher average bit rate (Table 8.10).

Table 8.9 Selection rates, accuracies, and mutual information based bit rates for the four stimulus paradigms.

	MI_{2e} (bits/min)			
Subject	RC	CB32	CB72	COMB
A	11.98	11.70	13.80	11.79
B	17.08	14.44	14.73	17.25
C	6.68	2.85	7.54	8.68
D	8.73	11.40	8.09	12.93
E	10.44	11.96	13.89	9.18
F	18.10	20.69	24.32	24.36
G	9.97	4.64	9.43	9.26
H	15.26	18.88	12.94	15.91
I	15.03	17.52	15.20	17.02
Average	12.59	12.68	13.33	14.04

8.6. Discussion

Using ITR as an evaluation metric, the 72 character checkerboard paradigm produced an average 25% improvement over the standard RCP as well as significantly higher values than the 32 character checkerboard (p=0.02) and the combinatorial paradigm (p=0.04). This result is consistent with previous studies (Townsend et al., 2010). However, this result is largely due to the assumption of ITR that all selections are equally likely. Character selections in a grid size of 72 are then seen as more complicated because of the increased number of options, even though most of those options are not characters and will not be chosen during spelling. As a result, the checkerboard paradigm can have a significantly higher ITR despite lower typing speed and accuracy.

The MI_{2e} metric takes selection probability into account, so it does not have the same bias towards larger grids. Using this metric, the checkerboard paradigm had a much more modest 5% increase over the RCP and the combinatorial paradigm had the highest bit rate. The differences between the results for the different paradigms were not statistically significant, indicating that additional trials must be conducted in order to draw conclusions about the optimal stimulus presentation paradigm.

8.6.1. Alternative designs

The extension from the f_{min} system to the f^* system can be further extended to ensure even greater gaps between successive flashes of the same character. For instance, if we shifted the second flash two stimuli later, the groups for the 36 character matrix become:

Table 8.10 Groups of placeholders for the characters in the minimal configuration for $\boldsymbol{n} = \binom{\boldsymbol{f_{min}}}{\boldsymbol{2}}$ characters when the constraint is added that no character can be repeated within a contiguous set of three flashes.

Groups	Elements							
1:	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8
2:	y_9	y_{10}	y_{11}	y_{12}	y_{13}	y_{14}	y_{15}	
3:	y_{16}	y_{17}	y_{18}	y_{19}	y_{20}	y_{21}		
4:	y_1	y_{22}	y_{23}	y_{24}	y_{25}	y_{26}		
5:	y_2	y_9	y_{27}	y_{28}	y_{29}	y_{30}		
6:	y_3	y_{10}	y_{16}	y_{31}	y_{32}	y_{33}		
7:	y_4	y_{11}	y_{17}	y_{22}	y_{34}	y_{35}		
8:	y_5	y_{12}	y_{18}	y_{23}	y_{27}	y_{36}		
9:	y_6	y_{13}	y_{19}	y_{24}	y_{28}	y_{31}		
10:	y_7	y_{14}	y_{20}	y_{25}	y_{29}	y_{32}	y_{34}	
11:	y_8	y_{15}	y_{21}	y_{26}	y_{30}	y_{33}	y_{35}	y_{36}

Now there are at least two stimuli between successive flashes of the same character at the price of one extra stimulus per repetition. Further constraints are necessary on the mappings as elements from group 11 in time step t cannot be mapped to groups 1 or 2 in time step $t + 1$ and elements from group 10 in time step t cannot be mapped to group 1 in time step $t + 1$.

Additional time between successive flashes is beneficial for two reasons. The first is that successive P300 signals do not interfere with each other. The second is that it allows us to reduce the ISI if the temporal resolution of the recognition signal is sufficient. In the f^* case, if we use an ISI of 100 ms, the minimum gap between successive flashes of the same character is 100 ms and one repetition takes 10*100=1000 ms. In the three flash shifting case, if we reduce the ISI to 80 ms, the time between successive flashes of the same character is 3*80=240 ms and one repetition takes 12*80=960 ms. This seems to be preferable, but it could result in problems with distinguishing between stimuli if the time resolution is not sufficient.

Other flashing schemata can be implemented with a tradeoff between number of required flashes and decision complexity. The above theory is extensible to a system where each character is defined by a unique set of three flashes. This would require fewer flashes, but would also mean more decisions as well as making it harder to increase the time between successive flashes. Obviously, a binary system would require the fewest number of flashes, but it would also require a much higher precision in the signal analysis step. It is probably more worthwhile to reduce the number of repetitions when signal processing improves rather than moving to one of these methods, but it is something that can be explored in the future.

8.6.2. Future Directions

One definite advantage of this model is the ability to modify the groups of characters that flash together. One possible use for this would be to try to reduce the number of characters that are spatially close in the character matrix. Another possible use would be to ensure that the most promising characters do not flash together in order to maximize the resolution between them. Partial decisions can also be made where some characters cease to flash once they have been ruled out, further reducing the required time.

9. UNSUPERVISED TRAINING

Neurological signals vary between people and over time, so each BCI session is usually preceded by a training session to calibrate the system. Because "locked-in" subjects are prone to fatigue, minimizing this training session could maximize the amount of time available for using the system for actual communication. Kaper et al. (2004) created a general classifier across subjects that eliminated the need for individual training, but drastically reduced performance. Panicker et al. (2010) created a semi-supervised approach that reduced the amount of training data needed by adapting the classifier during training. Spüler et al. (2012) have developed an unsupervised system that updates its classifier online from an initial general classifier in a c-VEP BCI system.

In this work, we build upon the systems created by Kaper et al. and Spüler et al. to create an unsupervised training method for the P300 speller. The first approach uses cross-validation to create a generic classifier on the other subjects in the dataset, similar to Kaper et al. In the second approach, BCI typing is modeled as a hidden Markov model (HMM) and the Baum-Welch algorithm is used to determine the best classifier given the observed EEG signal and prior knowledge of the language domain. These methods are compared offline with a supervised Viterbi classifier on a data set from 15 healthy subjects with the goal of demonstrating comparable results without the aid of labeled training data.

9.1. Baum-Welch Algorithm

9.1.1. Classifier

SWLDA is a classification algorithm that selects a set of signal features to include in a discriminant function using a labeled training set (Draper and Smith, 1981). This training set is usually obtained by running a session of copy spelling before using the system. In this study, two alternatives are presented. First, a training set is created using cross-validation across subjects to generate a general classifier. The second estimates labels using expectation maximization.

For input to SWLDA, each trial is assigned one of two classes: trials corresponding to flashes containing the attended character and those without the attended character. The algorithm uses ordinary least-squares regression to determine a linear combination of features which predict class labels for the training set. It achieves this by adding the most significant features in the forward stepwise analysis and removing the least significant features in the backward analysis step. These steps are repeated until either the target number of features is met or it reaches a state where no features are added or removed (Krusienski et al., 2006).

The score for each flash in the test set is then computed as the dot product of the feature weight vector, w, with the features from that trial's signal, z_t^i. Traditionally, the score for each possible next character, x_t, is computed as the sum of the individual scores for flashes that contain that character. After a predetermined set of stimuli, the character with the highest score is selected. It has been shown that scores can be approximated as independent samples from a Gaussian distribution given the target character (Speier et al., 2012):

$$p(\boldsymbol{z}_t|x_t) \propto \prod_i \phi\left({z_t^i}^T w; \mu_{a_t^i}, \sigma_{a_t^i}^2\right)$$

where ${a_t^i}^{(j)}$ is an indicator variable for whether ${x_t}^{(j)} \in A_t^i$ (i.e., ${a_t^i}^{(j)} = 1_{A_t^i}\left({x_t}^{(j)}\right)$ where A_t^i is the set of characters illuminated for the i th flash for character t in the sequence) and ${\mu_1}^{(j)}$, ${\sigma_1^2}^{(j)}$, ${\mu_0}^{(j)}$, and ${\sigma_0^2}^{(j)}$ are the means and variances for the attended and non-attended stimulus responses, respectively. The conditional probability of a target given the EEG signal and typing history can then be found

$$p(x_t|\boldsymbol{z}_t, x_{t-1}, \dots, x_1) \propto p(\boldsymbol{z}_t|x_t)p(x_t|x_{t-1}, \dots, x_1)$$

where $p(x_t|x_{t-1}, \dots, x_1)$ is the prior probability of character x_t given the selection history determined from a language model. In this study, the second order Markov assumption is used, so these probabilities can determined from relative trigram counts from the Brown English language corpus (Francis and Kucera, 1979).

$$p(x_t|x_{t-1}, \dots, x_1) \propto \frac{c(x_{t-2}, x_{t-1}, x_t)}{c(x_{t-2}, x_{t-1})}$$

where $c('a',' b', 'c')$ denotes the number of times the string “abc” occurs in the corpus.

9.1.2. Hidden Markov Models

Hidden Markov models are used to model Markov processes that cannot be directly observed, but can be indirectly estimated by state-dependent output. The goal of such systems is to determine the expected states in the Markov process that could have produced an observed output signal. A typed word is simply a sequence of states of the Markov process, $= (x_1, \dots, x_n)$. Because we cannot directly inspect the states of the process, we observe indirectly through the EEG signals. The EEG response is dependent only on the current state and governed by the conditional probability, $p(\boldsymbol{z}_t|x_t)$. The probability that the target character was x_t at time t can then be found by summing over the probabilities of all possible strings that contained x_t.

$$p(x_t|\boldsymbol{z}) = \sum_{x_{-t}} \prod_k p(\boldsymbol{z}_k|x_k)p(x_k|x_{k-1})$$

This equation is impractical because it is exponential in the size of the string. Instead, the forward-backward algorithm breaks the computation into two steps: computing the total probability into and state, , and computing the total probability out of a state, β.

$$\alpha_t(x_t) = \sum_{x_{t-1}} p(\boldsymbol{z}_t|x_t)p(x_t|x_{t-1})\alpha_{t-1}(x_{t-1})$$

$$\beta_t(x_t) = \sum_{x_{t+1}} p(\boldsymbol{z}_{t+1}|x_{t+1})p(x_{t+1}|x_t)\beta_{t+1}(x_{t+1})$$

The total probability of being in state x_t can then be found by multiplying the forward and backward probabilities.

$$p(x_t|\boldsymbol{z}) \propto \alpha_t(x_t)\beta_t(x_t)$$

9.1.3. Baum-Welch algorithm

The Baum-Welch algorithm is an expectation-maximization (EM) algorithm designed to find unknown parameters in an HMM. This iterative algorithm takes the known output sequence and transition probabilities, $p(x_t|x_{t-1})$, and tries to find the optimal target sequence, x.

In the expectation step, it is assumed that the weight vector and Gaussian parameters are known. In the initial step, these values are either given uninformative values (i.e., ${\mu_m}^{(0)} = 0$, ${\sigma_m^2}^{(0)} = 0$, $w^{(0)} = \boldsymbol{0}$) or are set based on a general model created through cross-validation across subjects. The forward-backward algorithm is then run to determine the probability distribution of X_t at each time step.

Traditionally, these probabilities are treated as fractional observations for the maximization step, but the SWLDA algorithm requires discrete input labels. Instead, samples ${x_t^i}^{(j)}$ are drawn from the probability distribution of ${X_t}^{(j)}$ and the ${a_t^i}^{(j)}$ values are computed

$${a_t^i}^{(j)} = 1_{A_t^i}\left({x_t^i}^{(j)}\right)$$

Given ${z_t^{im}}^{(j)}$, the maximization step finds the values of μ_m, σ_m^2, w which maximize the log likelihood of the data. The maximum value of $\boldsymbol{w}^{(j+1)}$ can be found through SWLDA by treating ${a_t^i}^{(j)}$ as supervised labels. The maximal values of μ_m, σ_m^2 can then be determined by maximizing the log likelihood function.

$${\mu_m}^{(j+1)} = \frac{1}{n_m} \sum_{\langle t,i \rangle} \delta_m^{{a_t^i}^{(j)}} {z_t^i}^T w^{(j+1)}$$

$${\sigma_m^2}^{(j+1)} = \frac{1}{n_m} \sum_{\langle t,i \rangle} \delta_m^{{a_t^i}^{(j)}} \left({z_t^i}^T w^{(j+1)} - {\mu_m}^{(j+1)}\right)^2$$

After every step in the Baum-Welch algorithm, the set of parameters is evaluated by finding the optimal character sequence. This is achieved by applying the Viterbi algorithm, which uses dynamic programming to find the optimal path to each character at each time point.

$$V_t(x_t) \max_{x_{t-1}} p(z_t|x_t) p(x_t|x_{t-1}) V_{t-1}(x_{t-1})$$

At the final time point, the highest probability path is chosen and the output string is determined by following the back pointers to the start.

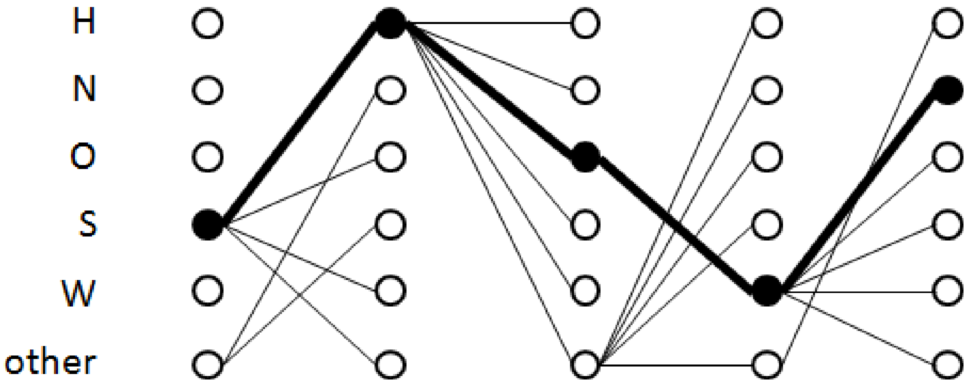


Figure 9.1 Simplified Viterbi trellis for subject B spelling the word "shown." The character 'N' has the highest value at the end of the trellis and back pointers are followed to determine the rest of the characters.

Each character in the resulting string is compared to the known target character to get accuracy values. These accuracies are then compared to the results using the general classifier alone and supervised results computed using the Viterbi algorithm. All statistical comparisons used paired t-tests with 14 degrees of freedom.

9.2. Validation

9.2.1. Dataset

The dataset for this project was previously described in chapter 3 (Speier et al., 2014a). The subjects in the offline dataset were 15 healthy graduate students and faculty with normal or corrected to normal vision between the ages of 20 and 35. Only one subject (subject F) had previous experience using a BCI for typing. The system used a 6 x 6 character grid, row and column flashes, and an interstimulus interval (ISI) of 125 ms. Each subject underwent between 8 and 10 trials consisting of spelling a five letter word with 15 sets of 12 flashes (six rows and six columns) for each letter.

9.2.2. Convergence

Using the uniform prior, 12 of the 15 subjects converged to classification accuracies above 90% (Fig. 9.2). This convergence occurred after between 3 and 9 steps in the Baum-Welch algorithm. Two subjects converged to 0% accuracy indicating a separate local maximum. A third subject converged to a local maximum around 25%, but eventually reached above 90% when the algorithm was run again for 20 iterations.

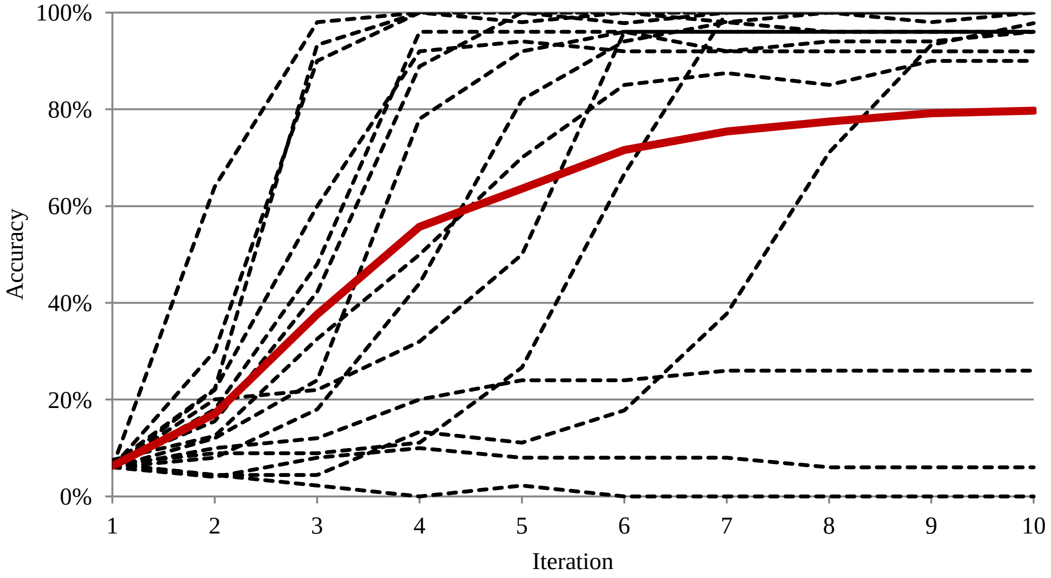


Figure 9.2 Classification accuracy for each subject (dashed lines) after each iteration of the Baum-Welch algorithm using a uniform prior. The average accuracy is denoted by the solid red line.

When probabilities determined through cross-validation were used as initial conditions, all subjects converged to probabilities above 90% within 3 steps of the Baum-Welch algorithm (Fig. 9.3). Three subjects reached the maximum after only one step and 11 reach above 90% after two steps of the algorithm. No local maxima were seen using informed prior probabilities.

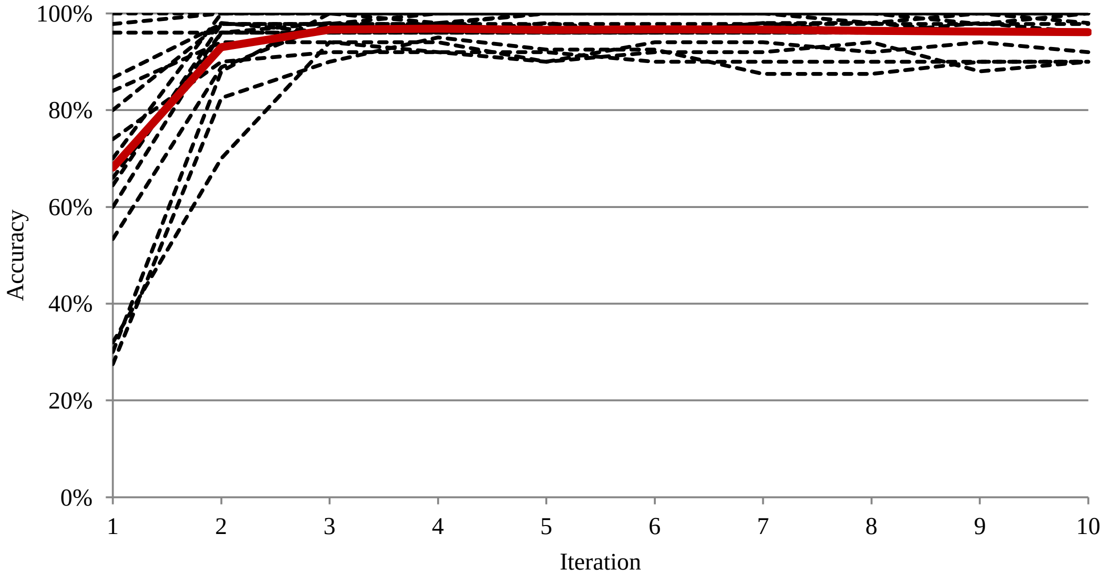


Figure 9.3 Classification accuracy for each subject (dashed lines) after each iteration of the Baum-Welch algorithm using a general prior from cross-validation. The average accuracy is denoted by the solid line.

9.2.3. Performance

Using the general classifier, the average accuracy was 68% with three subjects achieving accuracies below 35% (Table 9.1). After applying the Baum-Welch algorithm, the average accuracy improved to 97% after three iterations, which is significantly higher than the general classifier alone (p=0.00016). There was not a statistically significant difference between the results using the unsupervised Baum-Welch algorithm and the supervised Viterbi algorithm for classification (p=0.12) with 12 of the 15 subjects performing at least as well using the unsupervised classifier.

Table 9.1 Average accuracy using the general classifier, unsupervised training, and traditional supervised training.

Subject	General	Unsupervised	Supervised
A	98%	100%	100%
B	64%	98%	98%
C	28%	90%	95%
D	100%	100%	100%

E	53%	98%	100%
F	87%	98%	98%
G	70%	100%	100%
H	96%	96%	96%
I	60%	98%	98%
J	66%	96%	96%
K	32%	94%	92%
L	80%	96%	98%
M	84%	94%	94%
N	30%	100%	100%
O	74%	92%	92%
Average	68%	97%	97%

9.3. Discussion

The drastic decrease in accuracy from supervised training to the general classifier is consistent with the previous study by Kaper et al. (2004). Using the Baum Welch algorithm, all subjects achieved results comparable to those using supervised methods; the Viterbi algorithm performed consistently regardless of whether the training was supervised or unsupervised. This suggests that the Baum-Welch algorithm can be reliably used to train a BCI system. Convergence occurred for each subject within three iterations of the algorithm, which suggests the possibility of using it in an online system (see future directions).

It is interesting that the algorithm found the globally optimal solution for most subjects using the uniform prior. This shows that the bias in the language model was enough for the algorithm to learn the pattern in the data. This could be utilized in situations where a general model is not possible either because electrode locations are not standard (such as in electrocorticography when electrode distributions are irregular) or because the physiological state of the subject makes comparisons to other subjects impractical.

9.3.1. Limitations and Future Directions

There are two different ways to implement this work. The first is to run the method early on in a session to replace the training step of a BCI experiment. This could require less training data as it is able to adapt from previous sessions. Testing the amount of data needed for this algorithm to work has not yet been tested.

The other iteration would be to run it continuously during p300 speller use, in order to constantly update the weight vector, creating an adaptive system and possibly correcting errors. This algorithm could run in parallel on a separate thread from the main experiment or during the gap between trials. This was not possible in this study because analyses were performed offline and retrospectively. Adapting this method into online systems remains as future work.

9.3.2. Conclusion

Unsupervised training is possible using the Baum-Welch algorithm as classification accuracies can be achieved that are not significantly different from those using supervised methods. Choosing the initial conditions for the classification is important to speed up convergence and avoid local maxima.

10. DISCUSSION

Despite significant research interest, the P300 speller has remained relatively unchanged since its initial presentation in 1988. New classification algorithms have been presented (Kaper et al., 2004; Xu et al., 2004; Serby et al., 2005), but none produced substantially better results than the original SWLDA classifier (Krusienski et al., 2006) and SWLDA remained widely used. As recently as 2011, systems were presented showing average bit rates of 19.28 bits per minute (Ryan et al., 2011). While adequate accuracies were generally feasible, new systems presented selection rates that were up to an order of magnitude lower than the 15 characters per minute that the target patient population desired (Brunner et al., 2011; Huggins et al., 2011).

Incorporation of language domain knowledge provides a new source of information for classification that was previously ignored. Natural language contains structure that has been closely studied for applications such as medical informatics, speech recognition, and machine translation. Utilizing this information has provided a shift in focus from incrementally improving classification accuracy of raw signals to applying machine learning techniques to better incorporate domain knowledge. The work presented here has built on this idea to create a system with an average bit rate of 37.31 in online trials, close to double that of systems using traditional classifiers.

Another factor that may play an even larger role in widespread adoption is the usability of the system. Only minimal attention had been given to the actual practicality of the system, but concerns about system cost and maintenance as well as time requirements for setup and training could potentially present barriers for end users. Minimizing hardware requirements is an important step towards making the system more widely available. Reducing the channels required for using the system makes the system more affordable and simplifies setup and maintenance demands on caregivers, and unsupervised training can eliminate the training session and allow online adaptation to changes in the user's state. Further research into the usability of the system is necessary, particularly in the target population, to determine other barriers to adoption for patients.

Even with the improvements presented here, significant work remains to meet the needs of "locked-in" patients. The accuracy of the particle filtering method meets patients' specifications, but selection speed is still insufficient (Huggins et al., 2011). Furthermore, testing has only been conducted on healthy subjects and it has yet to be seen how well these results translate into the target population. This work serves as a starting point for integrating language information into the P300 speller. There are several ways to build off of this work that will provide additional improvements, possibly leading to a system sufficient for widespread use in BCI communication for "locked-in" patients.

10.1. Future Directions

10.1.1. Improved Language Models

Moving beyond n-grams is an important step in modeling language for BCI communication, but character level models are still relatively primitive in the domain of natural language processing. Ideally, a P300 system would take advantage of previous words and sentence structure in order to better predict target words. A first step would be to incorporate a simple bigram model of parts of speech into the word model from chapter 4 (Table 10.1).

Table 10.1 Examples of 100 character strings of generated text using five language models.

Language Model	Example generated text
Uniform characters	bg9qrhabyoa2x09zffbddevrm umow w4mf9xm1j4mo2nt1yzx0lhmi57qowi7paebej elohj nisqf866aqh5ogoqkdi8mnscj
Character Unigrams	roult ihves4nlcf tsietaakee9swd tst ed tisolpcfgeomece pc nn roosyicdhngdee le l s doh ar eac ote
Character Trigrams	uto mer surd ruct in pliclerinally he farted houth ors dickthe I flas aget com may had thell staprou
Word Unigrams	afraid none glowing from knew gushed jenkins munich that collar excess feathertop out ingested the o
Part of Speech Bigrams	a same earlier heaven to imply only other power removed for his combination was that decency is here

As before, probabilistic finite state automata are generated based on word frequency in a corpus of text. However, this model trains separate automata for each part of speech. This can be achieved by labeling the words in the training corpus either manually or automatically using a parser (Klein and Manning, 2002). The Brown corpus provides manual tags for the entire corpus, allowing it to be implemented directly in this method (Francis and Kucera, 1979).

As in the chapter 4, the start state for each part of speech corresponds to a blank string. Each state then links to every state that has a string that is a superstring that is one character longer. Thus, in the noun model, the state “c” will link to the states “ca” and “cu” (Fig. 10.1). States that represent complete words contain links to a root node based on bigram probabilities between parts of speech.

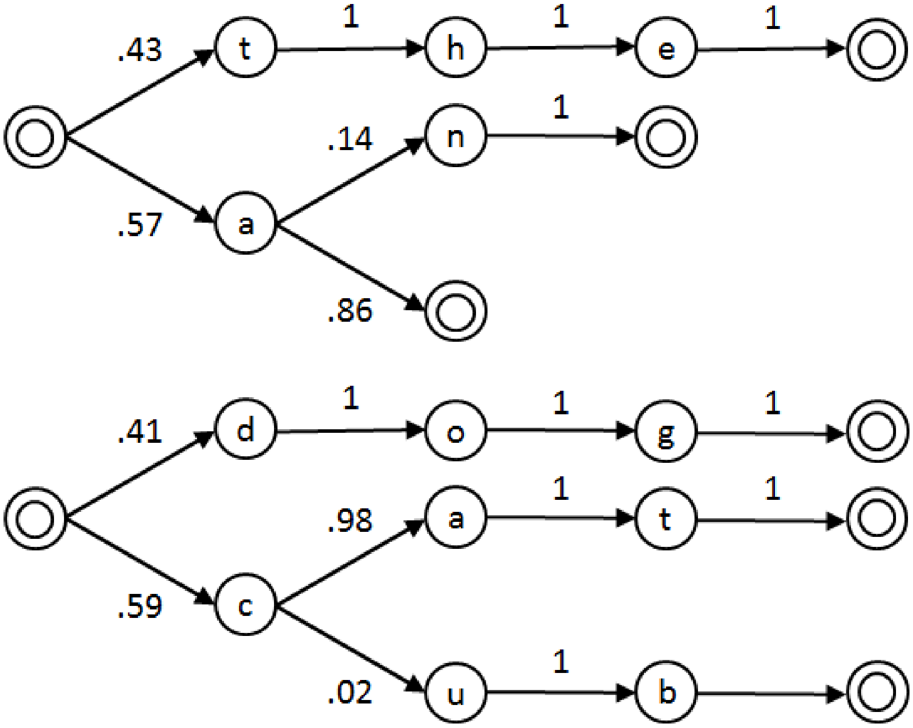


Figure 10.1 Simplified language model consisting of two parts of speech: article (top) and noun (bottom). A part of speech is first chosen based on bigram probabilities and a word is then found by traversing the probability tree.

As before, the transition probabilities are determined by the relative frequencies of words starting with the states' substrings in the Brown English language corpus (Francis and Kucera, 1979).

$$p(x_t|x_{0:t-1},t) = \frac{c(x_0,\dots,x_{t-1},x_t,t)}{c(x_0,\dots,x_{t-1},t)}$$

where $c('a','b',t)$ denotes the number of occurrences of a word tagged with part of speech t that starts with the string "ab" in the corpus. Similarly, the probability that a word ends and the model transitions back to the root is the ratio of the number of occurrences of complete words consisting of a string to the total number of occurrences of words beginning with that string.

$$p('\ '|x_{0:t-1},t) = \frac{c(x_0,\dots,x_{t-1},'\ ',t)}{c(x_0,\dots,x_{t-1},t)}$$

where $c(a, b, ' \ ', t)$ is the number of occurrences of the word “ab” that are tagged with part of speech t in the corpus.

In general, using the part of speech for a word should strengthen the prior probability, resulting in faster decisions. This method was tested offline using the dataset from chapter 2. These trials consisted of single word spelling, so they could not use the bigram part of speech probabilities. Instead, the parts of speech of the target words were given to the classifier as known values. This is not realistic, but it shows the value of the information provided by the part of speech in classifying BCI communication output. In reality, these results are likely upper bounds on the true value of including this information.

Table 10.2 Selection rate, accuracy, and ITR for subjects in offline analysis for the standard classifier and particle filtering with and without the part of speech given.

	SR (sel/min)			ACC (%)			ITR (bits/min)		
Subject	SWLDA	PF	POS	SWLDA	PF	POS	SWLDA	PF	Pos
A	8.07	10.47	12.07	93.33	97.78	95.56	36.13	51.33	56.50
B	5.33	7.33	6.87	88.89	88.89	100.00	21.82	30.05	35.49
C	3.96	6.71	10.59	80.00	85.00	72.50	13.54	25.43	30.83
D	9.96	11.75	10.63	88.89	95.56	100.00	40.82	54.99	54.95
E	3.98	6.54	9.11	95.56	91.11	88.89	18.64	27.98	37.32
F	7.21	10.63	11.67	95.56	95.56	100.00	33.76	49.74	60.33
G	5.33	9.33	8.88	94.00	92.00	100.00	24.18	40.64	45.90
H	8.66	9.77	11.70	88.00	96.00	94.00	34.86	46.13	53.06
I	5.08	10.68	9.30	78.00	80.00	98.00	16.68	36.55	45.80
J	4.78	6.55	7.61	90.00	98.00	100.00	20.03	32.28	39.33
K	3.98	7.81	10.26	84.00	84.00	82.00	14.80	29.01	36.58
L	5.62	7.71	12.04	92.00	98.00	76.00	24.47	37.98	37.86
M	4.13	6.16	7.54	82.00	90.00	80.00	14.72	25.82	25.80
N	7.60	9.58	10.99	94.00	100.00	100.00	34.48	49.55	56.81
O	4.40	9.43	10.78	88.00	82.00	76.00	17.71	33.62	33.88
average	5.87	8.70	10.00	88.82	91.59	90.86	24.44	38.07	43.36

Using the part of speech tags, classification accuracy remained relatively constant at 90.86%, but selection rate increased significantly to 10.0 characters per minute (p=0.004), resulting in a significantly higher average bit rate of 43.36 (p=0.00003). Eleven of the 15 subjects had an increased bit rate of at least 5 bits per minute and the remaining four saw only minor changes.

Bigrams on part of speech are only a starting point for incorporating additional language information in BCI communication. Targeted corpora can help reduce the vocabulary of the system, focusing on words that the individual user is likely to type rather than all words in the language. This would make training higher dimensional models such as word n-grams feasible, which should yield an even larger improvement. Including contextual information can also help guide prior probabilities in a language model. If the system can track the user's activities, for instance, it can anticipate likely commands from the user and apply a higher probability to the associated language. Learning can also be integrated into the system as prior output can be fed back into the language model in order to adapt to the user's vocabulary and speech patterns.

10.1.2. Continuous Trials

Current systems have a defined period for each character and stimulus presentation is paused when a character is selected. Traditionally, this pause is around 3.5 ms, which accounts for over half of the time required for a selection in the particle filtering method presented in chapter 4. Eliminating this delay could theoretically double the speed of the system without making significant changes to the stimulus presentation or classification algorithm.

There are several reasons for the delay between characters. First, there is a lag between stimulus presentation and classification due to the nature of the neural signal (600-800 ms) and computation time for classification (varies between algorithms), so this pause allows the system to analyze all of the stimulus responses before making a decision. Second, subjects are traditionally expected to correct all typing errors, so this pause allows them to check the output, determine whether it needs to be corrected, and choose the next cell in the grid, either the next character or a backspace. Finally, it makes classification simpler because it sets the number of stimuli for a given selection, removing the need for the system to determine when a user moved to the next character.

These reasons have been mitigated by some of the methods presented in this work. While the lag between stimulus presentation and classification is unavoidable, the dynamic classification used in all of the methods presented here requires continual updates of the probability distribution over the set of characters. Thus, decisions can be made without waiting for all data to be acquired and analyzed. Also, the hidden Markov model and particle filtering algorithms automatically correct many user errors, reducing the need for users to be concerned about manual correction.

The particle filtering algorithm can be modified to detect character transitions automatically. Currently, particles are all projected forward in the model at once when a threshold is reached. Instead, transitions in the model can be estimated by a homogeneous Poisson process where the length of time a user spends on a single character follows and exponential distribution

$$f(\Delta t; \lambda) = \lambda e^{-\lambda \Delta t}$$

Once it is determined that a particle will transition, the transition probabilities are found as before

$$p(x_t|x_{0:t-1}) = \begin{cases} \dfrac{c(x_{0:t}) + T(x_{0:t-1})p(x_t|x_{t-1}, x_{t-2})}{c(x_{0:t-1}) + T(x_{0:t-1})} & c(x_{0:t-1}) > 0 \\ p(x_t|x_{t-1}, x_{t-2}) & c(x_{0:t-1}) = 0 \end{cases}$$

A possible shortcoming of this model is that it continues to collect data while a subject transitions to a new character. The amount of time required for this transition and the characteristics of the signal during the transition are not well studied, so it is unknown whether this will interfere with classification.

Another potential challenge with this system is the incorporation of dynamic stopping. A large part of the performance improvement in the algorithms presented here comes as a result of being able to spend more time on difficult classifications (Table 10.3). Without defined trail lengths, it becomes difficult to inform the user that the system needs more information in order to make a classification. Visual cues such as changing cell background color have been explored, but they have distracted users. While the gains from removing the delay between trials may offset the loss of dynamic stopping, it would be preferable to be able to incorporate both into the P300 speller's classifier. Incorporating dynamic stopping with continuous trials remains as future work.

Table 10.3 Selection rates accuracies and information transfer rates for standard classification, particle filtering, and particle filtering without dynamic classification.

	SR (sel/min)			ACC (%)			ITR (bits/min)		
Subject	SWLDA	PF	PF STATIC	SWLDA	PF	PF STATIC	SWLDA	PF	PF STATIC
A	9.23	10.47	9.23	71.11	97.78	93.33	26.04	51.33	41.30
B	7.50	7.33	7.50	66.67	88.89	97.78	19.07	30.05	36.77
C	3.53	6.71	7.50	80.00	85.00	70.00	12.08	25.43	20.62
D	7.50	11.75	9.23	95.56	95.56	100.00	35.10	54.99	47.72
E	3.87	6.54	6.32	82.22	91.11	86.67	13.87	27.98	24.75
F	6.32	10.63	9.23	86.67	95.56	88.89	24.76	49.74	37.82
G	5.45	9.33	6.32	82.00	92.00	92.00	19.45	40.64	27.52

H	7.50	9.77	7.50	84.00	96.00	96.00	27.86	46.13	35.42
I	3.53	10.68	9.23	92.00	80.00	76.00	15.38	36.55	29.02
J	4.80	6.55	6.32	80.00	98.00	96.00	16.43	32.28	29.83
K	4.29	7.81	5.45	80.00	84.00	98.00	14.67	29.01	26.87
L	5.45	7.71	7.50	84.00	98.00	94.00	20.26	37.98	34.01
M	5.45	6.16	7.50	68.00	90.00	74.00	14.31	25.82	22.57
N	7.50	9.58	7.50	82.00	100.00	100.00	26.75	49.55	38.77
O	3.87	9.43	6.32	86.00	82.00	84.00	14.97	33.62	23.46
average	5.72	8.70	7.51	81.35	91.59	89.78	20.07	38.07	31.76

10.1.3. Patient Studies

All testing in the studies presented in this work have been completed with healthy volunteers. While this serves as a proof of concept of the methods, it does not necessarily translate directly into improved performance in the target population. Several studies that have explored efficacy of P300 speller systems in the target population have shown that the system can work, but often has decreased performance (Townsend et al., 2010; McCane et al., 2014). These studies have been limited to traditional classifiers, so it is possible that some of this deficit can be overcome by the improvements presented here.

Eye gaze restriction is a particular obstacle that exists in the target population. Studies have shown that performance drops significantly when subjects are unable to move their eyes (Brunner et al., 2010; McCane et al., 2014), which is the case in “locked-in” subjects who have the most to gain from a BCI system. While the studies addressing the classifier are likely robust to this situation, many of the usability studies such as the flashing paradigm or electrode placement will be highly affected by the state of the users. Future studies should test these methods in the target population to determine the true effects of these modifications in a realistic user setting.

11. CONCLUSION

The P300 speller has the potential to restore communication to "locked-in" patients, allowing them to interact with their environments and rejoin society. Unfortunately, limitations in the efficacy and usability of the existing system have prevented it from being widely adopted. System performance remains a major concern as typing speed and accuracy are still well below the benchmarks defined by Huggins et al. (2011) and while decreasing hardware costs have made these systems slightly more attainable for patients, the level of expertise required to set up and maintain these systems has makes them impractical for most potential users.

This work has focused on addressing these two aspects of BCI communication in order to help meet the needs of the target population. Here, it is shown that application of machine learning and natural language processing concepts in the BCI domain can provide the means for drastically improving performance while reducing the cost and time burdens of a complicated system. The methods presented here are meant as a primer for future advancement in these areas; the language models and machine learning methods covered in this work are not meant to be a finished product, but rather a proof of concept of the potential for advancement in the field. The future directions described in the discussion demonstrate the possibility of further advancements with relatively small modifications.

The field in general is starting to move towards the utilization of language information in BCI communication. Kindermans et al. (2014) have modified their offline unsupervised training method to create the first online spelling system that does not require a training session. Delgado Saa et al. (2015) recently presented a P300 speller that utilized a word-level language model rather than simple n-grams. Mainsah et al. (2015) have published the first study that tested the use of a language model in BCI communication when used by the target ALS population. Each of these studies shows promising results, moving closer to meeting the needs of the “locked-in” population.

While work remains to be done in order to reach the goals outlined by Huggins et al. (2011), the progress shown here demonstrates their attainability and provides a starting point for the road to creating a widely available solution for “locked-in” patients.

References

Allison B, Pineda J (2006) Effects of SOA and flash pattern manupulations on ERPs performance, and preference. Implications for a BCI system. *Psychophysiology* 59:127-140.

Acqualagna L, Blankertz B (2013) Gaze-independent BCI-spelling using rapid serial visual presentation (RSVP). *Clin Neurophysiol* 124:901-908.

Ball L, Nordness A, Fager S, Kersch K, Mohr B, Pattee G (2010) Eye-gaze access to AAC technology for people with amyotrophic lateral sclerosis. *Journal of Medical Speech-Language Pathology* 18:11-23.

Baxter S, Enderby P, Evans P, Judge S (2012) Barriers and facilitators to the use of high technology augmentative and alternative communication devices: a systematic review and qualitative synthesis. *Lang Commun Disord* 47(2):115-129.

Blankertz B, Curio G, Müller K (2001) Classifying Single Trial EEG: Towards Brain Computer Interfacing. *Adv neural Inf Proc Syst* 14:157-164.

Blankertz B, Dornhege G, Krauledat M, Schröder M, Williamson J, Murray-Smith R, Müller K. (2006) The Berlin brain-computer interface presents the novel mental typewriter hex-o-spell. *Proceedings of the 3rd International Brain-Computer Interface Workshop and Training Course, Graz, Austria*.

Bokil H, Andrews P, Kulkarni JE, Mehta S, Mitra P (2010) Chronux: A Platform for Analyzing Neural Signals. *J Neurosci Methods* 192(1):146-151.

Brunner P, Ritaccio AL, Lynch TM, Emrich JF, Wilson JA, Williams JC, Aarnoutse E, Ramsey N, Leuthardt E, Bischof H, Schalk G. (2009) A practical procedure for real-time functional mapping of eloquent cortex using electrocorticographic signals in humans. *Epilepsy Behav* 15:278-286.

Brunner P, Joshi S, Briskin S, Wolpaw J, Bischof H, Schalk G (2010) Does the 'P300' speller depend on eye gaze? *J Neural Eng* 7:056013.

Brunner P, Ritaccio AL, Emrich JF, Bischof H, Schalk G (2011) Rapid communication with a "P300" matrix speller using electrocorticographic signals. *Front Neur* 5:5.

Casagrande A, Jarmolowska J, Turconi M, Fabris F, Paolo P (2013) PolyMorph: A P300 Polymorphic Speller. *Lecture Notes in Computer Science* 8211:297-306.

Cecotti H (2010) A self-paced and calibration-less SSVEP-based brain-computer interface speller. *IEEE Trans on Neural Systems and Rehabil Eng* 18:127-133.

Cecotti H, Rivet B, Congedo M, Jutten C, Bertrand U, Maby E, Battout J (2011) A robust sensor-selection method for P300 brain-computer interfaces. *J Neural Eng* 8:016001.

Chen S, Goodman J (1996) An empirical study of smoothing techniques for language modeling. *Proc 34th ACL* 310-318.

Citi L, Poli R, Cinel C (2010) Documenting, modeling and exploring P300 amplitude changes due to variable target delays in Donchin's speller. *J Neural Eng* 7.

Colwell K, Ryan D, Throckmorton C, Sellers E, Collings L (2014) Channel selection methods for the P300 Speller. *J Neurosci Methods* 232C:6-15.

Croft R, Gonsalvez C, Gabriel C, Barry R (2003) Target-to-target interval versus probability effects on P300 in one- and two-tone tasks. *Psychophysiology* 40:322-328.

D'albis T, Blatt R, Tedesco R, Sbattella L, Matteucci M (2012) A Predictive Speller Controlled by a Brain-Computer Interface Based on Motor Imagery. *ACM Trans on Comp-Hum Interact* 19:20.

Delgado Saa J, de Pesters A, McFarland D, Cetin M (2015) Word-level language modeling for P300 spellers based on discriminative graphical models *J Neural Eng* 12:026007.

Draper H, Smith H (1981) *Applied Regression Analysis* (2nd ed.) New York: Wiley.

Duda R, Hart P, Stork D (2001) *Pattern Classification* (2nd ed.) New York: Wiley.

Dunlop M, Crossan A (2000) Predictive text entry methods for mobile phones. *Personal Technologies* 134-143.

Eng J, Eisner J (2004) Radiology Report Entry with Automatic Phrase Completion Driven by Language Modeling. *RadioGraphics* 24:1493-1501.

Farwell L, Donchin E (1988) Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalogr Clin Neurophysiol* 70:510-523.

Fazel-Rezai R, Ahmad W (2011) P300-based Brain-Computer Interface Paradigm Design. *Recent Advances in Brain-Computer Interface Systems* InTech

Fazel-Rezai R, Gavett S, Ahmad W, Rabbi A, Schneider E (2011) Comparison among Several P300 Brain-Computer Interface Speller Paradigms. *Clin EEG Neurosci* 42:209-213.

Francis W, Kucera H (1979) *Brown Corpus Manual*.

Furdea A, Halder S, Drusienski D, Bross D, Nijboer F, Birbaumer N, Kübler A (2009) An auditory oddball (P300) spelling system for brain-computer interfaces. *Psychophysiology* 46:617-625.

Gonsalves C, Polich J (2002) P300 amplitude is determined by target-to-target interval. *Psychophysiology* 39:388-396.

Gordon N, Salmond D, Smith A (1993) Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEEE Proc-F* 140:107-113.

Guger C, Daban S, Sellers E, Holzner C, Krausz G, Carabalona R, Gramatica F, Edlinger G (2009) How many people are able to control a P300-based brain-computer interface (BCI)? *Neurosci Lett* 462:94-98.

Hart-Davis G (2011) Entering and Editing Text in Your Documents. In: Office 2010 Made Simple. New York: Springer. Pp. 165-197.

Hoffmann U, Vesin J, Ebrahimi T, Diserens K (2008) An efficient P300-based brain-computer interface for disabled subjects. *J Neurosci Methods* 167(1):115-125.

Huggins J, Wren P, Gruis K (2011) What would brain-computer interface users want? Opinions and priorities of potential users with amyotrophic lateral sclerosis. *Amyotroph Lateral Scler* 12:319-324.

Jelinek F (1998) *Statistical Methods for Speech Recognition*. Cambridge, MA: MIT Press.

Jin J, Horki P, Brunner C, Wang X, Neuper C, Pfurtscheller G (2010) A new P300 stimulus presentation pattern for EEG-based spelling systems. *Biomed Tech* 55_203-210.

Jin J, Allison B, Sellers E, Brunner C, Horki P, Wang X, Neuper C. (2011) Optimized stimulus presentation patterns for an event-related potential EEG-based brain-computer interface. *Med Biol Eng Comput* 49:181-191.

Kaper M, Meinicke P, Grossekathoefer U, Lingner T, Ritter H (2004) BCI competition 2003 – data set IIb: support vector machines for the P300 speller paradigm. *IEEE Trans Biomed Eng* 50:1073-1076.

Kaper M, Ritter H (2004) Generalizing to new subjects in brain-computer interfacing. *Conf Proc IEEE Eng Med Biol Soc* 6:1073-1076.

Kaufmann T, Hammer E, Kübler A (2011) ERPs contributing to classification in the P300 BCI. *5th International Brain-Computer Interface Conference* Graz, Austria: University of Technology.

Kaufmann T, Volker S, Gunesch L, Kubler A (2012) Spelling is just a click away- a user-centers brain-computer interface including auto-calibration and predictive text entry. *Front Neurosci* 6:72.

Kellis S, Miller K, Thomson K, Brown R, House P, Greger B (2010) Decoding spoken words using local field potentials recorded from the cortical surface. *J Neural Eng* 7:056007.

Kindermans P, Verschore H, Verstaeten D, Schrauwen B (2012) A P300 BCI for the Masses: Prior Information Enables Instant Unsupervised Spelling. *NIPS* 25:719-727.

Kindermans P, Verstraeten D, Schrauwen B (2012) A Bayesian model for exploiting application constraints to enable unsupervised training of a P300-based BCI. *PLOS ONE* 7(4):e33758.

Kindermans P, Verschore H, Schrauwen B (2013) A unified probabilistic approach to improve spelling in an event-related potential based brain-computer interface. *IEEE Trans Biomed Eng* 60(10):1-1.

Kindermans P, Schreuder M, Schrauwen B, Müller K, Tangermann M (2014) True zero-training brain-computer interfacing – an online study. *PLOS One* 9:e.102504.

Klein D., Manning D (2003) Fast Exact Inference with a Factored Model for Natural Language Parsing. *NIPS* 3-10.

Krusienski D, Sellers E, Cabestaing F, Bayoudh S, McFarland D, Vaughan T, Wolpaw J (2006) A comparison of classification techniques for the P300 speller. *J Neural Eng* 3:299-305.

Krusienski D, Sellers E, McFarland D, Vaughan T, Wolpaw J (2008) Toward enhanced P300 speller performance. *J Neurosci Methods* 167(1):15-21.

Krusienski D, Shih J (2011) Control of a visual keyboard using an electrocorticographic brain-computer interface. *Neurorehabil Neural Repair* 25(4):323-331.

Krusienski D, Shih J (2011) Spectral Components of the P300 Speller Response in Electrocorticography. *IEEE EMBS Conference on Neural Engineering* FrA1.4.

Lauer R, Peckham P, Kilgore K, Heetderks W (2000) Application of cortical signals to neuroprosthetic control: a critical review. *IEEE Tran Rehabil Eng* 8:205-208.

Laureys S, Pellas F, Van Eeckhout P, Ghorbel S, Schnakers C, Perrin F, Berré J, Faymonville M, Pantke K, Damas F, Lamy M, Moonen G, Goldman S (2005) The lock-in syndrome: what is it like to be conscious but paralyzed and voiceless? *Progress in Brain Research* 150:495-511.

Lee S, Lim S (2011) Predicting text entry for brain-computer interface. In Future Information Technology. Springer:309-312.

Lenhardt A, Kaper M, Ritter HJ (2008) An adaptive P300-based online brain-computer interface. *IEEE Trans Neural Syst Rehabil Eng* 16:121-130.

Li Y, Bahn S, Nam C, Lee J (2013) Effects of Luminosity Contrast and Stimulus Duration on User Performance and Preference in a P300-Based Brain-Computer Interface (BCI). *Int J Hum-Comput Int* Taylor & Francis.

Liu J. Monte Carlo Strategies in Scientific Computing. New York:Springer, 2001.

Liu Y, Zhou Z, Hu D (2010) Comparison of Stimulus Types in Visual P300 Speller of Brain-Computer Interfaces. *Proc 9th IEEE Int Conf on Cognifive Informatics* 273-279.

Lu J, Speier W, Hu X, Pouratian N (2012) The effects of stimulus timing features on P300 speller performance. *Clin Neurophysiol* 124(2):306-314.

Mainsah B, Collins L, Colwell K, Sellers E, Ryan B, Caves K, Throckmorton C (2015) Increasing BCI communication rates with dynamic stopping towards more practical use: an ALS study *J Neural Eng* 22:837-846.

Manning C, Schütze H (1999) *Foundations of Statistican Natural Language Processing*. Cambridge, MA: MIT Press.

McCane L, Sellers E, McFarland D, Mak J, Carmack C, Zeitlin D, Wolpaw J, Vaughan T (2014) Brain-computer interface (BCI) evaluation in people with amyotrophic lateral sclerosis. *Amyotroph Lateral Scler* 15:207-215.

McFarland D, Sarnacki W, Wolpaw J (2003) Brain-computer interface (BCI) operation: optimizing information transfer rates. *Biol Psychol* 63:237-251.

McFarland D, Sarnacki W, Townsend G, Vaughan T, Wolpaw J (2011) The P300-based brain-computer interface (BCI): effects of stimulus rate. *Clin. Neurophysiol* 122:731-7.

Miller KJ, Schalk G, Fetz EE, den Nijs M, Ojemann JG, Rao RG (2010) Cortical activity during motor execution, motor imagery, and imagery-based online feedback. *Proc Natl Acad Sci* 107:4430-4435.

Mohri M (1996) On some applications of finite-state automata theory to natural language processing. *Natural Language Engineering* 2:61-80.

Orhan U (2014) RSVP keyboard: an EEG based BCI typing system with context information fustion.

Park J, Kim K (2012) A POMDP Approach to Optimizing P300 Speller BCI Paradigm. *IEEE Trans Neural Syste Rehabil Eng* 20:584-594.

Panicker R, Puthusserypady S, Sun Y (2010) Adaptation in P300 Brain-Computer Interfaces: A Two-Classifier Cotraining Approach. *IEEE Trans Biomed Eng* 57(12):2927-2935.

Patel S, Azzam P (2005) Characterization of N200 and P300: Selected Studies of the Event-Related Potential. *Int J Med Sci* 2:147-154.

Pierce J. (1980) *An Introduction to Information Theory* (New York: Dover)

Riccio A, Mattia D, Simione L, Olivetti M, Cincotti F (2012) Eye-gaze independent EEG-based brain-computer interfaces for communication. *J Neural Eng* 9:045001.

Ryan D, Frye G, Townsend G, Berry D, Mesa-G S, Gates N, Sellers E (2011) Predictive spelling with a P300-based brain-computer interface: increasing the rate of communication. *Int J Hum-Comput Interact* 27:69-84.

Salvaris M, Sepulveda F (2009) visual modifications on the P300 speller BCI paradigm. *J Neural Eng* 6:046011.

Samizo E, Yoshikawa T, Furuhashi T (2013) A study on application of rb-arq considering probability of occurrence and transition probability for p300 speller. In Foundations of Augmented Cognition. Springer:727-733.

Schalk G, McFarland D, Hinterberger T, Birbaumer N, Wolpaw J (2004) BCI2000: A General-Purpose Brain-Computer Interface (BCI) System. *IEEE Trans Biomed Eng* 51(6):1034-1043.

Schmidt E (1980) Single neuron recording from motor cortex as a possible source of signals for control of external devices. *Ann Biomed Eng* 8:339-349.

Schreuder M, Rost T, Tangermann M (2011) Listen, you are writing! Speeding up online spelling with a dynamic auditory BCI. *Front Neurosci* 5:112.

Sellers E, Krusienski D, McFarland D, Vaughan T, Wolpaw J (2006) A P300 event-related potential brain-computer interface (BCI): The effects of matrix size and inter stimulus interval on performance. *Biological Psychology* 73:242-252.

Sellers EW, Vaughan TM, Wolpaw JR (2010) A brain-computer interface for long-term independent home use. *Amyotroph Lateral Scler* 11:449-455.

Serby H, Yom-Tov E, Inbar G (2005) An improved P300-based brain-computer interface. *IEEE Trans Neural Syst Rehabil Eng* 13:89-98.

Sharbrough F, Chatrian G, Lesser R, Lüders H, Nuwer M, Picton T (1991) AEEGS guidelines for standard electrode position nomenclature. *Clin Neurophysiol* 8:202-204.

Shi J, Shen J, Ji Y, Du F (2012) A submatrix-based P300 brain-computer interface stimulus presentation paradigm. *J. Zhejiang Univ Sci C* 14:452-459.

Speier W, Arnold C, Lu J, Taira RK, Pouratian N (2012) Natural language processing with dynamic classification improves P300 speller accuracy and bit rate. *J Neural Eng* 9(1):016004.

Speier W, Fried I, Pouratian N (2013) Improved P300 speller performance using electrocorticography, spectral features and natural language processing. *Clin Neurophysiol* 1321-1328.

Speier W, Knall J, Pouratian N (2013) Unsupervised training of brain-computer interface systems using expectation maximization. *Int IEEE EMBS Conf Neural Eng* 707-710.

Speier W, Arnold C, Pouratian N (2013) Evaluating true BCI communication rate through mutual information and language models. *PLOS One* 8(10):e78342.

Speier W, Arnold C, Lu J, Deshpande A, Pouratian N (2014) Integrating language information with a hidden Markov model to improve communication rate in the P300 speller. *IEEE Trans Neural Syst Rehabil Eng*22(3):678-684.

Speier W, Deshpande A, Pouratian N (2014) A Method for Optimizing EEG Electrode Number and Configuration for Signal Acquisition in P300 Speller Systems. *Clin Neurophysiol* doi:10.1016/j.clinph.2014.09.021..

Speier W, Arnold C, Deshpande A, Knall J, Pouratian N (2015) Incorporating Advanced Language Models into the P300 Speller using Particle Filtering. Submitted for publication.

Spüler M, Rosenstiel W, Bogdan M (2012) Online Adaptation of a c-BEP Brain-Computer Interface (BCI) Based on Error-Related Potentials and Unsupervised Learning. *PLOS One* 7(12):e51077.

Squires N, Squires K, Hillyard S (1975) Two varieties of long-latency positive waves evoked by unpredictable auditory stimuli in man. *Electroencephalography and Clinical Neurophysiology* 38:387-401.

Sugg M, Polich J (1995) P300 from auditory stimuli: intensity and frequency effects. *Biol Psychol* 41:255-269.

Takano K, Komatsu T, Hata N, Nakajima Y, Kansaku K (2009) Visual stimuli for the P300 brain-computer interface: A comparison of white/gray and green/blue flicker matrices. *Clin Neurophysiol* 1120:1562-1566.

Townsend G, LaPallo B, Boulay C, Krusienski D, Frye G, Hauser C, Schwartz N, Vaughan T, Wolpaw J, Sellers E (2010) A novel P300-based brain-computer interface stimulus presentation paradigm: moving beyond rows and columns. *Cli. Neurophysiol* 121:1109-1120.

Ulas C, Cetin M (2013) Incorporation of a language model into a brain computer interface based speller through HMMs. *IEEE Conf on Acoustics, Speech and Signal Processing (ICASSP)* 1138-1142.

Ulas C, Cetin M (2013) The first Brain-Computer Interface utilizing a Turkish language model. *Signal Processing and Communications Applications Conference (SIU)*1-4.

Volosyak I, Moor A, Gräser A (2011) A Dictionary-Driven SSVEP Speller with a Modified Graphical User Interface. *Lecture Notes in Computer Science* 6691:353-361.

Wang P, King C, Do A, Nenadic Z (2012) Pushing the Communication Speed Limit of a Noninvasive BCI Speller. *Cornell University Library* arXiv:1212.0469 [cs.HC].

Wilson JA, Felton EA, Garell PC, Schalk G, Williams JC (2006) ECoG factors underlying multimodal control of a brain-computer interface. *IEEE Trans Neural Syst Rehabil EngI* 14:246-250.

Wolpaw J, McFarland KJ, Neat G, Forneris C (1991) EEG-based brain-computer interface for cursor control. *Electroencephalogr Clin Neurophysiol* 78:252-259.

Wolpaw J, Birbaumer N, Heetderks W, McFarland D, Peckham P, Schalk G, Donchin E, Quatrano L, Robinson C, Vaughan T (2000) Brian-Computer Interface Technology: A Review of the First International Meeting. *IEEE Trans Rehab Eng* 8(2):164-173.

Wolpaw J, Birbaumer N, McFarland D, Pfurtscheller G, Vaughan T (2002) Brain-computer interfaces for communication and control. *Clin Neurophysiol* 133:767-791.

Xu N, Gao X, Hong B, Miao X, Gao S, Yang F (2004) BCI competition 2003 – data set IIb: enhancing P300 wave detection using ICA-based subspace projections for BCI applications. *IEEE Trans Biomed Eng* 51:1067-1072.

Xu M, Qi H, Wan B, Yin T, Liu Z, and Ming D (2013) A hybrid BCI speller paradigm combining P300 potential and the SSVEP blocking feature. *J Neural Eng* 10:026001.

Yin E, Zhou Z, Jiang J, Chen F, Liu Y, Hu D (2013) A novel hybrid BCI speller based on incorporation of SSVEP into the P300 paradigm. *J Neural Eng* 10:026012.