A Flexible Brain-Computer Interface

By

Jessica D. Bayliss

Submitted in Partial Fulfillment

of the

Requirements for the Degree

Doctor of Philosophy

Supervised by

Professor Dana H. Ballard

Department of Computer Science

The College

Arts and Sciences

University of Rochester

Rochester, New York

2001

Curriculum Vitae

Jessica D. Bayliss was born in Hollywood, CA on December 29, 1970. As a Music Engineering major, Jessica attended the University of Miami from 1989 – 1991. This is where she became addicted to computers during late nights working at the university computer lab. After deciding to switch majors, Jessica attended California State University from 1991 to 1995 where she graduated *cum laude* with a Bachelor of Science degree in Computer Engineering. During a research semester at Lawrence Livermore National Laboratory, Jessica became intrigued by computer vision while reading the book *Computer Vision* by Dana Ballard and Chris Brown. She came to the University of Rochester in the fall of 1995 and began graduate studies in Computer Science. Jessica received a National Physical Science Consortium (NPSC) Fellowship from August 1995 until May 2001. This fellowship was sponsored by NASA Goddard Space Flight Center and Jessica explored topics in artificially intelligent agents as well as hyperspectral image analysis over two summers of interning at NASA Goddard. Working under the direction of Professor Dana H. Ballard, Jessica received a Master of Science in Computer Science in May of 1996. In late 1996, she became interested in brain-computer interfaces as a result of reading an article in *Scientific American*. Jessica has worked in collaboration with Dana H. Ballard on this topic and hopes someday that brain-computer interfaces will be a useful control alternative for handicapped users.

Acknowledgements

First and foremost I would like to thank my mother, Sandy. Without her encouragement, support, and wisdom, this document would never have been written. When I insisted that I didn't want to go to college, but would much prefer to be a hair dresser, it was mom who suggested that I might want to consider that hair dressers stood for long periods each day, and that my feet tended to hurt after a long walk. My sister, Kerensa, also deserves a special acknowledgement, as she has been my best friend through tough times. I remember when we braved a visit to the local neighborhood "haunted" house and became entranced by our first view of a pull chain toilet. Kerensa is the bravest person I know, because even when she is treated badly, she is able to get up and try again.

This work would not have been possible without my advisor and committee members: Dana Ballard, Chris Brown, Mary Hayhoe, and David Loiselle. I would like to thank my advisor Dana for the free reign he has given me. Most advisors would not have been as accepting of my need to forge ahead into untested waters. Not only was Dana accepting, but he also directed my research into the proper channels and gave me monetary support to build our current brain-computer interface. Chris gives wonderful, detailed comments about papers in short time frames. Mary has been especially helpful with the experimental paradigms used in this dissertation. I am not a psychophysicist and could never have hoped to do the right experiments without her help. The same goes for David, as he is the only clinical person on my committee and is not only responsible for teaching me how to apply electrodes, but for many interesting and informative discussions on EEG and EP signals. As a computer scientist, there are many mistakes I would have made without the support of Mary and David.

Research does not occur just within a committee. Bob Chapman has given good advice on experiments and research direction and I'd like to thank Rajesh Rao for the initial Matlab code on which the robust Kalman filter for this dissertation was built. Brent Auernheimer has patiently read through and made comments on a great deal of the dissertation and I especially would like to thank him for his discussions on the BCI design chapter. Jonas Karlsson at Xerox was also helpful. His dedication to new software engineering techniques such as XP helped me update my skill set when I worked at Xerox during a summer internship. And just when I thought the dissertation was finished, Paul Ilardi reminded me that my grammar was horrible and that I needed to explain some sections in greater detail. Thanks to Paul for his honest criticism and many interesting conversations over Friendly's ice cream.

On a personal level, I would like to thank the University of Rochester fencing club for giving me a place to take out my aggression, as well as many strange and novel fencing stories. For entertainment and unconditional love, Snoopy the Beaky bird, Zephyr the Bell bird, and Aquiel the Acrobat deserve my gratitude. There is something special about watching a sopping wet cockatoo named Snoopy wag his tail, an eight-year old budgie named Zephyr sleep with a bell covering his head, and a sprightly young budgie named Aquiel turn somersaults on the phone cord near the ceiling.

This research was supported by NIH/PHS grant1-P41-RR09283 and by a National Physical Science Consortium Fellowship with stipend support from NASA Goddard Space Flight Center.

Abstract

Recent advances in computer hardware and signal processing have made it feasible to use human EEG signals or "brain waves" to communicate with a computer. Locked-in patients now have a means to communicate with the outside world. Even with modern advances, such systems still suffer from communication rates on the order of 2-3 items/minute. In addition, existing systems are not likely to be designed with flexibility in mind, leading to slow systems that are difficult to improve. This dissertation presents a flexible brain-computer interface that is designed to facilitate changes in signal processing methods and user applications. In order to show the flexibility of the system, several applications, ranging from a brain-body actuated video game played with eye movements to a brain-computer interface for environmental control in a virtual apartment, are shown.

The P3 evoked potential is a positive wave in the EEG signal peaking at around 300 milliseconds after task-relevant stimuli and it can be used as a binary control signal. A virtual driving experiment shows that the P3 can be reliably detected within a virtual environment. Several on-line algorithms for processing single trial P3 evoked potentials are presented and compared. It is important that actual EEG signals rather than signal artifacts are being recognized and thus false recognition of artifacts is shown to be small.

Results from an environmental control application within a virtual apartment are presented. Subjects do not perform significantly different between controlling the application from a computer monitor and when fully immersed in the virtual apartment and subjects like the immersive VR environment better. This highlights the fact that the P3 component of the evoked potential is robust over different environments and that usability does not depend solely on performance, but on other factors as well. Future work is discussed within this context.

Table of Contents

List of Tables

List of Figures

1 Introduction

> *It is out of the question that the α-w and β-w of my E.E.G. exert any effect at a distance; they can not be transmitted through space. Upon the advice of experienced electrophysicists, I refrained from any attempt to observe possible distant effects.*
>
> \- *Berger [30]*

In 1929, Hans Berger published *On the Electroencephalogram of Man* [30]. His reports of regular quasi-sinusoidal oscillations were widely disbelieved by neurophysiologists of the day who could not believe that an organ as complex as the human brain could have such activity that could be recorded from the scalp. Berger himself doubted recordings of the 8-12 Hz activity that occurred when subjects had their eyes closed, but went away when their eyes opened. We now know this rhythm as the alpha rhythm.

Ahead of his time, Berger was convinced that mental processes were completely dependent upon human brain function. Due to an experience in 1893 after volunteering for service in the German army, Berger maintained a hope through much of his life and work that other, more fantastic processes might also be attributed to the human brain. In his last publication *Psyche*, Berger described this experience as follows:

> *As a 19 year old student, I had a serious accident during a military exercise near Würzburg and barely escaped certain death. Riding on the narrow edge of a steep ravine through which a road led, I fell with my rearing and tumbling horse down into the path of a mounted battery and came to lie almost beneath the wheel of one of the guns. The latter, pulled by six horses, came to a stop just in time and I escaped, having suffered no more than fright. This accident happened in the morning hours of a beautiful spring day. In the evening of the same day, I received a telegram from my father who enquired about my well being. It was the first and only time in my life that I received such a query. My oldest sister, to whom I had always been particularly close, had occasioned this telegraphic enquiry, because she had suddenly told my parents that she knew with certainty that I had suffered an accident. My family lived in Coburg at the time. This is a case of spontaneous telepathy in which at a time of mortal danger, and as I contemplated certain death, I transmitted my thoughts, while my sister, who was particularly close to me, acted as the receiver.*

Only in later years did Berger become convinced that the human electroencephalogram (EEG) could not exert any effect at a distance while still obeying the law of conservation of energy. We will never know what future path Berger would have taken. The Nazis had assumed full control of the university where he worked by 1938 and Berger was forced to retire. He lost his lab and did not have access to any of the equipment needed for experimentation. On May 30, 1941, in the depths of depression, Berger took his own life. But Berger was right about the EEG: his discoveries live on and have been expanded many times.

1.1 What is a Brain-Computer Interface?

A brain-computer interface uses electrophysiological signals to control remote devices. Most current BCIs are not invasive. They consist of electrodes applied to the scalp of an individual or worn in an electrode cap such as the one shown in 1-1 (Left). These electrodes pick up the brain's electrical activity (at the microvolt level) and carry it into amplifiers such as the ones shown in 1-1 (Right). These amplifiers amplify the signal approximately ten thousand times and then pass the signal via an analog to digital converter to a computer for processing. The computer processes the EEG signal and uses it in order to accomplish tasks such as communication and environmental control. BCIs are slow in comparison with normal human actions, because of the complexity and noisiness of the signals used, as well as the time necessary to complete recognition and signal processing.

The idea of using EEG signals for performing a simple task, such as selecting channels on a television set, is extremely difficult owing to the fact that signals are very small and embedded in noise. This difficulty often goes unappreciated, as brain-body actuated control devices are often marketed as true brain-computer interfaces. For example, the Interactive Brainwave Visual Analyzer (www.ibva.com) uses the following text on their web site:

> *IBVA is short for Interactive Brainwave Visual Analyzer. A system created and refined through 28 years of research. The IBVA provides easy real time analysis and intricate interactive biofeedback control of brainwave conditions for professionals as well as the curious. Put simply, the IBVA reads your brainwave activity in real time and allows you to use them to trigger images, sounds, other software or almost any electronically addressable device through its MIDI, serial and Expansion Pak features. With the network and modem features of the IBVA, your brainwaves can be analyzed and control equipment from anywhere in the world!*


Figure 1-1: (Left) A 32-channel electrode cap made by Neuromedical Supplies. (Right) A 32-channel set of analog Grass amplifiers.

This device sounds marvelous until one ascertains that all three of the electrodes used in their system are placed over the forehead. While EEG signals may be picked up at this location, eye movement signals are an order of magnitude larger and tend to overcome whatever EEG activity appears on the forehead. In addition, muscle activity from the forehead may also interfere with recording EEG activity at this location. Thus, this “brainwave analyzer” might be more accurately categorized as a frontalis muscle signal analyzer.

The phrase **brain-computer interface** (BCI) when taken literally means to interface an individual’s electrophysiological signals with a computer. A true BCI only uses signals from the brain and as such must treat eye and muscle movements as artifacts or noise. On the other hand, a system that uses eye, muscle, or other body potentials mixed with EEG signals, is a brain-body actuated system.

1.2 Why Detecting Ongoing Thoughts Isn't Practical

The first BCI, created around 1973 failed partially because it was programmed on a IBM 360/91[85]. This machine was a batch processing machine and could not be used for the real-time processing needs of a BCI. Recent advances in computers and signal processing have opened up a new generation of research on real-time EEG signal analysis and BCIs. Computers are now fast enough to handle the real-time constraints of BCI signal processing. Why isn't detecting ongoing thoughts and intentions from EEG signals practical?

A microelectrode placed next to a neuron can indeed record its cell firing patterns. However, there are many problems with using this idea for controlling a BCI. There would have to be a lot of electrodes to pick up an individual's thoughts, because each individual has billions of neurons. In addition, scientists do not understand the correlation between cell firing patterns and thought.

An electrode placed on the scalp cannot read the cell firing patterns of an individual neuron. In fact, the EEG is generated almost exclusively by the postsynaptic potentials of cortical neurons [78]. These potentials summate primarily at the vertically oriented large pyramidal cells in the cortex and extend to the scalp. Only a small fraction of the current penetrates through the meningeal coverings, spinal fluid, and skull to the scalp where it causes different parts of the scalp to be at different potential levels. These potential levels are on the order of microvolts and may be perturbed by extracerebral potential changes such as eye movements or other artifacts. Even if the EEG did not have signal-to-noise problems, it is far too complex to be deciphered in terms of underlying neural events. As an example, a potential change at the scalp could be caused by the same polarity produced near the surface of the cortex, but it may also be caused by a potential change of the opposite polarity occurring at cell bodies deeper in the cortex. Excitation in one place cannot be distinguished from inhibition in another place and thus individual thoughts cannot be divined.

Individual thoughts cannot be picked up and are probably not even correlated with the ongoing EEG activity. It is possible for an individual to be trained to produce a reliable signal or an individual may have a reliable response to a specific stimulus in a specific context. BCIs make use of such signals and if reactions to computer generated stimuli may be detected, then they can be used in order to control a light switch or a television set. If individuals may be trained to produce reliable signals that may be separated from ongoing EEG activity, then these signals may be used.

BCIs cannot detect ongoing thoughts and cannot even use EEG signals to communicate at half the speed of a person speaking. Why then should we develop BCIs? An important reason is for people with degenerative diseases.

1.3 Motivations

> *I am fading away. Slowly but surely. Like the sailor who watches the home shore gradually disappear, I watch my past recede. My old life still burns within me, but more and more of it is reduced to the ashes of memory.*
>
> *- Bauby [2]*

On December 8, 1995, Jean-Dominique Bauby, the editor-in-chief of French *Elle*, became a victim of locked-in syndrome due to a severe stroke. He was left with an unimpaired mind and the ability to blink his left eye. Rather than sink to the depths of despair in his self described *diving bell*, Bauby sought to release the *butterflies* of his imagination through writing a book about his new life. Since he only maintained the use of his left eye, how he could write a book? Without other knowledge, one might suppose that such an influential person would have all sorts of electronic gadgets to help. Instead, Bauby wrote the entire book by dictating it through blinking his eye. Each potential letter choice was placed in Bauby's field of view in the order of its frequency in the French language. Whenever Bauby wanted to choose a letter, he blinked. Not only did Bauby use this method to record his book, but he used this in order to communicate with others. It was predictable that Bauby would start to slip away from the world. He could not participate in real time conversation.

Worse yet, Bauby recounted the horrors of twenty-four hour care. He hated when the nurse left the television set on because he had no way of changing the channel or turning it off. The nurse did not always pay attention, even when she was around. Bauby talked about being ignored while madly blinking at the nurse to turn the TV off.

With available technology, why should individuals have to suffer silence because of an inability to communicate? Even in cases where an individual is completely locked-in: where no eye or muscle movements are controllable, individuals should have the ability to control their own environment and communicate with others. While there are other motivations for creating a brain-computer interface, the most important one is to give control back to even the most handicapped individuals. For instance, those in the later stages of amyotrophic lateral sclerosis should be helped.

1.3.1 Amyotrophic Lateral Sclerosis

According to the Doctor's Guide to ALS Information and Resources [24], Amyotrophic Lateral Sclerosis (ALS), commonly referred to as Lou Gehrig's disease, is a progressive fatal neuromuscular disease that attacks nerve cells and pathways in the brain to the spinal cord. Motor neurons reach from the brain to the spinal cord and from the spinal cord to the muscles throughout the body with connections to the brain as shown in 1-2. When the neurons die, the ability of the brain to initiate and control muscle movement dies with them. With voluntary muscle action affected, patients in the later stages are totally paralyzed or locked-in; yet, through it all, their minds remain unaffected. In fact, even during the later stages of the disease several individuals have written books about their experiences [32][75][52].

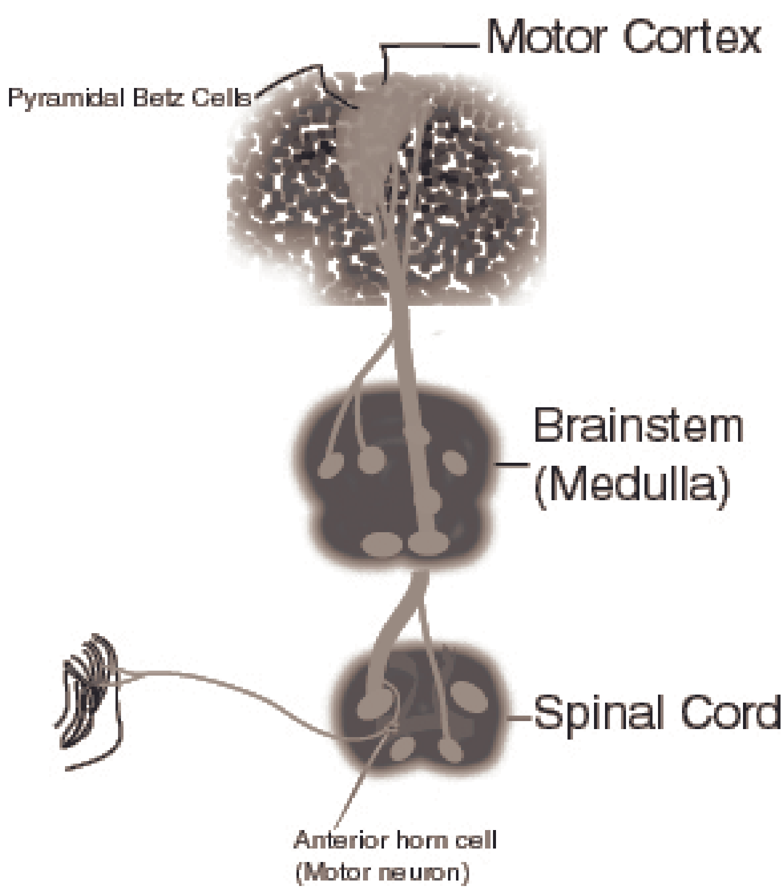


Figure 1-2: An illustration showing the degenerative path of ALS. In the earlier stages the Anterior horn cells in the spinal cord are affected. The disease may progress up from the spinal cord to the brain where it may affect the Pyramidal Betz cells in motor cortex.

ALS deserves special note as a motivation for the creation of BCIs. Around 5,000 people in the United States are newly diagnosed with ALS every year and the incidence rate appears to be rising. It is estimated that at any moment around 30,000 individuals have this disease and that of the U.S. population living today, in excess of 300,000 Americans will die from ALS. More people die every year of ALS than of Huntington's disease or Multiple Sclerosis [11]. Due to modern technology, half of all individuals with the disease live at least three years after diagnosis, about twenty percent live five years or more, and up to ten percent will survive more than ten years.

One of the primary questions to ask is whether or not a BCI system contains any benefits that put it above other methods of control already available. As discussed by Vaughan, Wolpaw, and Donchin [84], three options exist for restoring function:

1. augment the capabilities of remaining pathways
2. detour around the points of damage
3. provide the brain with wholly new channels for communication control

In the first option, muscles that remain under voluntary control substitute for paralyzed muscles. This is the option used for the most popular word processing systems, as eye movements generally remain under individual control in ALS. In the DECS system by LaCourse and Hludik, electro-oculographic techniques are used in order to choose an item from a computer screen [46].

The benefits of a system like this are that the percent of false triggers is low (1.4% in the DECS system) and that the necessary hardware/software is relatively inexpensive. The main problem with these systems is that they are awkward and limited to tasks where an individual's whole attention is always on the task at hand. Overloading eye movements as a form of control is problematic, because people use their eyes for many different tasks.

As an example, say that an individual should look *left* for driving a wheelchair left and *right* for driving right. *Up* perhaps would mean the individual wants to accelerate and *down* would mean stop. This system would work wonderfully until a nurse dropped a tray of food nearby. Loud noises act as an attentional draw and losing attention by looking around would cause incorrect control. While this example is somewhat exaggerated, it makes an important point: driving in the real world is dynamic and involves multiple tasks. The importance of visual information in perception means that using eye movements is inherently problematic as individuals like to look around in their environment.

Techniques that detour around points of damage restore function directly to affected muscles by detouring around breaks in the neural pathways that control them. In patients with spinal cord injuries electromyographic (EMG) activity from muscles above the level of the lesion may be used to control direct electrical stimulation of paralyzed muscles, thereby restoring useful movement.

This technique, known as functional electrical stimulation (FES) has been used to successfully restore hand grasping [46], as well as walking in people with thoratic cord lesions [45][53]. Of course, this method depends on the existence of some form of remaining muscle control in order to work and so would be of little use for people in the later stages of ALS.

EEG-based communication represents the creation of new channels for control and has the potential for overcoming the problems mentioned with the previous two approaches. It may be beneficial to use a BCI in dynamic tasks as the signal used can be specific to a task or group of tasks, making false signal classifications inherently less likely to occur than with eye movements. BCIs may also have the benefit of providing a useful control system throughout life cycle of ALS. While individuals in the early stages of ALS may not **need** a BCI for control since alternative muscle groups and eye movements may be used. Learning to us a BCI early in the disease provides the assurance that control will still be possible even if the individual loses eye or muscle movement control.

1.3.2 Military Uses

The Air Force is interested in using brain-body actuated control to make faster responses possible for fighter pilots. While brain-body actuated control is not a true BCI, it may still provide motivations for why a BCI could prove useful in the future. In the system discussed by Nelson [56], a combination of EEG signals and artifacts (eye movement, body movement, etc.) combine to create a signal that can be used to fly a virtual plane. The article does not state whether or not this speeds up a pilot's responses or whether such a system might prove useful in locating relevant targets. In general, one can imagine that the military would have multiple uses for a system that speeds up response times in areas such as tactical maneuvering and perhaps even in targeting and firing weapons.

Currently, the main focus of Air Force research is for Alternative Control Technology (ACT). The goal of the ACT program is to enable communication with computers while the computer user’s hands are busy with other tasks. As an example, alternative controls may be used to enable maintenance technicians to manually operate test equipment while accessing schematics on a head-mounted display [54]. While the Air Force has a special interest in hands-off control, this kind of control may be of interest in non-military areas. One can imagine surgeons switching augmented displays during surgery without having to talk or use their hands.

1.3.3 Other Possible Uses

While the University of Rochester BCI is not targeted for use by the general public, one can imagine several applications for everyday people. The actual use of the proposed system for the general public would be dependent on technological advancement in the equipment used for recording the EEG signal (currently several thousand dollars) as well as in techniques for applying electrodes easily. Still, such things are possible in the future and with such revolutions entire new realms of possibility would open up.

For instance, the area of entertainment would be greatly enhanced by the ability to control video games and to have video games react to actual EEG signals. Furthermore, one can imagine the EEG signal as an entirely new interface to the computer. It does not require clicking or other body movement, but also does not require the user to speak (potentially embarrassing when correcting a mistake).

Current brain-body actuated systems have software that is used for composing music, playing video games, or even to relax. An example of this sort of system may be found at www.brainfingers.com and is called the Cyberlink system. This system uses biofeedback techniques in order to relax muscles, but may also use combinations of eye movements and alpha waves for computer control. A clerical worker reported a novel reason for using the Cyberlink system [21]: after suffering from Carpel Tunnel Syndrome, the individual obtained a speech controlled system for typing. Since clerical work may involve a lot of typing, the user would become hoarse after a couple of hours using this system. The solution to this problem was to use the Cyberlink for computer communication.

While BCIs have not gained the commercial acceptance of brain-body actuated systems, there are good motivations for considering a BCI. Only a true BCI can help the most handicapped individuals with ALS on a day-to-day basis. Once a reliable BCI becomes commercially available at a reasonable cost, it will only be a matter of time before multi-modal systems based on separate brain and body actuation may become a reality.

1.4 Signals for BCI Control

What signals should be used for control in a BCI? This is an open question in the field and quite a few signals are in current use. Signals may be broken into three general categories: implanted methods, evoked potentials, and operant conditioning. Implanted methods offer control at the price of invasiveness. Both evoked potential and operant conditioning methods are normally externally-based BCIs, as the electrodes are located on the scalp. Chapter 2 describes the different signals in common use.

Evoked potentials (EPs) are brain potentials that are evoked by the occurrence of a sensory stimulus. They are usually obtained by averaging a number of brief EEG segments time-registered to a stimulus in a simple task. In a BCI, EPs may provide control when the BCI application produces the appropriate stimuli. This paradigm has the benefit of requiring little to no training to use the BCI at the cost of having to make users wait for the relevant stimulus presentation. EPs offer discrete control for almost all users, as EPs are an inherent response.

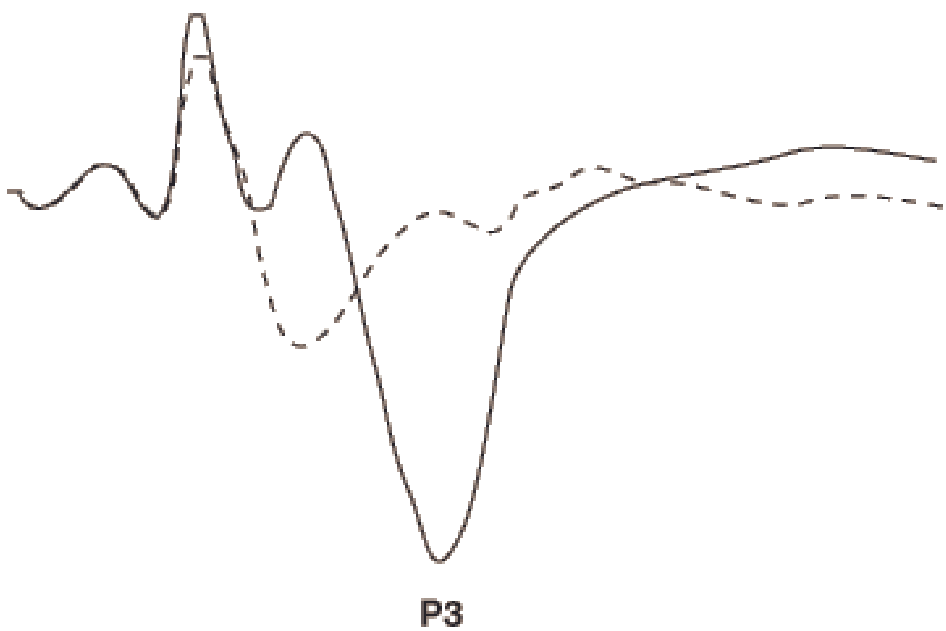


Figure 1-3: (Solid line) The general form of the P3 component of the evoked potential (EP). The P3 is a cognitive EP that appears approximately 300 ms after a task relevant stimulus. (Dotted line) The general form of a non-task related response.

Around 1964, Chapman and Bragdon [17] as well as Sutton et. al. [81] independently discovered a positive wave peaking at around 300 ms after task-relevant stimuli. This component is known as the P3 and its general form is shown in Figure 1-3. While the P3 is evoked by many types of paradigms, the most common factors that influence it are the frequency of stimulus occurrence (less frequent stimuli produce a larger response) and task relevance. The P3 has been shown to be fairly stable in locked-in patients, re-appearing even after brain stem injuries [60]. This makes it possible as a control choice for locked-in individuals. Farwell and Donchin first showed that this signal may be successfully used in a BCI [25]. The P3 is a non-specific response, meaning that it occurs in response to a wide variety of stimuli types. Using a broad cognitive signal like the P3 has the benefit of enabling control through a variety of modalities, because the P3 may enable discrete control in response to both auditory and visual stimuli.

All brain-computer interface experiments in this dissertation use the P3 component of the visual evoked potential. This signal is suitable for infrequent, discrete tasks such as environmental control, but it is recognized that other signals may also work. In the future, the flexibility of the University of Rochester BCI will be used to compare and contrast the utility of different signals for the same task as well as combinations of signals.

1.5 Contributions

The goal of this dissertation is to look at current BCIs and the improvements that can and should be made. BCIs should be flexible, useful, and usable. Flexibility is necessary so that BCIs may be augmented and maintained as new technology becomes available. Previous BCIs have concentrated on specific hardware solutions for specific programs. This practice leads to systems that quickly become outdated due to their inflexibility. A BCI must also be useful for a wide variety of tasks, as is necessary when a BCI is used as the main control device for a handicapped individual. Last, but most importantly, a BCI needs to be usable. Usability is often tied to signal recognition in that faster systems are more usable. While this is true, there are many other concerns, such as the display for the user application.

In order to further these goals, the following contributions have been made by the present study:

- A flexible BCI system has been designed and constructed. This system differs from systems currently available in that it is designed to make signal processing and user applications easy to change or update. Applications ranging from a P3 evoked potential BCI to a brain-body actuated game operated by eye movements are presented. (Chapter 3)
- Recognizing that virtual reality may prove useful for training individuals to use a BCI, for providing complex and controllable experimental environments for those improving BCIs, and for motivational reasons, it is shown that evoked potentials may be reliably obtained in a virtual environment. (Chapter 5)
- The P3 evoked potential has been shown to be robust under a variety of environments and conditions ranging from a virtual driving simulator (Chapter 5) to watching an apartment on a computer monitor (Chapter 6).
- Results from an electrophysiologically-based BCI that uses the P3 event related potential in a virtual apartment show that subjects may find an immersive VR apartment to be more usable than watching that same apartment on a computer monitor, even when the recognition accuracy is better on the computer monitor.

Usability may be determined by factors other than signal recognition. (Chapter 6)

- Signal processing results from routines involving different kinds of preprocessing and recognition algorithms are presented from the off-line analysis of data collected in a P3-based BCI. It is shown that increased recognition does not necessarily mean a better system, as there are most often trade-offs between recognition accuracy and speed. (Chapter 4)

2 Background

> *Can these observable electrical brain signals be put to work as carriers of information in man-computer communication or for the purpose of controlling such external apparatus as prosthetic devices or spaceships? Even on the sole basis of the present states of the art of computer science and neurophysiology, one may suggest that such a feat is potentially around the corner.*
>
> \- *Vidal [85]*

In 1973, Jacques Vidal published an article on the first BCI. In the 23-page paper, most of the space was devoted to describing EEG signal acquisition hardware/software and the signal processing of the obtained EEG signals. Real-time acquisition is imperative for a BCI system and the existing computer equipment was not up to the task. Still, many of the concepts used today in BCIs were discussed in Vidal's paper. After describing the future possibilities for BCIs, Vidal talked about neurophysical considerations. What brain signals should be used for a BCI and what were the properties of these signals? Vidal mentioned alpha rhythms, evoked potentials, and even event-related synchronization/desynchronization (ERS/ERD) of the EEG, all of which are used in BCIs today. The idea for advanced processing of single trial evoked potentials using principal component analysis appeared in Vidal's paper as well as the more common spectral analysis of EEG signals. The goal of the paper was to indicate the necessary components for a working BCI and this was done very well. Even with its forward thinking, Vidal could not have foreseen some of the more modern issues associated with getting a BCI to work *well*. These BCI *system* issues include designing the user application while taking human factors into consideration as well as the overall BCI system architecture.

2.1 The Nature of the EEG and Some Unanswered Questions

Much is known and much remains a mystery about the nature of EEG signals. Knowledge about EEG signals may help the BCI researcher in two ways. First, knowledge may help the researcher choose what signal conveys the most information for control and second, it may aid in developing signal processing algorithms for detecting the relevant signal. Lack of knowledge hinders the BCI researcher. When the true nature of the signal is unknown, it is difficult to choose the most appropriate signal processing routine for recognition.

Traditionally, electroencephalogram (EEG) is a display of brain voltage potentials written onto paper over time. A modern system for EEG acquisition digitizes these potentials for computer storage, although systems that output directly onto paper remain in use. Electrodes passively conduct voltage potentials from *columns* of neurons in the brain and must pick up microvolt level signals. The signal to noise ratio must be kept as high as possible and electrodes are constructed from such materials as gold and silver chloride in order to aid in this. Various conductive gels or pastes are used between an individual's skin and the electrode in order to reduce the impedance between the electrode and the scalp as much as possible.

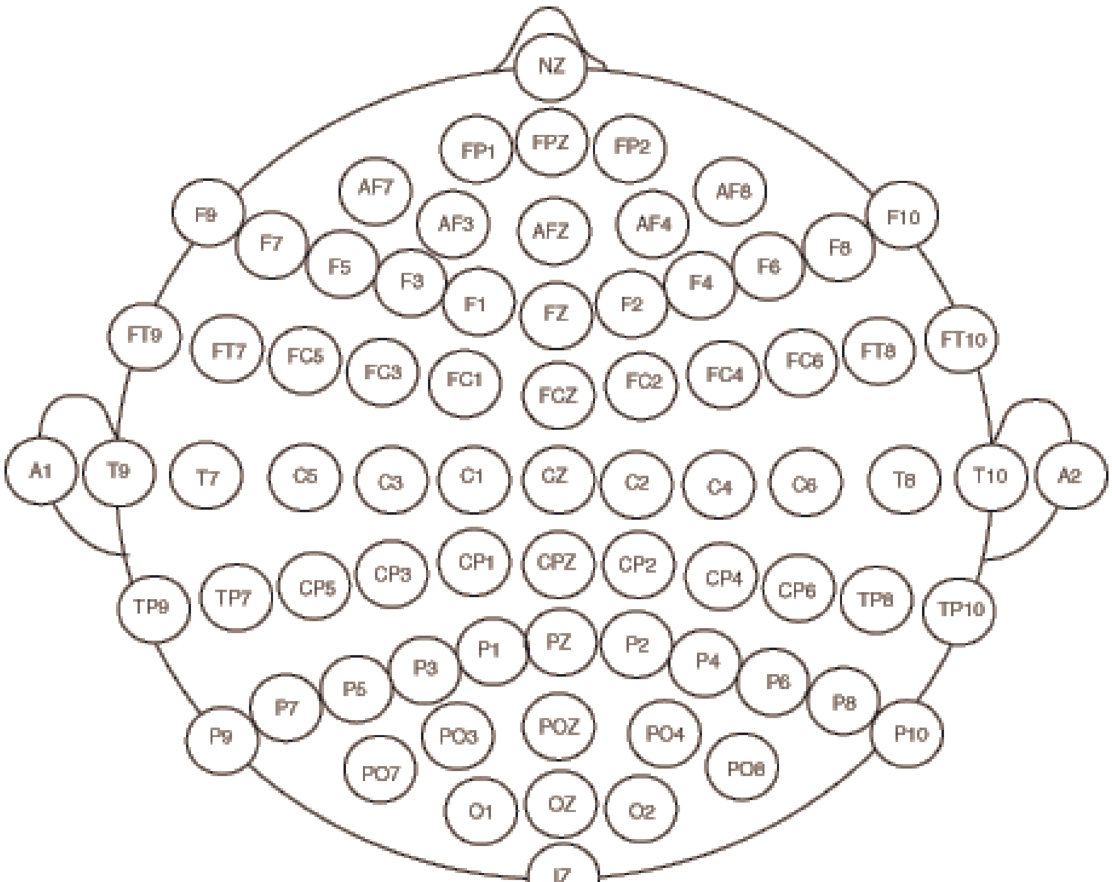


Figure 2-1: The extended 10-20 system for electrode placement. Even numbers indicate electrodes located on the right side of the head while odd numbers indicate electrodes on the left side. The letter before the number indicates the general area of the cortex the electrode is located above. A stands for auricular, C for central, Fp for prefrontal, F for frontal, P for parietal, O for Occipital, and T for temporal. In addition, electrodes for recording vertical and horizontal electro-oculographic (EOG) movements are also place. Vertical EOG electrodes are placed above and below an eye and horizontal EOG electrodes are placed on the side of both eyes away from the nose.

Configurations of electrodes usually follow the International 10-20 system of placement [35], although larger electrode arrays may follow the Modified Expanded 10-20 system as proposed by the American EEG Society (see Figure 2-1). The introduction of the Modified Expanded 10-20 system indicates an increase in the normal application of an expanded number of electrodes. Not surprisingly, more electrodes means increased spatial resolution of the signal over the head and arrays with as many as 256 electrodes have been used successfully in research applications.

The availability of large numbers of electrodes introduces the problem of how to connect them to the recording device. A plethora of different configurations exist, but two main classes of configurations or *montages* arise from the possibilities: referential and bipolar montages.

The distinguishing feature of referential montages is that all electrode potentials are calculated with respect to a reference electrode placed in an electrically quiet area. The main advantage of such a recording method is that referential recording can give an undistorted display of the shape of potential changes and is especially useful for the recording of potentials with a wide distribution. Since differential amplifiers are used, referential montages also make it simple to mathematically calculate other kinds of montages after recording.

Unfortunately, it is essentially impossible to find a reference electrode that is entirely inactive. Reference electrodes located everywhere from the ear to the big toe have failed in the attempt to find a truly quiet reference. In order to help overcome this problem, *average* reference electrodes (where two electrode sites contribute equally to the reference electrode) may be used. The most common average reference electrode configuration is known as the *linked ears* configuration due to the equal contribution of A1 and A2 to the reference electrode. A1 and A2 may also be attached to the mastoids instead of the ears, in which case the reference is known as a *linked mastoid* configuration. In order to remove the influence of the reference location from the recording, techniques such as the Hjorth transform [33] may be used.

Bipolar montages connect pairs of electrodes to the inputs of amplifiers. As an example, the longitudinal bipolar montage connects Fp1-F3, F3-C3, C3-P3, P3-O1, and so on, forming rows of electrodes. The advantage of these types of montages is that they distinguish local activity much more clearly than a referential montage. The disadvantage of bipolar montages is that they may distort the wave shape and amplitude of widely distributed potentials.

Clearly, the type of montage used will greatly effect the ability of a system to recognize certain events in the signal. Since BCIs tend to deal with widely distributed signals, most BCIs use a referential montage. After a montage is chosen, the electrode voltage potentials are differentially amplified on the order of ten to twenty thousand times the original voltage.

As discussed in *Spehlmann's EEG Primer* [78], the EEG reader needs to distinguish the following features: waveform, repetition, frequency, amplitude, distribution, phase relation, timing, persistence, and reactivity. These are common features distinguished by BCIs. Waveforms may be regular, having a fairly uniform appearance due to symmetrical rising and falling phases. One example of a regular waveform would be a sinusoidal wave. Other waveforms may be irregular, having uneven shapes and durations. The waveform frequencies of particular interest to clinical EEG readers range from 0.1 Hz to around 20 Hz. Many frequencies are apparent in the normal EEG and frequency bands help to set apart the most normal and abnormal waves in the EEG, making frequency an important criteria for assessing abnormality in clinical EEG. As electrodes are positioned over different parts of the head, the electrical activity recorded may appear over large or small areas. This is the *distribution* of a wave. Distributions may be lateralized on one side of the head or may be diffuse. *Focal activity* is activity that is restricted to one or a few electrodes over an area of the head. The *reactivity* of a signal refers to changes that may be produced in some normal and abnormal patterns by various maneuvers. A common example of this is the blocking of the alpha rhythm by eye opening or other alerting procedures [78].

While some descriptors of the EEG signal seem fairly obvious, there are others that have created controversy in the EEG community. One of the obvious questions on the nature of the EEG signal remains unknown - is the system linear or nonlinear? It is also unknown how chaotic the data is. Without the answers to these questions, it remains difficult to choose the proper routines for EEG signal recognition. Toda, Murai, and Usui present a measure of nonlinearity in time series [83]. The measure of nonlinearity is calculated from the weights of a trained feedforward neural network with nonlinear hidden units. As examples, they measure the nonlinearity of sunspot series and a carp's EEG. The sunspot is (of course) found to be nonlinear, but the carp's EEG is linear. While there are problems with this approach, such as the lack of complete data sets and noise effects, the approach raises the question of the possibility of globally linear neurocortical dynamics. Freeman's nonlinear model for the neocortex assumes chaotic nonlinear dynamics [28] [29]. Pyramidal cells are important neurons in the neocortex and Freeman’s model predicts that the sharp nonlinearity of the neuronal threshold could cause chaotic dynamics if both the firing rate and the field potential of any pyramidal cell were raised above a critical level of excitation. Simulations of his principles have yielded the predicted chaotic dynamic properties.

There is no incontrovertible proof that the EEG reflects any simple chaotic process [89]. Fundamental difficulties lie in the applicability of estimation algorithms to EEG data, because of limitations in the size of data sets, noise contamination, and lack of signal stationarity. Even with locally chaotic dynamics, does this mean that there must be globally chaotic dynamics? An important class of simulation studies suggest this must be the case [40] [41]. These studies concern one-dimensional chaotic numerical subprocesses of considerable generality (one-dimensional chaotic maps) that are globally coupled, each to all others. Such coupled maps exhibit global chaos and appear to escape from the law of large numbers and the central limit theorem. However, the escape from the law of large numbers does not occur in the presence of noise (a common element in any EEG) [40] [41].

The nonlinear model proposed by Freeman contrasts with one proposed by Nunez [58]. Nunez's model treats the EEG signal as a linear wave process and the global dynamics of the brain are treated as a problem of the mass action of coupled neuron-like elements [89]. While Freeman's model predicts an oscillation caused by neuronal firing at around 40 Hz that is consistent with experimental findings, Nunez's model predicts a wave propagation velocity of 7-11 m/sec for human alpha waves that is also consistent with experimental findings. Either model appears consistent with some experimental data, but is either model correct? Interestingly enough, due to the noise in an EEG signal, both models could be correct. Freeman's model might actually agree with Nunez's globally linear model for neocortical EEG.

Since the nature of EEG signals is unknown, difficulties lie in trying to decide on a particular signal recognition routine. At best, if EEG signals are linear, then the linear recognition algorithms that most BCIs use may be sufficient. At worst, linear recognition algorithms are poor descriptors of the signals they hope to recognize.

2.2 Neurophysiological Signals Used in BCIs

What signals should be used for control in a BCI? This is an open question in the field and quite a few signals are in current use. As previously stated, signals may be broken into three general categories: implanted methods, evoked potentials, and operant conditioning. Both evoked potential and operant conditioning methods are normally externally-based BCIs as the electrodes are located on the scalp. Table 2-1 describes the different signals in common use. It may be noted that some of the described signals fit into multiple categories. As an example, single neural recordings may use operant conditioning in order to train neurons for control or may accept the natural occurring signals for control. Where this occurs, the signal is described under the category that most distinguishes it.

Several questions are of relevance when considering what signal to use for a proposed BCI:

1. **What remaining control is necessary in order to use the BCI?**
   Some BCIs require the use of eye movement control and some do not require any remaining motor control.

2. **Does the user of the BCI need to be trained in order to elicit the necessary signal for control and if so, then how long does the training last?**
   Operant conditioning methods may require extensive training in order to use them for control.

3. **What percentage of the population can obtain control using the signal?**
   While almost everybody has apparent evoked potentials, not everybody appears to be able to use biofeedback in order to learn how to use a BCI based on operant conditioning. This is discussed further below.

4. **Does the signal provide continuous or discrete control?**
   Evoked potentials may only provide discrete control. Operant conditioned signals may provide continuous control, because they are obtained from ongoing EEG activity.

5. **Does the nature of the signal change over time?**
   Many of the signals currently used may change as a function of fatigue.

6. **Does the signal necessitate an invasive procedure in order to work?**
   While most BCIs obtain control using electrodes on the scalp, implanted methods are invasive.

Implanted methods use signals from single or small groups of neurons in order to control a BCI. These methods have the benefit of a much higher signal-to-noise ratio at the cost of being invasive. They require no remaining motor control and may provide either discrete or continuous control. Chapin and Gaal have successfully recorded up to 46 neurons and used their natural responses to enable four out of eight rats to obtain water with the neural processes [15][16]. While most systems are still in the experimental stage, Kennedy’s group has forged ahead to provide control for locked-in patient JR [42] [43]. Kennedy’s approach involves encouraging the growth of neural tissue into the hollow tip of a two-wire electrode known as a neurotrophic electrode. The tip contains growth factors that spur brain tissue to grow through it. Through an amplifier and antennas positioned between the skull and the scalp, the neural signals are transmitted to a computer, which can then use the signals to drive a mouse cursor. This technique has provided stable long term recording and patient JR has learned to produce synthetic speech with the BCI over a period of more than 426 days. It is unknown how well this technique would work on multiple individuals, but it has worked on both patients (JR and MH) who have been implanted.

Evoked potentials (EPs) are usually obtained by averaging a number of brief EEG segments time-registered to a stimulus in a simple cognitive task. In a BCI, EPs may provide control when the BCI application produces the appropriate stimuli. This paradigm has the benefit of requiring little to no training to use the BCI at the cost of having to make users wait for the relevant stimulus presentation. EPs offer discrete control for almost all users as EPs are an inherent response.

Exogenous components, or those components influenced primarily by physical stimulus properties, generally take place within the first 200 milliseconds after stimulus onset. These components include a Negative waveform around 100 ms (N1) and a Positive waveform around 200 ms after stimulus onset (P2). Visual evoked potentials (VEPs) fall into this category. Sutter uses short visual stimuli in order to determine what command an individual is looking at and therefore wants to pick [79]. He also shows that implanting electrodes improves performance in an externally-based BCI.

In a different approach, McMillan and colleagues have trained volunteers to control the amplitude of their steady-state VEPs to florescent tubes flashing at 13.25 Hz [36][54][84]. Using VEPs has the benefit of a quicker response than longer latency components. The VEP requires that the subject have good visual control in order to look at the appropriate stimulus and allows for discrete control. As the VEP is an exogenous component, it should be relatively stable over time.

Endogenous components, or those components influenced by cognitive factors, take place following the exogenous components. Around 1964, Chapman and Bragdon [17] as well as Sutton et. el. [81] independently discovered a positive wave peaking at around 300 ms after task-relevant stimuli. This component is known as the P3 and is shown in Figure 1-3. While the P3 is evoked by many types of paradigms, the most common factors that influence it are stimulus frequency (less frequent stimuli produce a larger response) and task relevance. The P3 has been shown to be fairly stable in locked-in patients, re-appearing even after severe brain stem injuries [60]. Farwell and Donchin first showed that this signal may be successfully used in a BCI [25]. Using a broad cognitive signal like the P3 has the benefit of enabling control through a variety of modalities, as the P3 enables discrete control in response to both auditory and visual stimuli. As it is a cognitive component, the P3 has been known to change in response to subject fatigue. In one study, a reduction in the P3 was attributed to fatigue after subjects performed the task for several hours [77].

As shown in Table 2-1, several methods use operant conditioning on spontaneous EEG signals for BCI control. The main feature of this kind of operant

Table 2-1: Common signals used in BCIs

Signal Name	Description
Mu, and Alpha Rhythm Operant Conditioning	The mu rhythm is an 8-12 Hz spontaneous EEG rhythm associated with motor activities and maximally recorded over sensorimotor cortex. The alpha rhythm is in the same frequency band, but is recorded over occipital cortex. The amplitudes of these rhythms may be altered through biofeedback training.
Event-related Synchronization/Desynchronization (ERS/ERD) Operant Conditioning	Movement-related increases and decreases in specific frequency bands maximally located over sensorimotor cortex. Individuals may be trained through biofeedback to alter the amplitude of signals in the appropriate frequency bands. These signals exist even when the individual imagines moving as the movement-related signals are preparatory rather than actual.
Slow Cortical Potential Operant Conditioning	Large negative or positive shifts in the EEG signal lasting from 300ms up to several minutes. Individuals may be trained through biofeedback to produce these shifts.
P3 Component of the Evoked Potential	A positive shift in the EEG signal approximately 300-400ms after a task relevant stimulus. Maximally located over the central parietal region, this is an inherent response and no training is necessary.
Short-Latency Visual Evoked Potentials	To produce the component, a response to the presentation of a short visual stimulus is necessary. Maximally located over the occipital region, this is an inherent response and no training is necessary.
Individual Neuron Recordings	Individuals receive implanted electrodes that may obtain responses from local neurons or even encourage neural tissue to grow into the implant. Operant conditioning may be used to achieve control or the natural response of a cell or cells may be used.
Steady-State Visual Evoked Potential (SSVER)	A response to a visual stimulus modulated at a specific frequency. The SSVER is characterized by an increase in EEG activity at the stimulus frequency. Typically, the visual stimulus is generated using white fluorescent tubes modulated at around 13.25 Hz or by another kind of strobe light. A system may be constructed by conditioning individuals to modulate the amplitude of their response or by using multiple SSVERs for different system decisions.

conditioning is that it enables continuous rather than discrete control. This feature may also serve as a drawback: continuous control is fatiguing for patients and fatigue may cause changes in performance since control is learned. As shown by the various groups using these methods, operant conditioning methods using spontaneous EEG are not easily learned by everybody.

Wolpaw and his colleagues train individuals to control their mu rhythm amplitude (discussed in Table 2-1) for cursor control [88]. Mu rhythm control does not require subjects to have any remaining motor control. For the cursor control task, normal subjects are trained on the order of 10-15 sessions in order learn to move the cursor up/down. In the several papers examined, it appears that not all subjects obtain control, although most seem to during this time frame. It is normal to see four out of five subjects who obtain greater than 90% accuracy with the other one obtaining around chance [88]. This implies that somewhere around 80% of the subjects may obtain good control.

In related work, the Graz brain-computer interface trains people to control the amplitude of their ERS/ERD patterns. Subjects are trained over a few sessions in order to learn a cursor control task. As in the mu rhythm control, not all subjects learn to control the cursor accurately. Obtaining two out of six subjects who are not able to perform the cursor control task has been reported [67]. Part of the charm of this system is that it gives biofeedback to the user in the form of a moving cursor after training. The use of areas over the sensorimotor cortex for both ERS/ERD and mu rhythm control might pose a problem in people with ALS because the cortical Betz cells in the motor cortex may die in the later stages of the disease [11].

Slow cortical potentials serve as the signal in the Thought Translation Device, a communication device for ALS patients created by Biurbaumer’s group in Austria [12]. Since this system is used with patients, it is difficult to tell how hard it is to learn the system. Patients may be medicated, depressed, or fatigued: all of which affect learning rates. Subjects are trained over several months to use the system. All subjects that have wanted to learn the system seem to have been successful. No remaining motor control is necessary in order to use the Thought Translation Device. Unlike mu rhythm control or ERS/ERD, the slow cortical potential has not been used for continuous control. It may take many seconds in order to produce and hold a slow cortical potential in order to trigger the system.

While the signals discussed are used currently, other signals may be possible. Several papers have been written on recognizing EEG signal differences during different mental calculations. These papers suggest that different parts of the brain are active during different types of mental calculation, and if these different tasks may be accurately recognized, they could be used in a BCI. Lin et. al. [48] describe a study where five tasks were compared: multiplication problem solving, geometric figure rotation, mental letter composing, visual counting, and a baseline task where the subject was instructed to think about nothing in particular. Results from this experiment suggest that the easiest tasks to identify are multiplication problem solving and geometric figure rotation, but even these tasks are not easily identified. Other papers have concentrated on mental tasks, but none have found easily recognizable differences between different tasks [23][26].

2.3 Existing Systems

Current systems range from simple experimental interfaces meant to test the suitability of a specific EEG signal to full applications used by patients. The system includes the hardware used in the BCI, the underlying BCI backend software, and the user application. While the hardware used in a research testbed does not matter as long as it performs as needed, expense, portability, and reliability become very real issues in a BCI for patient use.

The underlying BCI backend software is not discussed in many papers. It is, however, as important as the hardware. The backend includes software for reading in the EEG signals, scheduling them for processing, and processing them into a form that may be used by the user application. The backend software determines the BCI portability, extendibility, and flexibility. It also determines how maintainable the software will be over a period of time. For instance, the construction of the software may provide the flexibility to enable users to choose from a wide variety of user applications or the user may only be able to use one application if the BCI system is monolithic.

In assessing current user applications, it is important to consider the usability of the application. The field of human factors tells us repeatedly that a poorly designed user application may injure performance. This applies to a BCI as well as to many other items in everyday use and will occur regardless of the signal recognition routines used. Several important factors should be considered in the design of the application, including five mentioned by Ben Shneiderman [74]:

1. What is the time to learn the system?
2. What is the speed of performance?
3. How many and what kinds of errors do users make?
4. How well do users maintain their knowledge after an hour, a day, or a week? What is their retention?
5. How much did users like using various aspects of the system? What is their subjective satisfaction?

Table 2-2: A comparison of several features in existing BCIs

System	***Training Time***	***Number of Choices***	***Speed*** +	***Errors***	***Retention***	***Subjective Satisfaction***
Brain Response Interface	*10-60 minutes*	*64*	*30*	*10%*	*Excellent*	*Considered*
SSVEP Training	*6 hrs.*	*N/A*	*Not Available*	*20% or less*	*Not mentioned*	*Not Discussed*
P3 Character Recognition	*Minutes*	*36*	*4*	*5%*	*Excellent*	*Not Discussed*
Mu Rhythm Training	*15-20 sessions*	*2*	*20*	*10%*	*Not mentioned*	*Not Discussed*
ERS/ERD	*2-2.5 hrs.*	*2*	*Not Available*	*11% or less*	*Not mentioned*	*Not Discussed*
Thought Translation Device	*Months**	*27*	*2*	*10-30%*	*Not Good*	*Indirectly discussed*
Implanted Device	*Months**	*N/A*	*2*	*Not reported*	*Excellent*	*Considered*

**This time period is heavily influenced by the fact that patients are being trained rather than healthy individuals.*

+ *average number of items or movements per minute.*

Several features of existing BCIs are compared in Table 2-2. Surprisingly, most BCI papers do not discuss subjective satisfaction at all and so the category for subjective satisfaction only includes whether or not it was considered in the papers about the system. In addition to these considerations, the application designer might want to consider the following general goals as specified by the U.S. Military Standard for Human Engineering Design Criteria [74]:

1. Achieve required performance by operator, control, and maintenance personnel
2. Minimize skill and personnel requirements and training time
3. Achieve required reliability of personnel—equipment combinations
4. Foster design standardization within and among systems

When measured using these considerations, all BCIs fall short in some manner. This could be because most BCIs are research instruments or grow out of a research project. In the future, it will be very important to consider the system-wide aspects of BCIs.

2.3.1 The Brain Response Interface

Sutter's Brain Response Interface (BRI) [79] is a system that takes advantage of the fact that large chunks of the visual system are devoted to processing information from the foveal region. The BRI uses visually evoked potentials (VEP's) produced in response to brief visual stimuli. These EP's are then used to give a discrete command to pick a certain part of a computer screen. This system is one of the few that have been tested on severely handicapped individuals. Word processing output approaches 10-12 words/min. and accuracy approaches 90% with the use of epidural electrodes. This is the only system mentioned that uses implanted electrodes to obtain a larger, less contaminated signal.

A BRI user watches a computer screen with a grid of 64 symbols (some of which lead to other pages of symbols) and concentrates on the chosen symbol. A specific subgroup of these symbols undergoes a equiluminant red/green fine check or plain color pattern alteration in a simultaneous stimulator scheme at the monitor vertical refresh rate (40-70 frames/s). Sutter considered the usability of the system over time and since color alteration between red and green was almost as effective as having the monitor flicker, he chose to use the color alteration because it was shown to be much less fatiguing for users.

The EEG response to this stimulus is digitized and stored. Each symbol is included in several different subgroups and the subgroups are presented several times. The average EEG response for each subgroup is computed and compared to a previously saved VEP template (obtained in an initial training session), yielding a high accuracy system.

This system is basically the EEG version of an eye movement recognition system and contains similar problems because it assumes that the subject is always looking at a command on the computer screen. On the positive side, this system has one of the best recognition rates of current systems and may be used by individuals with sufficient eye control. Performance is much faster than most BCIs, but is very slow when compared to the speed of a good typist (80 words/min.).

The system architecture is advanced. The BRI is implemented on a separate processor with a Motorola 68000 CPU. A schematic of the system is shown in Figure 2-2. The BRI processor interacts with a special display showing the BRI grid of symbols as well as a speech synthesizer and special keyboard interface. The special keyboard interface enables the subject to control any regular PC programs that may be controlled from the keyboard. In addition, a remote control is interfaced with the BRI in order to enable the subject to control a TV or VCR. Since the BRI processor loads up all necessary software from the hard drive of a connected PC, the user may create or change command sequences. The main drawback of the system architecture is that it is based on a special hardware interface. This may be problematic when changes need to be made to the system over time.

2.3.2 P3 Character Recognition

In a related approach, Farwell and Donchin use the P3 evoked potential [25]. A 6x6 grid containing letters from the alphabet is displayed on the computer monitor and users are asked to select the letters in a word by counting the number of times that a row or column containing the letter flashes. Flashes occur at about 10 Hz and the desired letter flashes twice in every set of twelve flashes. The average response to each row and column is computed and the P3 amplitude is measured. Response amplitude is reliably larger for the row and column containing the desired letter. After two training sessions, users are able to communicate at a rate of 2.3 characters/min, with accuracy rates of 95%. This system is currently only used in a research setting.

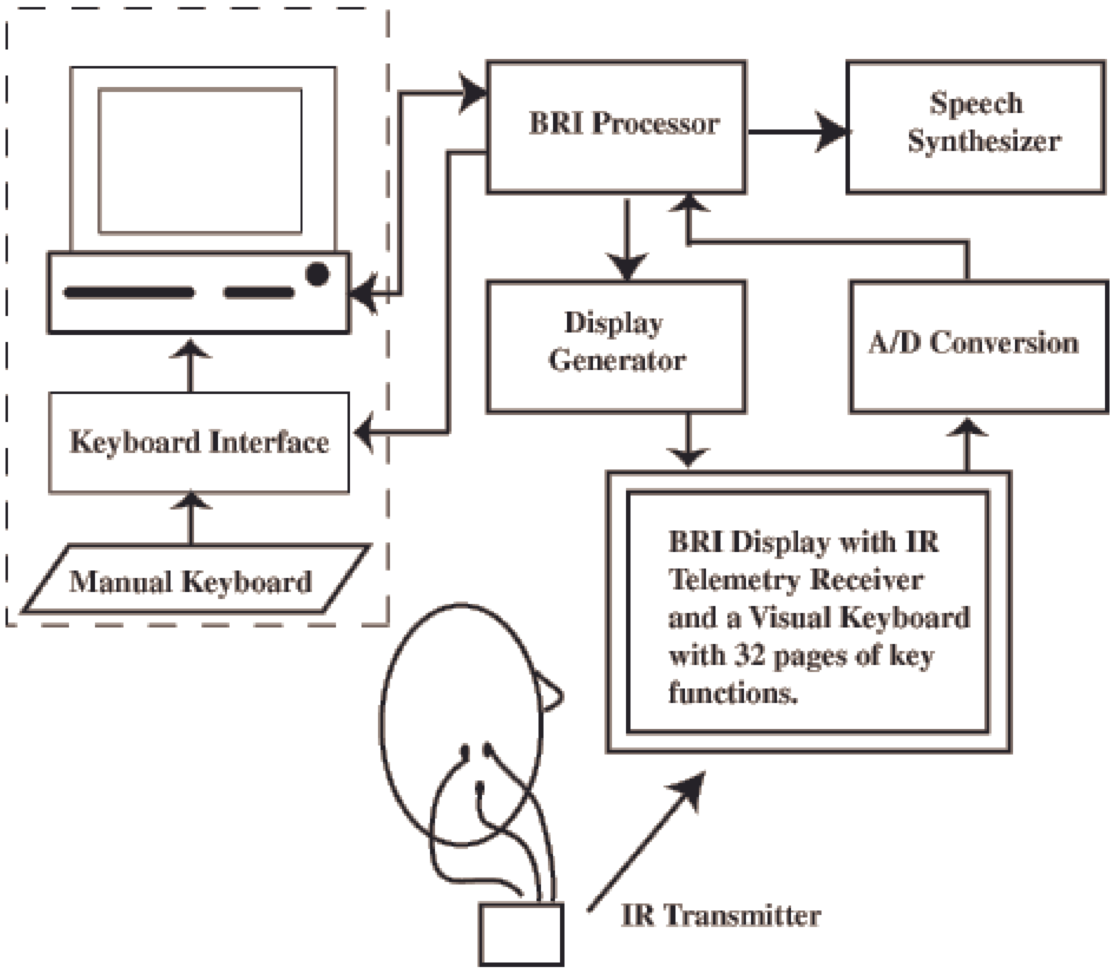


Figure 2-2: A schematic of the Brain Response Interface (BRI) system as described by Sutter.

A positive aspect of using a longer latency component such as the P3 is that it enables differentiating between when the user is looking at the computer screen or looking someplace else (as the P3 only occurs in certain stimulus conditions). Unfortunately, this system is also agonizingly slow, because of the need to wait for the appropriate stimulus presentation and because the stimuli are averaged over trials. While the experimental setup accomplishes its main goal of showing that the P3 may be used for a BCI interface, the subjective experiences of a subject with this system have yet to be considered. The 10 Hz rate of flashing may fatigue users as Sutter mentions and this rate of flashing may cause epilepsy in some subjects.

2.3.3 ERS/ERD Cursor Control

Pfurtscheller and his colleagues take a different approach [58] [66] [67] [68][39]. Using multiple electrodes placed over sensorimotor cortex they monitor event-related synchronization/desynchronization (ERS/ERD) [64]. In all sessions, epochs with eye and muscle artifact are automatically rejected. This rejection can slow subject performance speeds.

As this is a research system, the user application is a simple screen that allows control of a cursor in either the left or right direction. In one experiment, for a single trial the screen first appears blank, then a target box is shown on one side of the screen. A cross hair appears to let the user know that he/she must begin trying to move the cursor towards the box. Feedback may be delayed or immediate and different experiments have slightly different displays and protocols. After two training sessions, three out of five student subjects were able to move a cursor right or left with accuracy rates from 89-100%. Unfortunately, the other two students performed at 60% and 51%. When a third category was added for classification, performance dropped to a low of 60% in the best case [39].

The architecture of this BCI now contains a remote control interface that allows controlling the system over a phone line, LAN, or Internet connection. This allows maintenance to be done from remote locations. The system may be run from a regular PC, a notebook, or an embedded computer and is being tested for opening and closing a hand-orthesis in a patient with a C5 lesion. From this information, it appears that the user application must be independent from the BCI, although it is possible that two different BCI programs were constructed.

This BCI system was designed with the following requirements in mind [31]:

1. The system must be able to record, analyze, and classify EEG-data in real-time.
2. The classification results must have the ability to be used to control a device on-line.
3. The system must have the ability to have different experimental paradigms and give multimodal stimulations.
4. The system must display the EEG channels on-line on a monitor.
5. The system must store all data for later off-line analysis.

The system has the ability to record up to 96 channels of EEG simultaneously through the use of multiple A/D boards. Simulink and Matlab are the two software packages used: Simulink to calculate the parameters of the EEG state in real-time and Matlab to handle the data acquisition, timing, and experimental presentation. This design has the benefit of separating data processing from acquisition and application concerns. This may lead to greater encapsulation of data and maintainability. This design has the drawback of trying to use Matlab for both data acquisition and the BCI application. For simple applications such as the cursor control task, this decision makes sense. When the application becomes more complex this design decision may lead to problems. Matlab is not an object-oriented language and data encapsulation is not necessarily easy to accomplish. This may lead to poor maintainability. In addition, the system depends on Matlab for all program capabilities. This is fine for simple graphical interfaces, but may break down when the programmer wants to communicate with another program or even over the web. For these cases Matlab may offer several special program extensions, but buying many extensions becomes problematic and expensive. It would be easier to enable the application creator to use a variety of languages for the application.

2.3.4 A Steady State Visual Evoked Potential BCI

Middendorf and colleagues use operant conditioning methods in order to train volunteers to control the amplitude of the steady-state visual evoked potential (SSVEP) to florescent tubes flashing at 13.25 Hz [84][54][36]. This method of control may be considered as continuous as the amplitude may change in a continuous fashion. Either a horizontal light bar or audio feedback is provided when electrodes located over the occipital cortex measure changes in signal amplitude. If the VEP amplitude is below or above a specified threshold for a specific time period, discrete control outputs are generated. After around 6 hours of training, users may have an accuracy rate of greater than 80% in commanding a flight simulator to roll left of right.

In the flight simulator, the stimulus lamps are located adjacent to the display behind a translucent diffusion panel. As operators increase their SSVER amplitude above one threshold, the simulator rolls to the right. Rolling to the left is caused by a decrease in the amplitude. A functional electrical stimulator (FES), has been integrated for use with this BCI. Holding the SSVER above a specified threshold for one second, causes the FES to turn on. The activated FES then starts to activate at the muscle contraction level and begins to increase the current, gradually recruiting additional muscle fibers to cause knee extension. Decreasing the SSVER for over a second, causes the system to deactivate, thus lowering the limb.

Recognizing that the SSVEP may also be used as a natural response, Middendorf and his colleagues have recently concentrated on experiments involving the natural SSVEP. When the SSVEP is used as a natural response, virtually no training is needed in order to use the system. The experimental task for testing this method of control has been to have subjects select virtual buttons on a computer screen. The luminance of the virtual buttons is modulated, each at a different frequency to produce the SSVEP. The subject selects the button by simply looking at it as in Sutter’s Brain Response Interface. From the 8 subjects participating in the experiment, the average percent correct was 92% with an average selection time of 2.1 seconds. Middendorf’s group has advocated using visual evoked potentials, in this manner as opposed to their previous work on training control of the SSVEP, for multiple reasons. Using an inherent response means that less time is spent on training. The main drawback of this group’s approach appears to be that they flicker light at different frequencies. Sutter solved the problem of flicker-related fatigue by using alternating red/green illumination. The main frequency of stimulus presentation at 13.25 Hz may also cause epilepsy.

2.3.5 Mu Rhythm Cursor Control

Wolpaw and his colleagues free their subjects from being tied to a flashing florescent tube by training subjects to modify their mu rhythm [49][88]. This method of control is continuous as the mu rhythm may be altered in a continuous manner. It can be attenuated by movement and tactile stimulation as well as by imagined movement.

A subject's main task is to move a cursor up or down on a computer screen. While not all subjects are able to learn this type of biofeedback control, the subjects that do perform with accuracy greater than or equal to 90%. These experiments have also been extended to two-dimensional cursor movement, but the accuracy of this is reported as having “not reached this level of accuracy” when compared to the one-dimensional control [84].

Since the mu rhythm isn't tied to an external stimulus, it frees the user from dependence on external events for control. The BCI system consists of a 64-channel EEG amplifier, two 32-channel A/D converter boards, a TMS320C30-based DSP board, and a PC with two monitors. One monitor is used by the subject and one by the operator of the system [50]. Only a subset of the 64-channels are used for control, but the number of channels allows recognition to be adjusted to the unique topographical features of each subject’s head. The DSP board is programmable in the C-language, enabling testing of all program code prior to running it on the DSP board. Software is also programmed in C in order to create consistency across system modules. The architecture of the system is shown in Figure 2-3.

Four processes run between the PC and the DSP board. As signal acquisition occurs, an interrupt request is sent from the A/D board to the DSP at the end of A/D conversion. The DSP then acquires the data from all requested channels sequentially and combines them to derive the one or more EEG channels that control cursor movement. This is the data collection process. A second process then takes care of performing a spectral analysis on the data. When this analysis is completed, the results are moved to dual-ported memory and an interrupt to the PC is generated. A background process on the PC then acquires spectral data from the DSP board and computes cursor movement information as well as records relevant trial information. This process runs at a fixed interval of 125 msec. The fourth process handles the graphical user interfaces for both the operator and the subject and records data to disk.

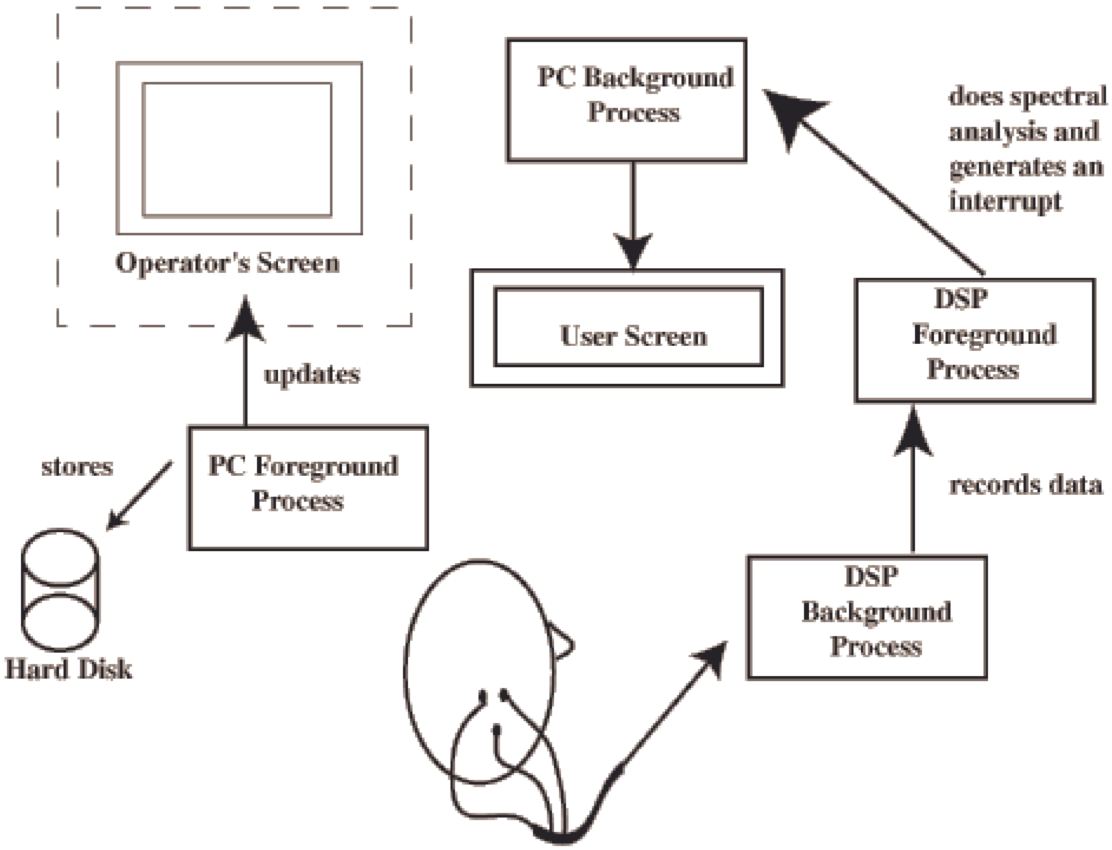


Figure 2-3: A schematic of the mu rhythm cursor control system architecture. The system contains four parallel processes. The PC foreground process must be linked to several of the other processes in order to obtain data, but these links have not been shown as they were not explicitly stated in the reference paper.

The separation of data collection and analysis enables different algorithms to be inserted for processing the EEG signals. All algorithms are written in C, which is much easier to program in than Assembly language, but is not as easy as the commercial Matlab® scripting language and environment, which contains many helpful functions for mathematically processing data. The third and fourth processes contain design decisions that may make maintenance and flexibility difficult. The graphical user interface is tied to data storage. Conversion of EEG signals to cursor control numbers happens over the DSP foreground/background processes and in the PC background process. This lack of encapsulation promises to make changing the application and signal processing difficult if such changes are planned.

2.3.6 The Thought Translation Device

As another application used with severely handicapped individuals, the Thought Translation Device has the distinction of being the first BCI to enable an individual without any form of motor control to communicate with the outside world [12]. Out of six patients with ALS, 3 were able to use the Thought Translation Device. Of the other three, one lost motivation and later died and another discontinued use of the Thought Translation Device part way through training, and then later was unable to regain control. The paper implies that users do not want to use the BCI unless they absolutely must, but does not disambiguate subjective user satisfaction of the system from general user depression.

The training program may use either auditory or visual feedback. The slow cortical potential (see Table 2-1) is extracted from the regular EEG on-line, filtered, corrected for eye movement artifacts, and fed back to the patient. In the case of auditory feedback, the positivity/negativity of a slow cortical potential is represented by pitch. When using visual feedback, the target positivity/negativity is represented by a high and low box on the screen. A ball-shaped light moves toward or away from the target box depending on a subject's performance. The subject is reinforced for good performance with the appearance of a happy face or a melodic sound sequence.

When a subject performs at least 75% correct, he/she is switched to the language support program. At level one, the alphabet is split into two halves (letter-banks) which are presented successively at the bottom of the screen for several seconds. If the subject selects the letter-bank being shown by generating a slow cortical potential shift, that side of the alphabet is split into two halves and so on, until a single letter is chosen. A "return function" allows the patient to erase the last written letter. These patients may now write email in order to communicate with other ALS patients world-wide. An Internet version of the thought translation device is under construction. The authors comment that patients refuse to use pre-selected word sequences because they feel less free in presenting their own intentions and thoughts.

2.3.7 An Implanted BCI

The implanted brain-computer interface system devised by Kennedy and colleagues has been implanted into two patients [42][43]. These patients are trained to control a cursor with their implant and the velocity of the cursor is determined by the rate of neural firing. The neural waveshapes are converted to pulses and three pulses are an input to the computer mouse. The first and second pulses control X and Y position of the cursor and a third pulse as a mouse click or enter signal.

The patients are trained using software that contains a row of icons representing common phrases (Talk Assist developed at Georgia Tech), or a standard ‘qwerty’ or alphabetical keyboard (Wivik software from Prentke Romich Co.). When using a keyboard, the selected letter appears on a Microsoft Wordpad screen. When the phrase or sentence is complete, it is output as speech using Wivox software from Prentke Romich Co. or printed text. There are two paradigms using the Talk Assist program and a third one using the visual keyboard. In the first paradigm, the cursor moves across the screen using one group of neural signals and down the screen using another group of larger amplitude signals. Starting in the top left corner, the patient enters the leftmost icon. He remains over the icon for two seconds so that the speech synthesizer is activated and phrases are produced. In the second paradigm, the patient is expected to move the cursor across the screen from one icon to the other. The patient is encouraged to be as accurate as possible, and then to speed up the cursor movement while attempting to remain accurate. In the third paradigm, a visual keyboard is shown and the patient is encouraged to spell his name as accurately and quickly as possible and then to spell anything else he wishes.

This system uses commercially available software and thus the BCI implementation does not have to worry about maintenance of the user application. Unfortunately, the maximum communication rate with this BCI has been around 3 characters per minute. This is the same rate as quoted for EMG-based control with patient JR and is comparable with the rates achieved by externally-based BCI systems. Kennedy has founded Neural Signals, Inc. in order to help create hardware and software for locked-in individuals (see www.neuralsignals.com for more information) and the company is continually looking for methods to improve control. JR now has access to email and may be contacted through the email address shown on the company’s web site.

3 Designing a Flexible BCI

... the requirements of a software system often change during its development, largely because the very existence of a software development project alters the rules of the problem. Seeing early products, such as design documents and prototypes, and then using a system once it is installed and operational, are forcing functions that lead users to better understand and articulate their real needs. At the same time, this process helps developers master the problem domain, enabling them to ask better questions that illuminate the dark corners of a system's desired behavior.

- Booch [13]

3.1 The Important Components of a BCI for Research

After reading the BCI background material in Chapter 2, the reader will notice several similarities between different BCI systems. All BCIs contain a signal processing module or engine and use it with a specific application. The application may present various stimuli for the user to react to as well as cause various actions, such as turning on a light, to occur. These similarities may be seen clearly in the Brain-Computer Interface Technology Conference of 1999 where separate speaker panels existed for signal analysis and applications [14]. Less obvious, but equally important, all BCIs must have communication between the user application and the rest of the BCI. This communication may range from internal lightweight thread communication to communication over a network.

Existing BCIs are monolithic creations or designed to run on special hardware. This is to be expected, as the primary consideration in building a BCI has most often been speed. As a field matures, one quite often sees prototypes and initial systems put onto special hardware boards. This enables a system to run faster. BCIs are not yet a mature technology, and so flexibility deserves special attention in the design of a BCI system. Special hardware makes flexibility difficult and often increases the cost of a BCI system. When a better signal processing algorithm becomes available, it is difficult to make use of the new algorithm. User applications may also become more difficult to change.

An example of inflexibility due to the use of special hardware may be seen in Sutter's 1991 Brain Response Interface. This interface works well for its intended purpose: using visual evoked potentials for environmental control and communication. Subjects need to have good eye control in order to use this system. Say that the designer wanted to extend the system to use auditory displays. This system might use an auditory evoked potential rather than a visual one and different signal processing would most likely need to be done. In addition, the main user application, a screen with a grid of visual choices that alternate in a red/green check pattern, would need to be significantly changed. These changes would necessitate changing the hardware display generator and keyboard interface: both special purpose hardware boards. These boards would have to be redesigned for every new user application and signal recognition routine change.

While the user application may be a part of the whole BCI system, as in the cursor control system used by Wolpaw et. al., it does not have to be. As an example, web pages are not part of the browser program that presents them. Imagine if all web pages had to be included as part of the browser installation! The browser size would be huge and individuals could not easily modify their own web pages. As BCI researchers, we do not know the full range of user applications that should run on the BCI and change should be planned for. Just as the browser is a client that shows web pages sent over the Internet by servers, the BCI user application should be a client that communicates with a BCI server or backend.

Why create a flexible architecture when current systems work? Years of experience in software engineering tell us that monolithic or inflexible systems are difficult to maintain and extend. This is one of the reasons that object-oriented design and programming became so popular. In object-oriented design, abstraction, encapsulation, modularity, and hierarchy are all carefully considered [13]. Component-oriented design takes this approach one step further by taking into account programming/scripting language use, libraries, interfaces, architecture patterns, frameworks, and whole system architectures [82]. Techniques from software engineering are useful in designing a flexible BCI and the construction of such a BCI is discussed in this chapter. As the quote at the beginning of the chapter indicates, the fact that there are existing systems that demonstrate BCI feasibility frees the designer to worry about issues such as flexibility: a “darker corner” of a BCI system.

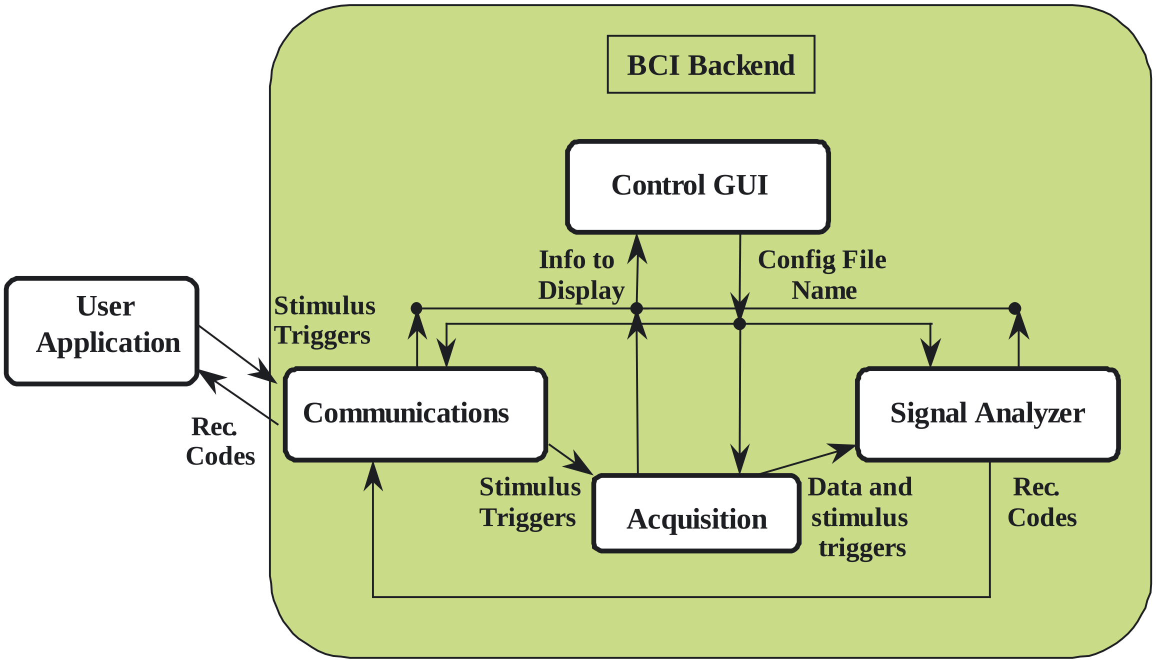


Figure 3-1: The general program architecture of the University of Rochester BCI. Each small box represents a different BCI component and the arrows represent connections between the components.

3.2 System Requirements and Architecture

The hardware requirements of the present system were kept as simple as possible and are the only part of the system that is based around special equipment. The main hardware constraints are shown in Figure 3-2.

In order to maximize flexibility, the software architecture has been modularized and as much off-the-shelf software has been used as possible. Figure 3-1 shows the basic components of the software. The user application is a separate process from the rest of the BCI backend and signal processing also exists as a separate process. This design enables system multitasking, so that data is never missed. The system has been tested with up to 32 channels of EEG data acquisition running at 1000 Hz. Overlapping EEG data trials are also supported. This means that the time slice of data analyzed may be longer than the stimulus trigger rate.

Hardware Constraints

1. The acquisition part of the BCI must be run on a PC (running winX) with as few special requirements as possible.
2. Any system shall use the 32-channel Grass amplifiers loaned by the University of Rochester Clinical Neurophysiology Laboratory in the Department of Neurology. It is to be recognized that if the system is to become usable by the handicapped, it will need to have a more inexpensive set of amplifiers, thus flexibility to choose different acquisition hardware is needed.
3. The system will use the Keithley Metrabyte A-D card. All EEG data acquisition shall have the ability to be saved to the Neuro Scan continuous file format (.cnt) (see www.neuro.com/neuroscan for format information) in order to make data viewing and editing uniform and because the continuous file format compresses the data, thus yielding smaller file sizes. Software should be built in order to make the possibility of using different packages as easy as possible.

Figure 3-2: The basic hardware requirements for the BCI system.

Acquisition

Description: Data input may be either from the Keithley A-D board (on-line) or a Matlab® (off-line) file. The file formats supported are the NeuroScan continuous file format (.cnt), the NeuroScan epoched file format (.eeg), and the native Matlab file format (.mat). The name of the configuration file will be passed to the acquisition module on start of the acquisition module thread and the appropriate configuration information will be loaded for the file. The acquisition module will be responsible for passing configuration information for display on the control GUI (graphical user interface). The appropriate signal input is passed to the signal analyzer. Data input is synchronized in time with any stimulus triggers sent from the communications module. Data acquisition is an intensive module and shall run in its own thread. Data is scaled into microvolts in the acquisition part of the class although an external calibration file is necessary at this time. It is necessary to read a calibration file into the program every time the amps are re-calibrated (every time they're used!). The calibration routine shall save whatever the last calibration file was and load that up in order to make multiple runs of the program easier. A wrapper will be needed for the acquisition module in order to make the interface generic. The Control GUI shall not need to know what kind of acquisition is being done and what classes need to be constructed.

Expected changes: Acquisition can be done on different hardware and the program architecture shall facilitate this. Calibration shall eventually be done from within the program.

Basic Testing Requirements: The acquisition module shall take in data and save it to a NeuroScan .cnt file at various frequency rates with a various number of channels.

Figure 3-3: Basic software requirements for the acquisition component of the BCI system.

The original design called for using the Neuro Scan signal acquisition package (see www.neuro.com/neuroscan for more information). This package has a feature that enables all continuously acquired data to be passed to an external program via a program written by the user in either Visual C++® or Visual Basic®. Unfortunately, this program did not perform fast enough for the on-line processing needed in a BCI, thus the signal acquisition software was written in-house. Both on-line and off-line acquisition are supported: the on-line acquisition uses a special hardware configuration, while the off-line acquisition uses Matlab®. The requirements for the acquisition software are shown in Figure 3-3.

The BCI includes a user application component that may run on a separate computer through the use of a serial port as discussed by Bayliss and Ballard [7][7]. Please see Figure 3-4 for a description of the user application. The application was designed as a separate component in order to facilitate the creation of different applications. While only one application runs at a time with the current BCI, the possibility for different applications to run at the same time exists. The architecture also allows for the addition of communication over a network rather than a serial port through the use of a communications component (see Figure 3-6). In order to showcase the fact that an application need not be written on a similar computer to the one the BCI is run on, BCI applications have been done in virtual reality on a Silicon Graphics Onyx computer.

<u>User Application</u>

Description: The user application is by far the most important part of the BCI for the user. It is necessary to support a wide variety of user applications on a wide variety of machines including PCs and the SGI Onyx in the VR lab. The easiest way to do this is to use a serial port interface in order to pass standardized recognition codes from the communication module to the user application. This requires the use of two machines or two serial ports on one machine.

Expected changes: It shall be possible to easily extend the interface technique to work between two processes on the same machine or between different machines on a network. The whole application may change. Multiple user applications using different signals may be run at the same time. Recognition codes shall be easy to change.

Testing requirements: The BCI backend shall be able to output dummy recognition codes in order to test the user application.

Example Recognition Codes

Description	Code
NOTHING	0
BLINK	1
EYE LEFT	2
EYE RIGHT	3
FOREHEAD MOVEMENT	4
CONTINGENT NEG. VARIATION	5
P3	6

Figure 3-4: The basic software requirements for the user application.

One of the biggest challenges in BCI research is to create the ultimate signal processing routine. A separate component for signal processing exists in the BCI system. In order to make changing the signal processing as simple as possible, Matlab® was chosen for all signal processing. As new signal processing routines come out, the BCI program does not need to be changed in order to use them. A configuration file names the Matlab® function to be called during the use of the BCI and changing the configuration file may change what signal processing algorithm is run. Sample configuration file requirements are shown in Figure 3-7 and the software requirements for the signal analyzer are shown in Figure 3-8. A couple of sample configuration files are then shown in Figures 3-9 and 3-10. A separate graphical user interface (GUI) maintains information about the configuration setup (see Figure 3-5), so that users are not left in doubt as to the state of the main variables in the BCI backend.

Control Graphical User Interface (GUI)

Description: The control GUI has the job of starting all BCI threads, allowing users to terminate the program at any time, giving the name of the configuration file to all parts of the BCI backend needing them, and displaying configuration data obtained for parts of the BCI backend.

Expected changes: This configuration file might contain details of the last program run so that it may ask the user if certain data files shall be overwritten or renamed as it is undesirable to lose data when one forgets to change the configuration file information. The control GUI has a user interface that could be used to change configuration data as well as display it. The user interface to the control GUI is sure to change.

Basic Testing Requirements: Starting and stopping the threads in the program are the most important things to test! Passing configuration file information into the other threads and properly displaying configuration information from all applicable threads need to be tested.

Figure 3-5: The requirements for the BCI backend graphical user interface.

Communications

Description: The communications module is responsible for communicating between the user application and the rest of the BCI backend. The communications module must sync trigger inputs from the user application up with the EEG data being acquired. In order to do this information must be exchanged between the acquisition module and the communication module. The communication module must also allow the signal analyzer to send recognition data to the user application via a serial port The communication module does not process the recognition codes, but passes them directly to the user application. The name of a configuration file will be obtained from the control GUI and will use this file to set configurable information. It will be responsible for sending textual information back to the control GUI when requested. A wrapper needs to be put around the communication module, so that the control GUI doesn't have to know what specific classes to create for communication.

Expected changes: Currently, all communication is done through the serial port. Network or other communication would be an extension of the communication module.

Basic Testing Requirements: Timing analysis of sending and receiving data from the appropriate parts of the BCI needs to be performed.

Figure 3-6: The requirements for the communications parts of the BCI backend.

Configuration File

Description: While the configuration file is just a file and not a piece of code, it needs to have a standardized syntax so that all appropriate parts of the BCI may read the information found in the configuration file. The general syntax will consist of a configuration variable name in either capital or lower case letters followed by the value for that variable on the same line. Configuration file comments will consist of a line beginning with a hash mark (#) and will end with a new line. Only one variable will be allowed per line.

Expected changes: Different configuration information might be needed in the future.

Basic Testing requirements: Testing would require using different combinations of variables on the system as a whole.

Items to configure in the BCI backend

1. Input type: keithley or matlab (default: matlab)
2. EEG epoch size: a number > 0 (default: 500)
3. Number of channels in data acquisition: a number between 1 and 32 (no default)
4. Accept stimulus trigger codes or not. This also tells whether or not to wait for stimulus codes to occur: 1 = yes, 0 = no (default: 0)
5. Write recognition codes out to the user application: 1=yes, 0=no (default: 0)
6. Save recognition information to a file: needs the full file name and path (default: don't save)
7. Save EEG data collected to a file: needs the full file name and path (default: no file)
8. Stimulus codes that should not be included in recognition: numbers from 1 – 255 (no defaults)
9. Accept Matlab file names for input files: needs the full file name and path (no default)
10. Accept Matlab file names for parameter files used in recognition: needs the full file name of the .mat file (no default)
11. Set the number of pre-epoch data points collected for triggered recognition: number must be >= 0 (default: 0)
12. Read in an amplifier calibration file: requires the full file name and path (no default)
13. Read in an A-D board config file: requires the full file name and path (no default)
14. The name of the Matlab algorithm to use: a text name (no default)
15. The number of trials to collect data for: a number greater than 0 (default: 1000)

Figure 3-7: The requirements for the configuration file used by the BCI backend.

~~Signal Analyzer~~

Description: The job of the signal analyzer is to take a matrix of EEG data input and produce recognition codes. In order to do this the analyzer may need to reduce artifacts such as eye movements and eye blinks as well as run pattern recognition routines. Signal analysis will be done in Matlab.

Expected Changes: Signal analysis routines in style change daily. Hence, the use of Matlab in order to facilitate trying out new algorithms. Future extensions could involve allowing independent signal processing algorithms to run rather than Matlab. Robustness is a current concern that needs to be addressed. BCI backend code should have to be changed as little as possible when changing algorithms. The number and type of recognition codes will probably change. Multiple algorithms may be used at one time in the BCI.

Testing Requirements: The signal analyzer should have a random signal output in order to facilitate testing the rest of the system. All Matlab routines should be tested on known files before inputting into the signal analysis module so that they can be re-tested on the known results from within the program.

Figure 3-8: The software requirements for the signal analyzer.

```
# keithleybciconfig.cfg
input keithley

# keithley necessary information
adrate 500
nchannels 5
epochsize 800
pretrial 0
kconfigfile c:\das1800\asowin\das1800.cfg
calibfile c:\tmp\testcalib.dat

# stimulus triggers that should be ignored
exception 255
exception 254

# communication information
writetriggers 1
readtriggers 1

# matlab needed info
mparamfile c:\tmp\tmpparam.mat
savedatafile c:\tmp\test.cnt
saveeventfile c:\tmp\testevnt.mat
algorithm returnRandomWithStats
numtrials 500
```

Figure 3-9: A sample configuration file for on-line EEG experiments.

```
# matlabbciconfig.cfg
input matlab

# stimuli to be ignored
exception 255
exception 254

# matlab info
minputfile c:\tmp\group1train.eeg
mparamfile c:\eeg\trainparam.mat
saveeventfile c:\tmp\testevent.mat
algorithm varAveP3WithStatsOnline
```

Figure 3-10: A sample configuration file for EEG experiments doing off-line analysis with data loaded into Matlab.

3.3 Class Design

Several parts of the BCI backend are written in Visual C++® and as such, the software classes have been designed in order to allow maximum flexibility. This means that some of the classes serve as higher order wrappers in order to hide or encapsulate the internal workings of classes that may have multiple components underneath them. The design contains the following elements in order to facilitate flexibility:

1. Conceptually different parts of the system have been made into separate classes.
2. There should be a simple graphical user interface that shows configuration details without knowing anything about the underlying configuration. Each separate part of the system should handle its own configuration, so that configuration information is not stored/duplicated in multiple parts of the system.
3. The main conceptual objects in the system should have a common interface irrespective of their underlying nature. For instance, the communications part of the system should have a common interface which provides read/write access without having to specify for each access whether the read is from the serial port or over the network.
4. Since configuration information is kept in a file and multiple classes need access to this information, it should not matter what class reads this information first, but each class should individually read and handle its own configuration data.

The overall class architecture is shown in Figure 3-11 in UML (the Unified Modelling Language – see [27] for details). In UML, a class is represented by a box,

and arrow pointing towards a box means that the class has been inherited from, a dashed line imposes a named constraint on a class relationship, and textual annotations describe the relationships between boxes. If one class box is inside of another, then it is a contained class.
The name of the class and its instantiation appear in the top of the box.

The following naming conventions have been observed. All classes start with a capital C. A class name is followed by BCI when it is a class specially created for the BCI backend program. Lightweight threads always contain the name Thread in them to indicate their function. In a likewise fashion, base classes contain the name Base.

The only class shown that is not a class created for the BCI is Microsoft's CWinApp class. This is the general class from which applications are derived in Visual C++® and is shown in Figure 3-11.

3.4 Main BCI classes and their functionality

The main BCI classes are those that do the majority of the work in the BCI backend. The relationships of these classes to each other are shown in Figures 3-11, 3-12, 3-13 and 3-14. These classes and their functionality are listed below:

1. **CBCIControlApp**: The main application class. This class starts up and closes all threads for the functionality of the BCI backend.
2. **CBCIControlDlg**: The class that implements the requirements for the Control GUI.
3. **CBCICommBaseThread**: A base class that adds communication specific functionality that is needed for all BCI communication classes.
4. **CBCISerialCommThread**: A class that inherits functionality from CBCICommBaseThread and implements communication over the serial port of a PC.
5. **CBCISigProcBaseThread**: A base class that adds signal processing specific functionality that is needed for all BCI signal processing classes.
6. **CBCIMatlabSigProcThread**: A class that uses the Matlab® engine in order to process all data with a routine provided by the configuration file.
7. **CBCIAcqBaseThread**: A base class that provides EEG acquisition specific functionality that is needed for all BCI acquisition classes.
8. **CBCIMatlabAcqThread**: A thread class that acquires EEG data from a stored file that is read via Matlab.
9. **CBCIKeithleyAcqThread**: A thread class that acquires EEG data from the Keithley Metrabyte A/D board.

While the signal processing is currently done in Matlab®, the class design gives the infrastructure for adding the ability to handle free standing signal processing routines. In addition, one can easily imagine adding different classes for handling communications and acquisition.

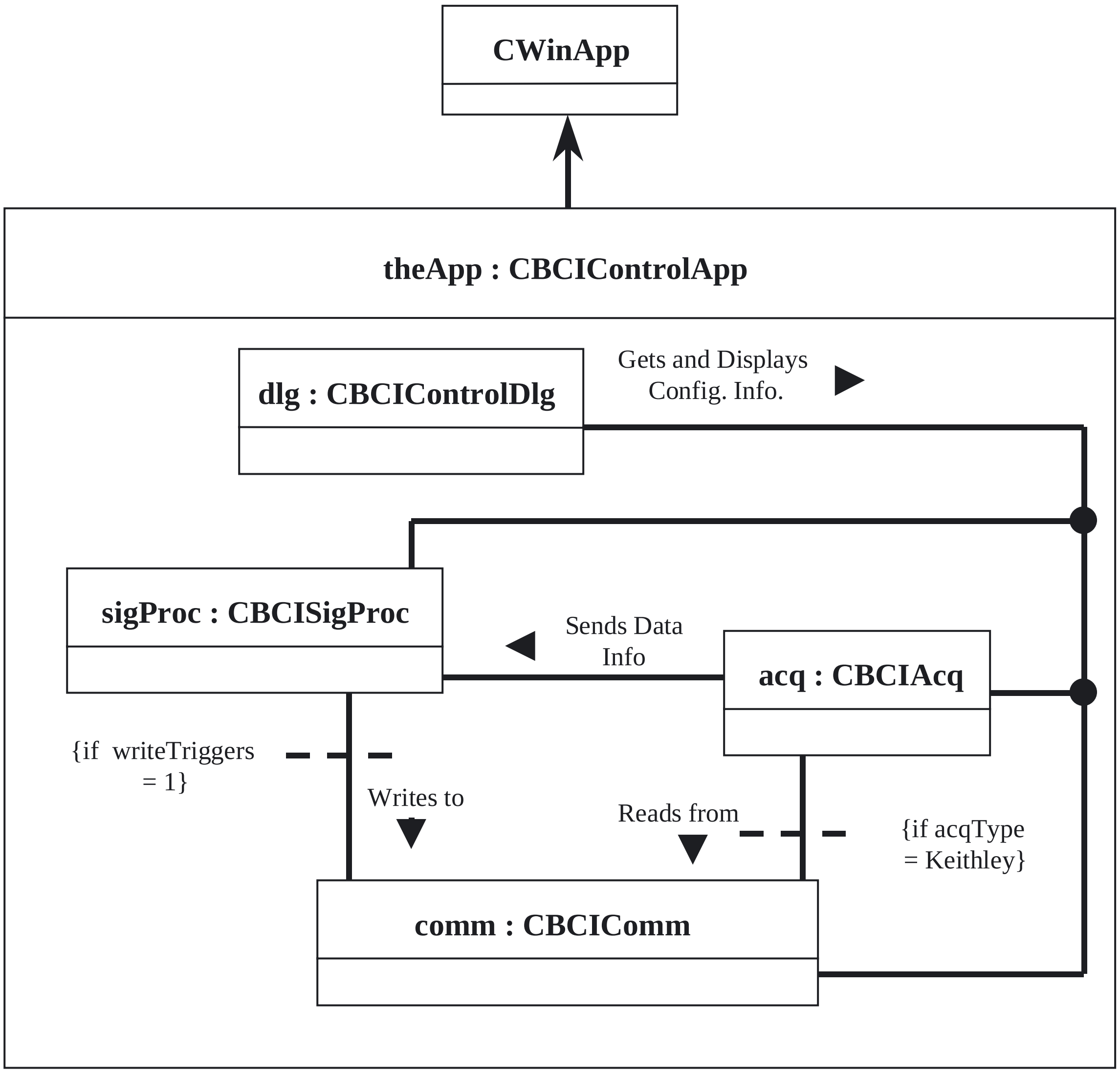


Figure 3-11: The overall class architecture for the BCI backend. True to most applications now written in Visual C++®, the main application inherits from the CWinApp class. The application contains a control dialog box, a signal processing object, an EEG acquisition object, and a communications object. These classes perform the necessary functions of acquiring data, processing it, and then sending it to/from the separate user application.

3.5 Interface classes

Interface classes act as wrappers around the communication, signal processing, and acquisition classes. These classes are used to handle all configuration needed before a thread is started and are in charge of stopping their respective threads. The interface classes make sure that the interface between a calling class and contained classes is standard. For instance, when one wants to write a byte of data, one does not want to have to worry about whether the communication medium is a serial port or network. The interface class for BCI communications hides the details of the medium and allows the calling function to have no knowledge of the communications medium. The three interface classes and their functionality are shown in Figures 3-12, 3-13, and 3-14. They are also described below:

1. **CBCIComm**: The communications class that encapsulates knowledge of what communication medium is being used.
2. **CBCIAcq**: The acquisition class that encapsulates knowledge about how the data is being acquired and allows a calling class to just grab data.
3. **CBCISigProc**: The signal processing class that encapsulates knowledge about what signal processing routine is being used and allows the calling class to send data to the routine and returns recognition information for the user application.

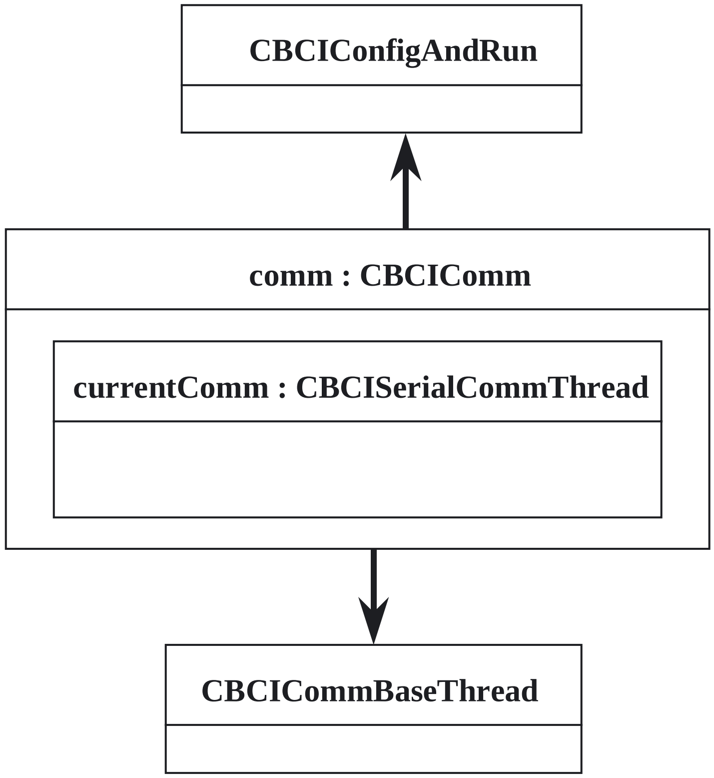


Figure 3-12: The communications objects for the BCI backend. CBCIComm acts as an interface or wrapper that encases the details of what kind of communications occur within the object. The only type of communication thread that has been written to date is for the serial port. This thread inherits from a base communication thread that provides the outline of the necessary read/write functionality for all communications classes.

3.6 Helper classes

There are two classes that help create the functionality used in the BCI backend. These classes are:

1. **CMatlab**: A class that starts up the Matlab® engine and facilitates passing information to/from Matlab®.
2. **CBCIConfigAndRun**: A class that provides file reading capabilities for reading configuration files and provides various helpers for controlling threads.

3.7 Demonstrations of application and signal processing flexibility

One claim of the thesis is that the BCI is designed to allow flexibility in both applications and signal processing. To this end a few diverse applications with very different applications and signal processing needs are presented: a P3-based environmental control application done in a virtual environment, an eye and forehead movement based application to play a Pong game programmed in QBasic, and another eye and forehead movement application written to play a Tetris game programmed in the Java language.

3.7.1 Environmental control in a virtual apartment environment

The main results from an experiment done in a virtual apartment environment may be found in Chapter 6. The goal of this application is to simulate an apartment where a user may choose to control various environmental items. Common items that are controlled include the lights (on/off), the radio (on/off), and the television (on/off), as well as simple verbal commands to a virtual person (hi/bye). This particular application shows the flexibility of the architecture in a couple of ways:

1. The user application takes place on a Silicon Graphics Onyx machine where the virtual environment is rendered. This demonstrates that the user application does not need to be limited to running on a PC and that an application may make use of a complex environment such as the virtual apartment environment.
2. The signal processing algorithm was chosen separately from the user application and several algorithms were tried before the final one that was used in the experiment was chosen.

The BCI backend runs on a Pentium PC and communicates with the Silicon Graphics (SGI) machine through a serial port connection. Communication takes place approximately every half a second, although communication rates as little as 100ms apart were tried. The application enabled tests to be performed in order to determine whether looking at the apartment on a computer monitor was better or worse for subjects than being fully immersed in the virtual apartment.

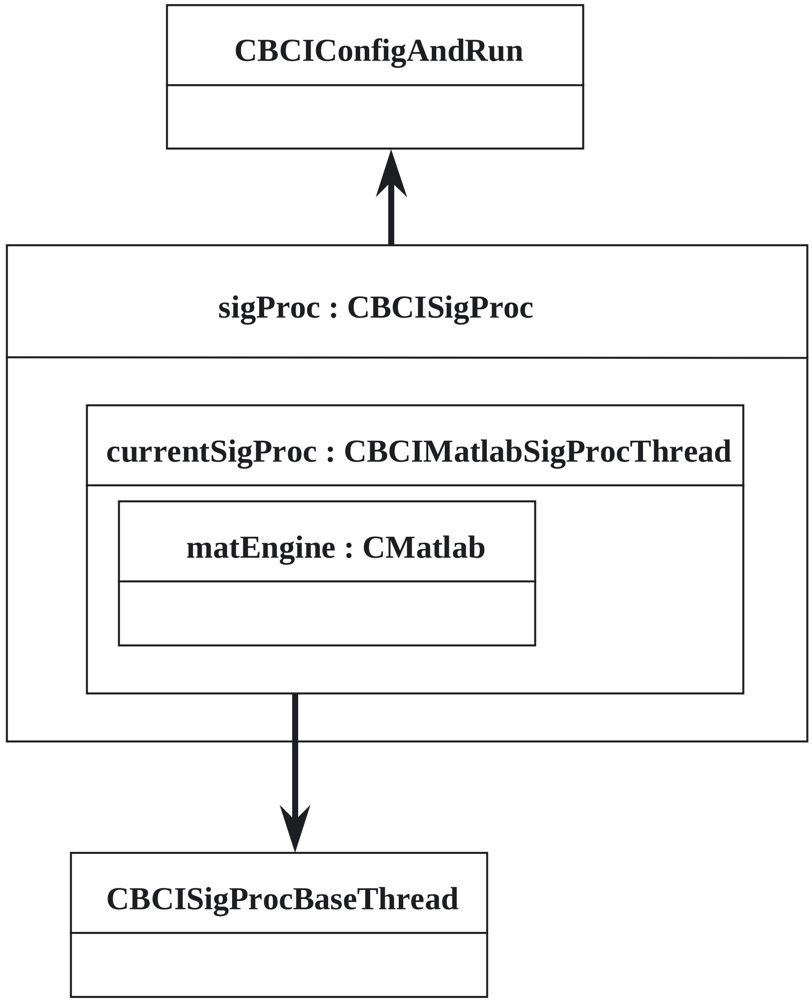


Figure 3-13: The signal processing objects for the BCI backend. CBCISigProc acts as an interface or wrapper that encases the details of the signal processing within the object. The only type of signal processing thread that has been written to date is for the running routines in Matlab®. This thread inherits from a base signal processing thread.

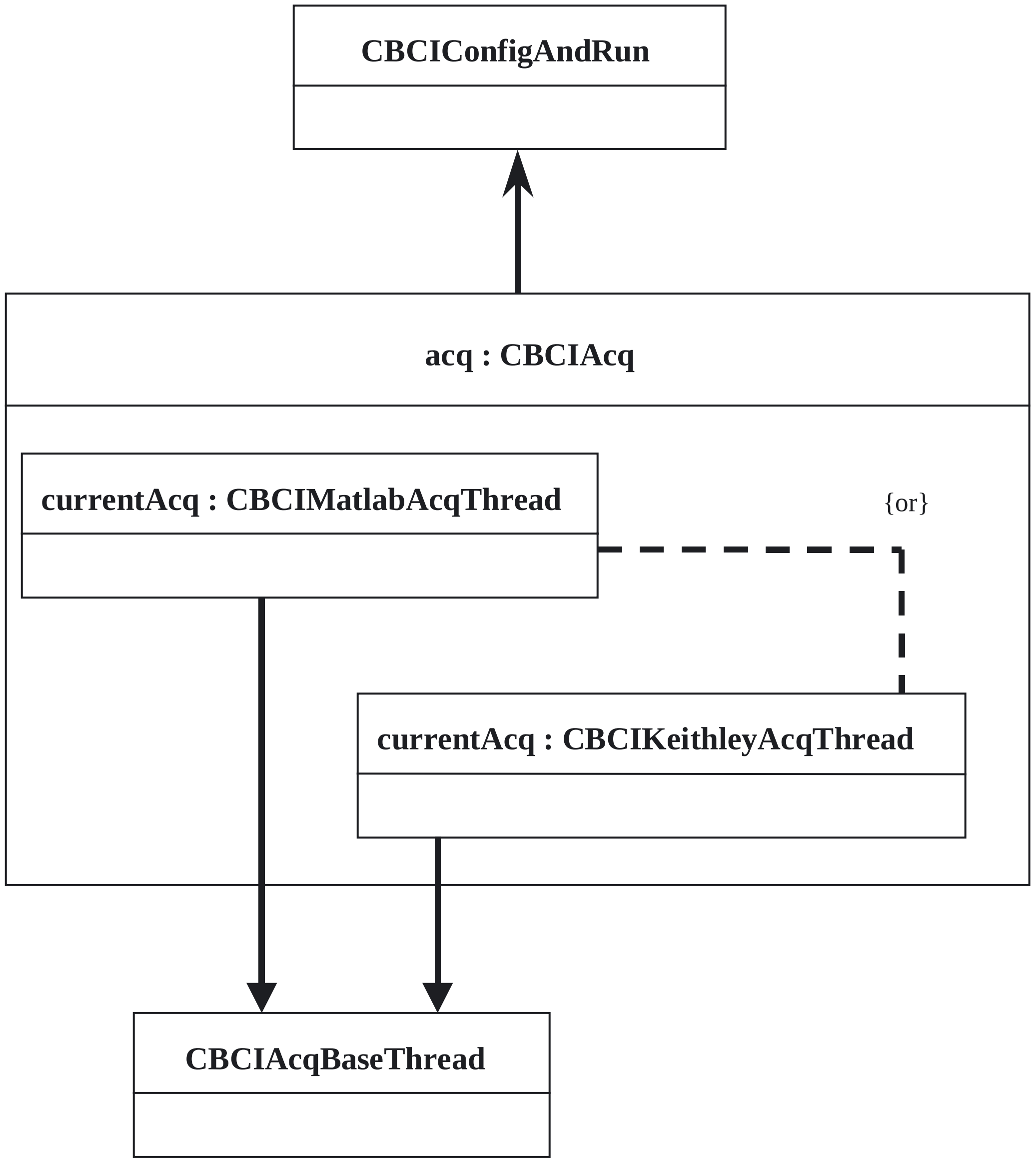


Figure 3-14: The acquisition objects for the BCI backend. CBCIAcq acts as an interface or wrapper that encases the details of the kinds of acquisition that may occur within the object. Two kinds of acquisition classes exist: a class for acquiring data from files read by Matlab®, and a class to acquire on-line data from a Keithley Metrabyte A/D card. This thread inherits from a base acquisition thread.

3.7.2 An eye and forehead movement based Pong game

While teaching an introductory Computer Science course for high school students during the summer of 1999, several of the students wanted a bigger challenge. A separate programming project was given to these students: take a game with published source code off of the Web and change it so that it could be played with eye movements. The signal processing routine was simple and provided for the students. It consisted of finding blinks through a simple threshold, determining left/right eye movements through a positive/negative rise in the signal that was smaller than a blink, and an algorithm for finding large forehead movements by using an electrode on top of the head and setting a threshold to determine when a forehead movement occurred.

While this routine was simple and prone to error, the goal was to determine how flexible the system was in the face of large changes in the signal processing routines and user applications. This challenge also showed that students with little prior BCI experience could successfully construct an application. Students were told that they had to control the movements of the game using the following commands:

1. Left eye movement
2. Right eye movement
3. Blink
4. Forehead movement

Out of two student groups, one chose to use a Pong game written using the QBasic programming language. The Pong game is shown in Figure 3-15 and consists of two paddles. One paddle is moved left, right, or left in a halted position. This paddle is played by the user. The other paddle is played by the computer. The goal of the game is to hit the ball with the users paddle in such a way that the computer cannot hit the ball back.

Since QBasic is an interpreted language, the student team first chose to have the following command sequence:

1. Left eye movement means move the paddle left
2. Right eye movement means move the paddle right
3. Blink means occasionally put up an obstacle (this makes the game more difficult for all players).
4. Forehead movement means stop the paddle.

After correctly changing the game to work with the serial port, the students spent a whole class period playing with the game and noticed that some people were able to use different aspects of the game better than others. For instance, the signal processing routine picked up on the blinks of some people much better than their left/right eye movements. Towards the end of the session, the students decided that since the routine for blinking worked well on most people, they would use blinking to change the direction of paddle movement. Forehead movement stopped the paddle and left/right eye movements were used to occasionally put up obstacles.

Figure 3-15: A QBasic Pong game altered by students to be played using eye and forehead movements.

3.7.3 An eye and forehead movement based Tetris game

As discussed above, two student groups decided to download the source code for games from the Web. While one group decided to do a Pong game written in QBasic, the other wanted to learn about the Java language and decided to try changing the Java Tetris game shown in Figure 3-16. The goal of Tetris is to move and rotate puzzle pieces to lie as flat as possible on the bottom of the screen. When entire rows are filled in by the puzzle pieces, they disappear and the user gains points. The game ends when the puzzle pieces fill the window to the top of the playing screen.

This game proved to be much more difficult to play with eye movements because more commands are necessary. Not only do the pieces need to move left/right, but they need to rotate. Incorrect rotations caused many problems when an increase in playing level made the pieces move faster. Students were never able to overcome this difficulty.

Figure 3-16: The Java Tetris game altered by students to be played with eye and forehead movements.

3.8 Known bugs, problems, and future considerations

The program works fairly well once a configuration file is correctly written, but the user may have problems before reaching this stage of development in an experiment. The main cause of a program crash occurs when an experiment is using a new and untested signal processing routine. If the routine is incorrect and causes a Matlab error, it may then crash the rest of the program.

This crash may actually necessitate rebooting the computer if the Keithley A/D board is being used for on-line acquisition, because the Keithley board uses DMA (direct memory access) and the memory may not be correctly freed when the BCI backend crashes. Future work will include increasing the robustness of the application when using outside programs that may crash or hang.

A related problem may occur when a Matlab error occurs, but does not cause a crash. These kinds of errors may not even be reported, as often happens with logical errors. In this case the program may appear to run normally, but the results of the signal processing module will be odd. Due to these problems, it is very important to test any signal processing routines separately from the BCI backend.

Future work includes adding an on-line display of the acquired EEG data. Currently, data display must be performed in Matlab when the signals are processed, if the data is to be displayed at all. This kind of display enables the experimenter to watch only signals of imminent interest, although all signals may be recorded to disk.

4 Signal Processing of the P3 Component of the Evoked Potential

> *These data have an interesting implication. It appears that during the period in which the stimulus was presumably task-relevant some trials yield an AEP [average evoked potential] wave form that is typical of irrelevant stimuli. It is, of course, to be expected that the subjects would perform such tasks with a fluctuating degree of zeal, and we might thus be able to detect such trials. However, this cannot be established without access to some independent behavioral measure of trial-to-trial variability in "zeal".*
>
> *- Donchin [22]*

The signal processing of evoked potentials remains a difficult and unsolved problem. Donchin noticed that in the P3 component of the evoked potential, not all P3 trials appear to have a good P3 peak. This may be due to a lack of "zeal" on the individual subject's part, normal subject signal variations, outside contamination of artifacts, or even just because of the low signal-to-noise ratio available from an EEG signal. It is notable that individual subjects may cause signal recognition problems and that signal processing cannot fix these problems. That is the job of the user interface. There are a variety of different techniques for dealing with recognition problems due to the noisiness of the signal and its occlusion by artifacts.

Not all P3 recognition techniques are suitable for use in an on-line BCI. Averaging over trials has been used to improve signal detection, but this technique requires a costly trade-off between recognition accuracy and the time taken to recognize a particular signal. In order to be useful in a BCI, this trade-off necessitates severely limiting the number of trials in an average. A compromise may be reached when single trials are only averaged when the recognition routine is unsure of its results. Results from such an algorithm are presented. Three routines that may be run on single trial data in real-time are also compared. Peak picking and correlation are used individually for recognition of the P3 component. The third routine uses a robust Kalman filter to preprocess data, while using correlation to recognize the existence of the P3 component. The algorithm using the robust Kalman filter is shown to perform the best under conditions containing heavy amounts of artifact while correlation performs the best under conditions of less artifact.

Recognition algorithms normally assume that the algorithm is responding to the presence of a P3 rather than the presence of one or more artifactual signals in raw EEG data. On-line recognition algorithms must deal with or ignore artifacts. The performance of the routines in this chapter are compared on EEG recordings containing different kinds of artifacts including chewing, vertical eye movements, blinks, horizontal eye movements, foot movements, forehead movements, jaw clenching, talking, and heavy breathing. While artifacts are normally expected to produce many false positives, because a single artifact may mimic the peak of a P3 component, it is shown that the artifact ridden data tends to produce false negatives.

4.1 Why Averaging May not be Advantageous

Traditionally, evoked potentials (EPs) are obtained by averaging EEG signals from specific electrode sites over many trials. Averaging to obtain an evoked potential contains the following benefits:

6. It reduces the contribution from unrelated spontaneous EEG signals.
7. It reduces spurious noise.
8. It allows the observation of a response that would otherwise be unobservable.

An average may tell the clinician of an abnormality that a single trial could not, because of the natural variance in both the latency and morphology of an EP. Results suggest that P3 amplitude does not stabilize until approximately 20 trials have been averaged, although peak latency does not change significantly during this time frame[18].

Even though the P3 signal may vary, the on-line nature of a BCI requires that the P3 signal be recognized in a timely fashion if it is to be useful. There is a distinct trade-off between the time taken to recognize a P3 signal and the percent of P3's that are recognized correctly. During a two-dimensional cursor recognition task, Polikoff et. al. found that single trial P3's could be detected around 50% of the time based on comparisons among peak levels within a single set of trials [70]. This recognition rate rose to 60% when three successive sets of trials were averaged and continued up to a recognition rate of around 85% when 8 successive trials were averaged. The initial increase of 10%, when the average went from one to three trials, was also accompanied by a threefold increase in the amount of time necessary to arrive at a decision. With a 50% accuracy in the goal cursor direction (and assuming that errors in P3 detection are uniformly distributed over the 3 remaining target directions), the subject would be expected to need 30 steps to reach the goal of 10 steps in the attended target direction. At 4 seconds per step, this goal would require 2 minutes to accomplish. For a 60% accuracy, the expected number of steps would drop to 21, but at 12 seconds per step, it would require over 4 minutes to reach the same goal. In this case, the increased recognition rate does not justify the extra time needed.

Farwell and Donchin dealt with this tradeoff by shortening the amount of time needed for a single trial. A different stimulus flashes every 100 milliseconds in their P3 character recognition system [25]. The shortening of time means that a P3 epoch (stimulus trial) may overlap with a non-P3 epoch, and so averaging is needed to disambiguate the trials. A newer evaluation of this system, done by Spencer et. al., suggests that for off-line data analysis, when using stepwise discriminant analysis with a discrete wavelet transform, 80% recognition may be achieved for an average recognition time of approximately 6.9 characters/minute [80]. Approximately, five stimulus sets are averaged to achieve this rate. At a recognition rate of 95%, around 3.8 characters/minute may be chosen. For on-line recognition, using a bootstrapped stepwise discriminant analysis algorithm that yielded 90% recognition during off-line analysis, the recognition rate turned out to be 56%. Possibly, the algorithm did not generalize well.

4.2 Variable Averaging

While averaging over many trials throughout an experiment can reduce throughput, the variable averaging algorithm used in the on-line virtual apartment experiment of Chapter 6 attempted to mediate the effects of averaging by only averaging over trials when uncertain about the recognition results. The recognition part of the algorithm was based on the following correlation formula:

$$\rho_{\mathbf{x},\mathbf{y}} = \frac{\mathrm{cov}(\mathbf{x},\mathbf{y})}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}}$$

where **x** represented the data for a single EEG trial, **y** represented the P3 or non-P3 average, cov(**x**,**y**) was the covariance of **x** and **y**, and σ was the standard deviation of the appropriate signal. A single trial consisted of the EEG signal from area PZ associated with a particular stimulus event. Trials were either goal trials (from stimuli that the subject was supposed to accomplish) or non-goal trials. As an example, for the goal “turn on the light”, a goal trial would occur whenever a stimulus related to the light occurred and all other stimuli would be associated with non-goal trials. **x** and **y** were both 1×819 vectors. Note that the covariance is a scalar, because the **x** and **y** vectors both represent multiple instances of a single variable. There are two parameters that may be set for this recognition technique: the minimum correlation value that indicates a P3 and the trial size. The trial size went from 100 ms before a stimulus event until 1500 ms after a stimulus event for a total of 1600 ms. The threshold was varied for maximal recognition over a training set and tended to be set at around 0.5.

Averages for both P3 and non-P3 data were obtained from the training task. For each trial, two correlations were performed in order to make a decision: one with the P3 average and one with the non-P3 average. These correlations make up the basis of the variable averaging algorithm, which operated thus:

1. **Correlate**: Perform single trial correlations. If this is the last trial halt recognition.
2. **Decide whether or not to average and if not classify**: If either correlation is at or above the threshold value, classify the trial as the type of the highest correlation and return to Step 1. For trials below threshold where the correlation with the P3 average is higher than the correlation with the non-P3 average (a potentially noisy P3 trial), wait for the next trial of the same stimulus type and go to Step 3. For all other trials below threshold, classify the trial as a non-P3 and return to Step 1.
3. **When averraging, classify all trials as the type of the second trial if the second trial correlation is above threshold**: Perform single trial correlations. If either correlation is above threshold value, then classify the current and previous trial to be the type with the highest correlation and return to Step 1. Else go to Step 4.
4. **If all else fails, average the two trial correlations and classify**: If the second stimulus also has correlation values below threshold, then divide the first stimulus correlations by 2 and then average them with the most current stimulus correlations. This is done because the most recent stimulus is more indicative of the subject's current intentions (which change over time). If either of the two correlations are above the threshold value, then declare the current and previous trial to be the highest of the two correlations and return to Step 1. If the correlations are below threshold, then declare the trial to be a non-P3 and go to Step 1.

This algorithm only uses a maximum of two trials combined together. The algorithm may be easily extended to combine any number of trials before stopping and declaring a result. The experiment in Chapter 6 used an on-line version of this algorithm and results for different parts of the experiment are shown in that chapter.

The difference between using straight correlation with the averages and variable averaging is shown below in Table 4-1. In almost all cases, variable averaging performs much better than straight correlation. Does the improvement in recognition offset the extra time needed to recognize the data trials? This depends on how often trials are combined and on how large the recognition increase is. Table 4-2 shows how many trials use information from the second trial of a stimulus type. Roughly, second trial information is used for 27% of all trials. The increase in throughput is minor when the extra time taken to average the data is taken into account.

Table 4-1: Recognition percentages for all subjects using the correlation and variable averaging algorithms on the same data.

	Subjects	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
Algorithms										
Correlation	*P3*	*66*	*54*	*71*	*68*	*58*	*43*	*58*	*36*	*60*
	Non-P3	*95*	*94*	*94*	*94*	*88*	*98*	*97*	*94*	*99*
Variable Averaging	*P3*	*76*	*54*	*63*	*74*	*76*	*53*	*33*	*27*	*70*
	Non-P3	*96*	*96*	*98*	*95*	*85*	*98*	*100*	*98*	*100*

Table 4-2: The total number of trials used by the variable averaging algorithm on the first attempt (single trial), the second attempt (second trial), and after the first and second trials were combined (combined trial). The total number of trials is shown in the last row and is the sum of the previous three columns.

	Subjects	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
Trials										
Single Trial Decision		*192*	*229*	*178*	*140*	*234*	*172*	*60*	*172*	*79*
Second Trial Decision		*21*	*24*	*19*	*9*	*14*	*24*	*11*	*25*	*7*
Combined Trial Decision		*61*	*49*	*46*	*38*	*44*	*36*	*10*	*54*	*21*
Total Number of Trials		*274*	*302*	*243*	*187*	*292*	*232*	*81*	*251*	*107*

As an example, Subject 1 shows a very good recognition percentage increase as demonstrated in Table 4-1. From Table 4-2, this subject has 274 trials with 192 trials classified in a single trial and 82 classified at the second stimulus. The trials classified at the second stimulus will be referred to as combination trials, even though not all the trials are averaged (see the variable averaging algorithm at the beginning of the section).

The combination trials make up roughly 30% of the total trials. Since the particular BCI used has 5 types of stimuli (light, TV, stereo, hi, and bye) and only one stimulus at a time can be a goal, only 1 out of every 5 of these trials will actually affect throughput. This means that there will be an average of 16.4 combination trials out of a general total goal number of around 55 goals (1/5 of the total trials). At a recognition rate of 76% (see Table 4-1), this means that approximately 42 goals will be recognized and that it will take 290.4 seconds in order to recognize these 42 goals since each stimulus lasts a second and goal trials occur approximately once every 5 seconds.

This leads to an overall throughput rate of 8.6 items/min. If only correlation is used, then 66% of the 55 items on average would be recognized correctly or 36.3 goals would be achieved. This would occur in 274 seconds yielding a throughput rate of 7.9 items/min. The difference between 8.6 items/min. and 7.9 items/min. is less than 1 and turns out to be almost insignificant when the subject's concern over waiting for something to happen is taken into consideration.

4.3 Single Trial Recognition

In both the cursor control system of Polikoff and the character recognition system of Farwell and Donchin, it may be seen that reducing the number of trials averaged can give a time advantage that outweighs the increase in error rate. Interest in single trial evoked potentials has long existed. In 1966, Palmer, Derbyshire, and Lee proposed a method of analyzing individual cortical responses to auditory stimuli [61]. Donchin studied individual P3 trials using discriminant analysis in 1969 [22]. Recent work has been done by Makeig et. al. using independent component analysis (ICA) to recognize single trial P3's [38]. Other papers include a neural network for single trial recognition [63], a robust parametric estimator [47], and a maximum likelihood method [1], as well as common features such as the P3 peak voltage height [25]. Algorithms usually obtain a recognition rate anywhere from 50% to 85% when the algorithm is run on new input data obtained from the same subject as the training data.

The necessary recognition rate depends heavily on the task at hand. In a cursor task, every move must be corrected in order to get to the target cursor location. Such is not necessarily the case for spelling. Say that the average word used in a sentence is around four characters in length. If one of those 4 characters is wrong (again, assuming that errors are uniformly distributed), it is often possible to automatically correct the word using a spell checker and even if it is not, the word is usually recognizable. The recognition rate needed for this task would be a minimum of 75%. At the theoretical rate of 80% accuracy noted above, a 20-character sentence would take an average of 2.9 minutes to accomplish. There would be an average of 4 errors in the sentence. If the same sentence were spelled by averaging the number of trials necessary to achieve 95% accuracy, it would take approximately 5.3 minutes to spell and there would be an average of 1 character wrong in the sentence. The extra time needed is not worth the error reduction of three characters.

The data for this chapter consists of on-line data collected for controlling items in a virtual apartment. This experiment is discussed in Chapter 6. As in cursor control, it is very important not to have many false positives for environmental control. Too many false positives may lead to various household items turning on and off in a random manner and will prove frustrating for the user. Some commands must be reversed when they go off accidentally because they may change the state of the environment unfavorably. Examples of these kinds of commands are turning on/off the television, light, or radio. It is also possible to have commands that do not need to be corrected. For instance, when flipping TV channels, an unexpected channel change may occur. Since the user was already flipping channels, this error does not need to be corrected and may even yield a favorable result (a channel with a show that the user wants to watch).

The VR apartment has five commands: three of which should be reversed when accidentally set off (turning the LIGHT, TV, and RADIO on/off) and two of which do not need to be reversed (say HI and BYE to a three-dimensional graphics figure). In this environment, stimuli are presented at a rate of one per second. This presentation rate eliminates the remote possibility of causing a seizure in subjects and the need to average in order to disambiguate overlapping trials.

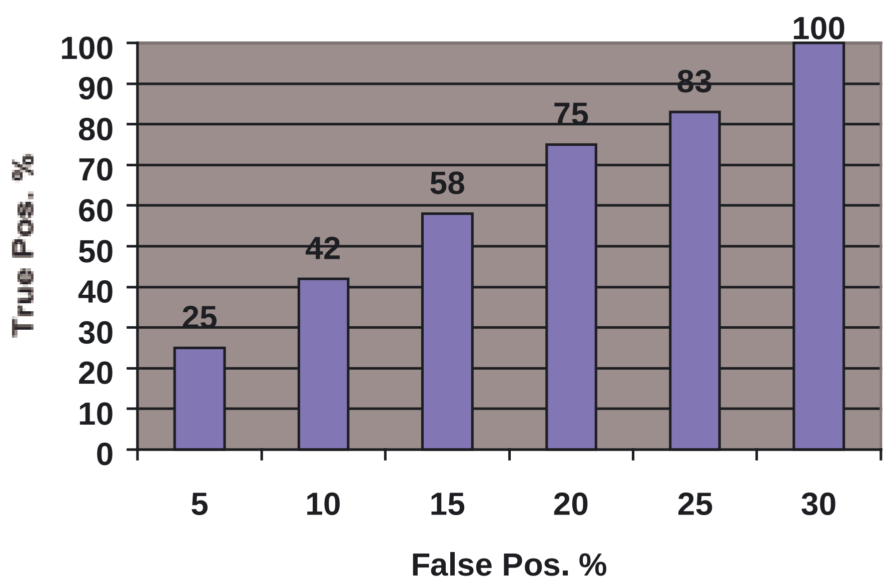


Figure 4-1: A demonstration of the relationship between the minimum necessary true positive rate and the allowable false positive rate for an environmental control task. The control task has 5 controls, of which 3 need to be corrected when a mistake is made.

In the experiments done by both Polikoff et. al. [70] and Spencer et. al. [80], the user was required to make a choice every stimulus round. The stimulus that was considered to have the largest P3 peak was the chosen stimulus. For the environmental control experiment, the possibility of the user choosing none of the stimuli or more than one stimulus during a stimulus set was considered. When this possibility is taken into account, it is possible to reduce the number of errors caused during a round of poor or artifact affected responses. Considering this possibility also separates the true and false positive rates in the system. It becomes possible to increase the true positive rate without decreasing the false positive rate, since more than one false positive may occur during a stimulus round. In the reverse case, more than one true positive may occur during a stimulus round, creating the possibility for a speedier system response time.

The importance of the false positive rate in such a system cannot be overstated. In order to demonstrate how important the false positive rate is, say that there is a 5% false positive rate with 60 stimuli presented per minute. With this rate there will be an average of 3 errors in which something happens that the user did not choose. Of these errors, on average 1.8 of them will need to be corrected. In order to correct the errors while still reaching the necessary goal, the true positive P3 recognition rate needs to be at 25% or above. At a 10% error rate, there will be 6 errors. Of these errors, an average of 3.6 of them will need to be corrected, thus yielding a P3 recognition rate that needs to be at 42% or above. The full graph for the relationship between the true positive and false positive rates is shown in 4-1.

4.4 Signal Processing Algorithms

In order to do single trial analysis in a BCI, it is beneficial for the all signal preprocessing and recognition to be easily calculated without the need of special hardware. Three routines that meet this requirement are presented: peak picking, correlation, and a third algorithm that uses a robust Kalman filter to preprocess the data and correlation for recognition.

The simplest algorithm that is commonly used is called peak picking. The difference between a prototypical P3 and non-P3 (assuming not artifact) is the existence of a fairly large peak around 300 ms for the P3 that does not exist for the non-P3 trial. Thus, peak picking looks for trials with an amplitude difference greater than a specified voltage difference between the minimum and maximum voltage points within a specified time window. Peak picking has two parameters: the time window in which to look for the peak and the voltage difference threshold that is needed in order to declare the peak a P3 component. For recognition, the time window with the best results was between three and six hundred milliseconds. The voltage difference parameter was varied in experiments to yield the best result.

Peak picking offers the benefit of knowing when there is a P3 peak. As suggested by Donchin, there is not always a good peak in the appropriate time frame [22]. In addition, peak picking suffers from the drawback of responding poorly in the presence of spurious high frequency noise and artifacts. Since single trials were recognized, it was necessary to an 8 Hz low pass filter in order to reduce this higher frequency noise contamination. Eye movements will contaminate the activity in the more anterior electrodes with the effect generally lessening towards the more posterior electrodes. If eye movements aren't accounted for in a peak picking situation, the subject may be able to achieve the desired results by moving the eyes rather than by a neural event. One of the goals of looking at the peak picking algorithm was to find out exactly what kinds of effects varying degrees of artifact had on recognition.

A slightly more complex, but still easily calculated recognition algorithm is correlation. Correlation may be looked at as template matching when the correlation is performed between single trials and a template of what each kind of trial should look like. Single trials were correlated with the P3 and non-P3 averages from electrode site PZ for each subject using the following formula:

$$\rho_{\mathbf{x},\mathbf{y}} = \frac{\mathrm{cov}(\mathbf{x},\mathbf{y})}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}}$$

where **x** represents the data for a single EEG trial, **y** represents the P3 or non-P3 average, cov(**x**,**y**) is the covariance of **x** and **y**, and σ is the standard deviation of the appropriate signal. **x** and **y** were both vectors. Note that the covariance is a scalar, because the **x** and **y** vectors both represent multiple instances of a single variable. There are two parameters that may be set for this recognition technique: the minimum correlation value that indicates a P3 and the trial size. Each of these parameters were varied in the experiments below.

Correlation has the ability to use a template size that is bigger than the area around a P3 signal. In this way, even if there is not a large P3 peak, correlation still retains the possibility of giving correct recognition. Unfortunately, much like peak picking, correlation may respond to spurious noise and artifacts.

The third recognition algorithm is the most complex, but is also the most statistically robust. The robust Kalman filter framework formulated by Rao [71] was used for preprocessing and correlation for recognition. The Kalman filter assumes a linear model $\mathbf{x} = A\mathbf{s}$, where **x** is the observable output of a generative or measurement matrix A and an internal state vector **s** of Gaussian sources. The output may also have an additional noise component **n**, a Gaussian stochastic noise process with mean zero and a covariance matrix given by $\Sigma = E[\mathbf{n}^T]$, leading to the model expression:

$$\mathbf{x} = A\mathbf{s} + \mathbf{n}$$

s may be learned and in order to find the most optimal value of **s**, a weighted least-squares criterion is formulated:

$$J = (\mathbf{x} - A\mathbf{s})^T \Sigma^{-1} (\mathbf{x} - A\mathbf{s}) + (\mathbf{s} - \bar{\mathbf{s}})^T M^{-1} (\mathbf{s} - \bar{\mathbf{s}}) \qquad \text{Eq. 1}$$

where **s** follows a Gaussian distribution with mean $\bar{\mathbf{s}}$ and covariance *M*. Minimizing this criterion by setting $\partial J / \partial \mathbf{s} = 0$ and using the substitution $N = (A^T \Sigma^{-1} A + M^{-1})^{-1}$ yields the Kalman filter equation, which is basically equal to the old estimate plus the Kalman gain times the residual error.

$$\mathbf{s} = \bar{\mathbf{s}} + NA^T \Sigma^{-1} (\mathbf{x} - A\bar{\mathbf{s}}) \qquad \text{Eq. 2}$$

In an analogous manner, the measurement matrix *A* may be estimated (learned) if one assumes the physical relationships encoded by the measurement matrix are relatively stable. The learning rule for the measurement matrix may be derived in a manner similar to the rule for the internal state vector. The standard derivation of the Kalman filter minimizes Equation 1, but does not say how to obtain Σ. A common choice is to use a constant matrix or even a scalar, as is discussed by Rao [71]. For instance, the experiments in this section use the number 0.6. The weakness of this approach is that it is susceptible to outliers. Thus, a statistically robust version of the Kalman filter is used.

Robustness comes from ignoring parts of the EEG signal that fall outside a standard deviation of 1.0 from the signals on which the Kalman filter was trained. Thus, a robust filter should treat sudden noisy activity as an *occlusion* of the EEG signal. As stated by Rao [71], the problem of outliers can be handled using a procedure such as Maximum likelihood type estimation, which involved minimizing a function of the form:

$$J' = \sum_{i=1}^{n} \vartheta(\mathbf{x}^i - A^i \mathbf{s})$$

where ϑ is taken to be a less rapidly increasing function than the square. This ensures that outliers do not influence the optimization of J', thus the outliers are rejected. The following weighted least squares criterion is a special case:

$$J' = (\mathbf{x} - A\mathbf{s})^T S (\mathbf{x} - A\mathbf{s})$$

where S is a diagonal matrix whose diagonal entries $S^{i,i}$ vary according to the corresponding data residual $\mathbf{x}^i - A^i\mathbf{s}$. A simple but attractive choice for these weights is the non-linear function given by:

$$S^{i,i} = \min\{1, c/(\mathbf{x}^i - A^i\mathbf{s})^2\}$$

where c is a threshold parameter that can be modulated according to the application at hand. To understand the behavior of this function, note that S effectively clips the *ith* summand in *J'* to a constant value c wherever the *ith* squared residual $(\mathbf{x}^i - A^i\mathbf{s})^2$ exceeds the threshold *c*; otherwise, the summand is set equal to the squared residual.

By substituting $\Sigma^{-1} = S$ in the optimization function of Equation 1, we can rederive the Kalman filter update equation. The resulting equation is:

$$\mathbf{s} = \bar{\mathbf{s}} + NA^T G(\mathbf{x} - A\bar{\mathbf{s}})$$

where G is an nxn matrix whose diagonal entries at time instant t are given by:

$$G^{i,i} = \begin{cases} \underline{0 \text{ if } (\mathbf{x}^i - A^i\mathbf{s})^2 > c(t)} \\ 1 \,\text{otherwise} \end{cases}$$

G can be regarded as the gating matrix, which determines the gain on various components of the residual error. By effectively filtering out the high residuals, *G* allows the Kalman filter to ignore the corresponding outliers in the input **x**, thereby enabling it to robustly estimate the state **s**.

In order to recognize the P3 component, the responses of individual trials were correlated with the responses obtained from the P3 and non-P3 average signals. Ordinarily, the type of signal best correlating with the individual trial was the recognition type. An exception to this occurred when the correlation with the P3 response was below a threshold set for each individual in order to obtain a false positive rate of 10%. In this case, the highest correlation could be with the P3 response, but the trial would still be declared a non-P3, because of overall poor correlation values.

In our experiments, both the internal state matrix **s** and the measurement matrix **A** were learned by training them on the average P3 and non-P3 signals for each individual (as $p \times 1$ input vectors where *p* equals 512). The signal is measured from the start of the trial, which is known since it is triggered by the software. As mentioned by Rao [71], a decay term is often needed in order to avoid overfitting the data set. The decay used was equal to 0.3. After training, the signal estimate for each epoch is correlated with the P3 and non-P3 signal estimates. The higher correlation indicates the closest match, but it is possible that the single trial does not match any of the training signals well. In this case, it is often best to declare the trial a non-P3 rather than to allow more false positives in recognition. A threshold was used in order to cut down on the number of false positives. As this threshold is varied, an ROC curve such as the one in 4-2 may be obtained.

4.5 Recognition Results Using Low Artifact Data

Eight subjects' data, from the experiment in Chapter 6, was used in order to calculate a base recognition rate on low artifact data. Low artifact data was defined as data where the maximum voltage difference in the vertical eye channels fell equal to or below 50 microvolts. While this definition may not remove all artifacts, it was designed to remove any large eye or muscle movements, both of which are prominent causes of artifact in P3 component recordings.

Prior to any recognition, it was necessary to find the best trial size to use for correlation and the routine using the robust Kalman filter. The best epoch duration was determined to be around 1.5 seconds of data following the stimulus trigger. Since the experimental trial length is 1 second, this size overlaps the next trial period, giving the recognition algorithm information from the next trial.

In the experiment, the same stimulus is never presented twice in a row. This means that when considering a conglomerate of information from two trials there are three possibilities:

5. The first trial is a goal trial that produces a P3 signal and the next trial is an irrelevant (non-P3) trial.
6. The first trial is an irrelevant (non-P3) trial and the second trial is a goal trial that produces a P3.
7. The first trial is a non-P3 trial and the second trial is another non-P3 trial.

Only one of these conditions indicates that the current trial is a P3 trial. The information from the next trial should prove helpful in reducing false positives since a P3 trial should never occur twice in a row. In general, this is what occurred with false positive decreases ranging from 3-10% with the use of data from the next trial.

Using data from two successive trials is not the same as averaging two stimulus trials. When averaging stimulus trials, one must wait for the same stimulus to occur twice before averaging and most likely when stimuli are randomly presented, the same stimulus will not occur twice in a row. When using a conglomerate of data from two trials, one needs only to wait for the second trial to occur. Thus, in an experiment with five controllable items and a stimulus presentation rate of 1 second, the user must wait an average of 5 seconds in order to average two stimulus trials. The same user must only wait 2 seconds if using data from the current trial and all of the next trial. The user would only have to wait 1.5 seconds if only 0.5 seconds from the next data trial were used.

After determining the best overall trial sizes, ROC curves were constructed for each subject over a range of values for the final parameter in each algorithm. An example ROC for subject 1 is shown in 4-2. As expected, the true positive probability is low when the false positive probability is low and the true positive probability is high when the false positive probability is high. Correlation demonstrates the best overall performance with an ROC curve that gains quicker in the true positive direction with the least gain in the false positive direction.

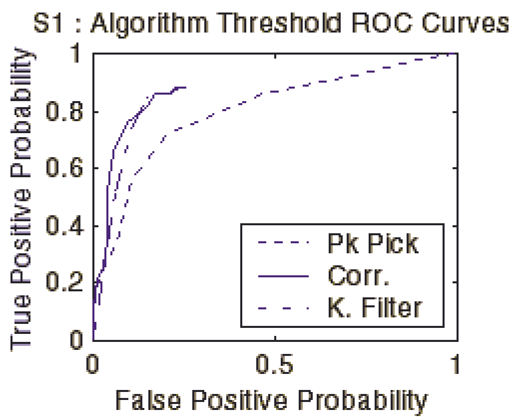


Figure 4-2: The ROC curves for a variety of parameter thresholds for Subject 1. Thresholds for both the robust Kalman filter and correlation algorithms run from 0 to 1.0, and so have a data range that is more limited than the range of peak picking, where the threshold parameter is equal to the minimum peak difference that is declared as a P3 peak. ROC curves for other subjects looked similar and may be viewed in Appendix A.

The average signals for true positives, false positives, false negatives, and true negatives at a 10% false positive rate were calculated for all subjects and are shown for subject 1 in Figure 4-3. In this experiment, true positives represented trials where the user was trying to pick a stimulus and succeeded due to correct recognition. False positives occurred when the user wasn't trying to pick an item, but the recognition algorithm declared it a P3 trial anyway. False negatives were expected P3 trials that the algorithm missed and true negatives were trials where the recognition algorithm did not detect a P3 and the user was not trying to produce a P3.

The averages of the four kinds of recognition trials demonstrate some of the difficulties of single trial recognition. As expected, the true positive averages all look like stereotypical P3s and the true negative averages lack this peak. The peak picking averages suggest that even though high frequency noise is filtered out, there are still large enough voltage swings in the trial to fool the peak picking algorithm. These voltage swings are not consistent among trials and so a P3-like peak does not occur in the false positive peak picking average. A P3-like peak does occur for the false negative average. This indicates that some small peaks were incorrectly classified by the peak picking algorithm. Peaks occurring slightly before the peak window would have been similarly penalized with incorrect recognition.

The average eye movement voltage is smaller than the average signal at site PZ. Some subjects show larger eye movement voltages (see Appendix A). Only subject 6 has an eye movement average that is more than the voltage at site PZ for trials recognized as true positives. Even in this case, it may be noted that the signals at site PZ are not very similar to the eye movement averages, suggesting that the slightly larger voltages from the vertical eye movement electrodes is not greatly affecting the electrode at site PZ.

Consistent P3-like peaks appear in the false positive averages for both correlation and the routine using the robust Kalman filter. In the case of correlation, the peak is more prominent than the slight peak in the false negative average. These results strengthen the suggestion made by Donchin that not all expected P3s appear to actually be a P3 [22]. In addition, it looks as though some expected negatives might actually be P3 signals. Both results are consistent with subjective comments made by several subjects that they were sometimes losing focus during the experiment or that were occasionally distracted by nearby stimuli. It would be expected that locked-in patients dependent on the interface for communication would probably be more motivated, but due to other factors such as medication and fatigue, these users might also have problems with a “boring” interface.

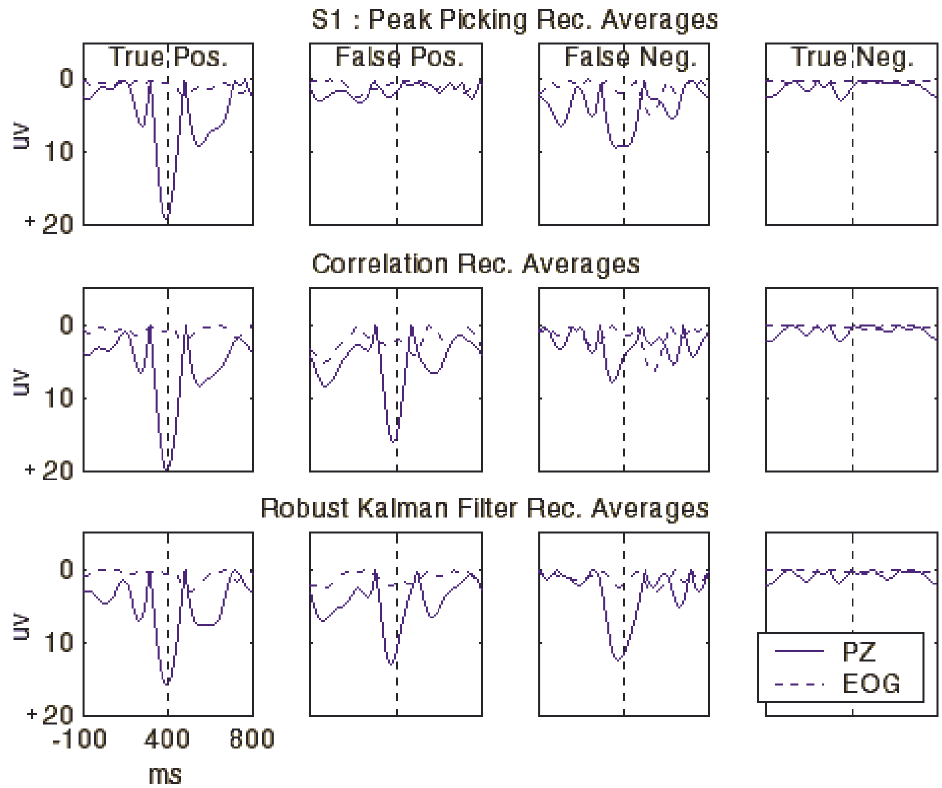


Figure 4-3: The solid line represents classified signal averages for subject 1 at site PZ, using peak picking, correlation, and the routine using the robust Kalman filter. The dotted line shows average vertical eye movement. True positives occurred when expected P3 trials were correctly recognized and false positives were trials incorrectly classified as P3's. False negatives occurred when the algorithm failed to detect an expected P3 and true negatives occurred when the algorithm correctly classified non-P3 trials. For peak picking, this subject had an average of 3 true positives, 7 false positives, 9 false negatives, and 62 true negatives. For both correlation and the routine using the robust Kalman filter, the true and false positive averages consisted of 7 trials. The false negative average consisted of 5 trials and the true negative average of 62 trials. All averages were filtered at 30 Hz for display purposes.

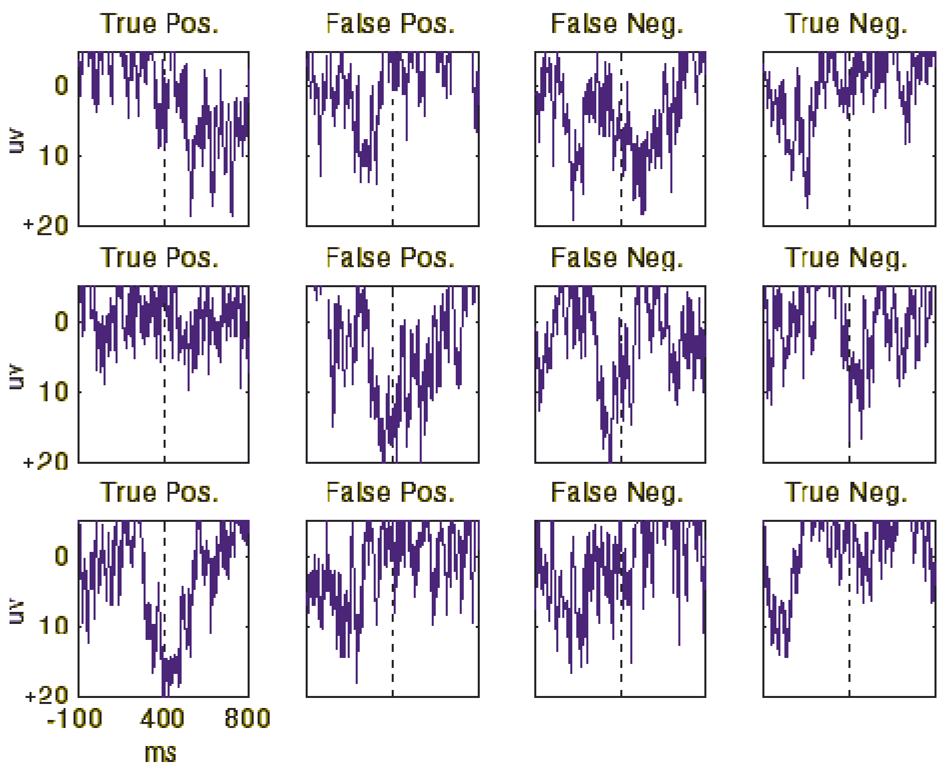


Figure 4-4: Examples of single trials from subject 1 and how they were recognized by the peak picking algorithm. The data epochs appear noisy because they have not been filtered down to 30 Hz for display. When looking at the peaks in the false negative category, keep in mind that the rise or fall of the peak must fit within a predetermined time window for the P3 signal. When the EEG signal rises slightly before the P3 window, the true peak difference is not calculated and thus the P3 peak is not detected.

The difficulties in recognizing single trial P3's are especially apparent from the single trials shown in Figure 4-4. Only one of the true positive trials looks like a stereotypical P3. Some of this is due to natural variation in the waveform morphology while other variations are unexplained. The second false positive trial looks like a stereotypical P3 and the recognition algorithm would be expected to declare it a P3 since that's what the signal appears to be.

A comparison of recognition results for all subjects, at a false positive recognition percentage of 10%, is shown in Figure 4-5. Peak picking performed the worst. Correlation performed the best in all cases with the exception of subjects 4 and 7 where correlation performed similar to the routine using the robust Kalman filter.

These results differ from the results obtained in Chapter 5, where the routine using the robust Kalman filter outperformed correlation. This may be due to two factors:

1. The data obtained in the VR driving environment is noisier than the data used in this experiment. The routine using the robust Kalman filter should perform best under artifact conditions because of its robustness.
2. This experiment uses a different epoch size from the experiment in Chapter 5.

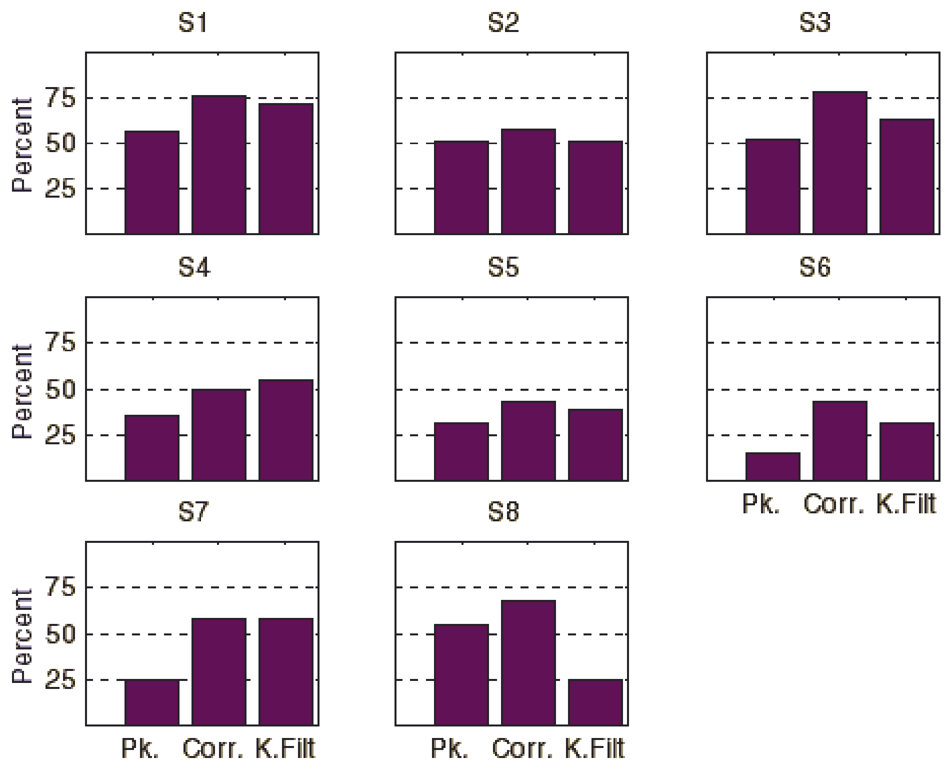


Figure 4-5: The recognition rates for all subjects under the three different algorithms when the false positive rate is equal to 10%.

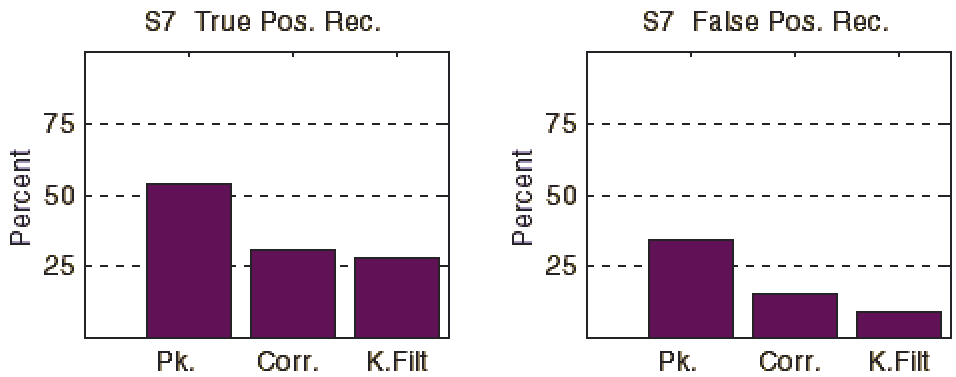


Figure 4-6: True positive and false positive recognition percentages on artifact contaminated data over all three algorithms.

While peak picking performs worse than the other two algorithms, in all but two subjects peak picking is above the minimal recognition rate of 42% that is needed in order to perform the environmental control task. Most algorithm results are well above this minimum and are similar to the recognition rates obtained by Spencer et. al. at 56% for on-line recognition [80] and Polikoff et. al. at 50% for off-line recognition [70].

4.6 Signal Processing Results on Artifact Contaminated Data

Data from the subject with the most contaminated trials were used in order to calculate recognition on artifact contaminated data. Artifact contamination was defined as data where the maximum voltage difference in the vertical eye channels was above 50 microvolts. While it is possible for uncontaminated single trial data to be above 50 microvolts, it was assumed that the larger the vertical eye channel voltage, the more eye movement artifacts there should be in the data. The results in this section use parameters from the data in the previous section, which yielded a 10% false positive rate.

The true and false positive recognition rates for all algorithms are shown in 4-6. True positives are those data trials where a recognized P3 occurs for a subject goal task (please see Chapter 6 for details). Comparing these plots with the recognition percentages from Figure 4-5 demonstrates that artifact contamination has a negative impact on the recognition rate for both correlation and the algorithm using the robust Kalman filter for preprocessing. The bulk of the errors for both of these routines exist as false negatives (an existing P3 signal is not detected) rather than false positives (recognizing the artifact as a P3 signal). Using these algorithms with artifact contaminated data impacted the true positive recognition rate without significantly changing the false positive rate. While the changes in recognition appear to be significant, they are not due to the low number of true positive trials that are used for non-artifact recognition.

For the peak picking algorithm, the opposite is true. The false positive rate for peak picking is over 30% (as opposed to 10% for non-artifact data). Similarly, a greater percentage of true positive trials are correctly recognized. This is probably due to the fact that artifact contamination tends to lead to larger voltage swings, thus leading to larger voltage differences in the P3 time window.

Overall averages for Subject 7 using the three different recognition algorithms are shown in Figure 4-7. It can be seen that all recognition categories have some form of artifact in the average. Even the true negatives show a very P3-like signal in the vertical eye channel average. The false negatives show artifacts at locations other than the area where the P3 is found. When the averages for site PZ are compared with the average vertical eye movement, it may be seen (especially in the true negative averages) that the signals at site PZ are influenced by vertical eye movements.

While this data shows a general downward trend in signal recognition due to artifact contamination, did this trend continue as the size of the eye movement channel voltage was increased? Results from this part of the experiment are shown in Figure 4-8.

Peak picking shows a slowly rising false positive rate as artifacts become larger. Correlation remains the most constant with a false positive rate that slowly climbs a little and then falls a little. The consistence of the correlation algorithm may be due to the fact that when an artifact contributes a positive voltage signal to the signal of interest, it usually also contributes a negative signal. If these two signals are about the same in voltage, their affects may cancel out. The algorithm using the robust Kalman filter is the most interesting, as its total recognition rate slowly rises as the amount of artifact increases. This increase suggests a trend ($p < 0.1$) that may be attributed to the robust nature of the Kalman filter, as the only difference between this routine and the correlation algorithm is the use of the robust Kalman filter in preprocessing the data. Since the algorithm treats signals outside of 1 standard deviation from the training samples as occlusions to be ignored, the Kalman filter is simply ignoring the portions of each trial with large artifacts. Results from this section indicate that both correlation and the routine using the robust Kalman filter withstand artifact contamination well. As may be seen from Figure 4-9, the single trials from site PZ are only slightly different from the single trials shown in 4-4. This may be due to location of the electrode site, as site PZ is fairly distant from the eyes.

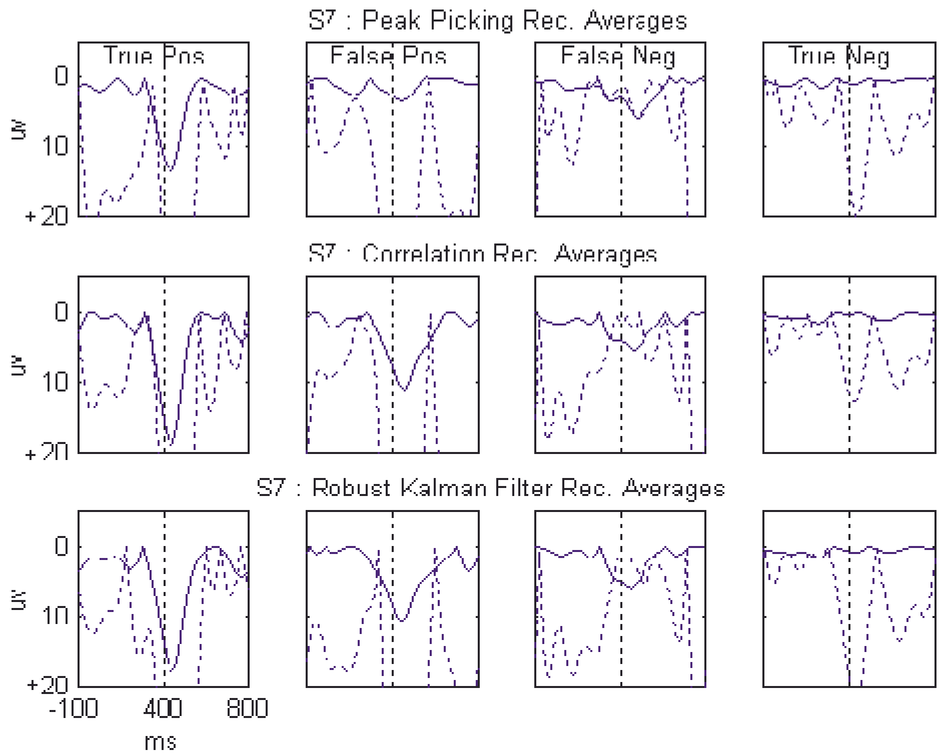


Figure 4-7: Signal averages for Subject 7 for peak picking, correlation, and the routine using the robust Kalman filter. All trials used have a maximum vertical eye movement voltage larger than 50 microvolts and are considered to be artifact contaminated trials. The solid line represents the average signal from site PZ and the dotted line represents the vertical eye movement average. All averages are filtered at 30 Hz for display purposes.

4.7 Signal Processing Results on Artifacts

Artifacts are signals recorded from the scalp, which do not come from cerebral activity. Numerous types of artifacts exist and some, such as eye blinks cause large amplitude deflections that obscure cerebral activity while others almost mimic the underlying EEG activity and are difficult to distinguish. The previous section suggests that eye movement artifacts may have a negative impact on signal recognition. Many different kinds of artifacts exist and this section considers how several kinds of artifacts including heavy breathing, chewing, foot movement, forehead movement, horizontal eye movement, jaw movement, talking, and vertical eye movements may contribute to false positive P3 recognition.

Sample signals of specific artifacts are shown in Figures 4-10 and 4-11. Further samples and plots may be found for all artifacts in Appendix B. Vertical eye movements have the largest effect on other electrode sites and some artifacts such as heavy breathing perturb other sites very little. Recordings for all artifacts was done using a single subject. Electrode impedances were between 2 and 5 kOhms. The EEG signal was amplified using Grass amplifiers with an analog bandwidth from 0.1 to 100 Hz. Signals were then digitized at a rate of 500 Hz and stored to a computer. For all artifacts not involving eye movement, the subject was asked to keep his/her eyes closed. Occasionally , artifacts other than the primary artifact of interest were recorded. As an example, during heavy breathing the subject also tended to move his/her shoulders.

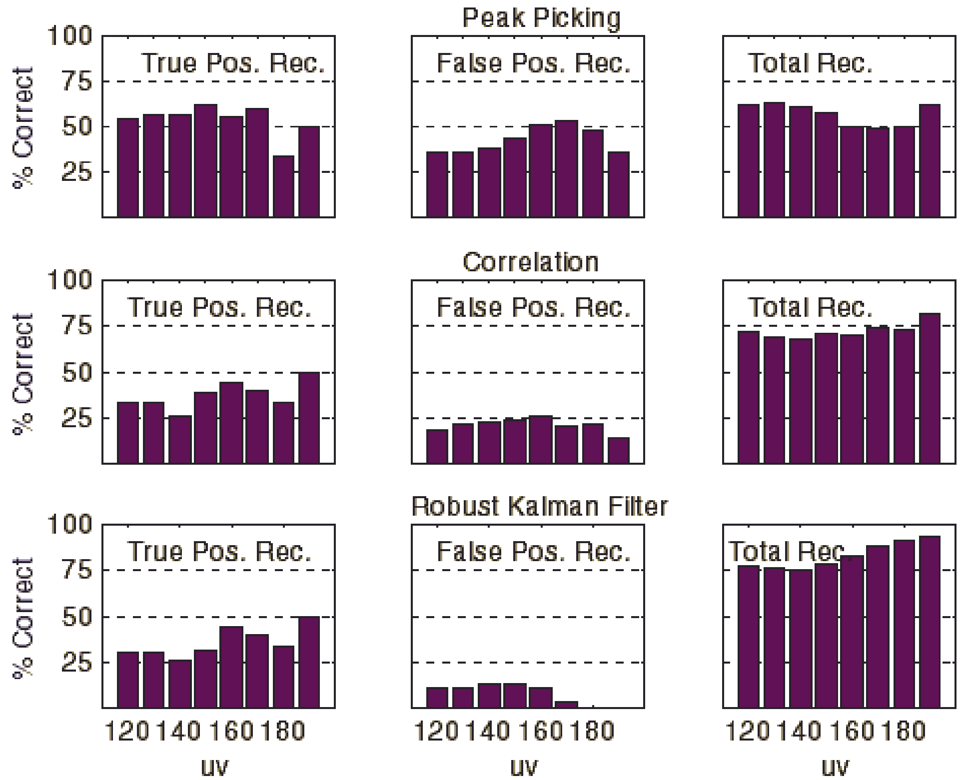


Figure 4-8: Recognition rates obtained from increasing the size of the vertical eye movement voltage from a trial maximum of 120 microvolts or above to a trial maximum of 190 microvolts or above. All data is from subject 7. Eye movement voltages from 60 to 120 microvolts are not shown because the recognition rates are similar to the rates discussed for 50 microvolts.

Trials were created from the artifact data by using a moving window who's start successively moved 400 milliseconds to create each new trial. As may be seen from Appendix B, this moving time window created an average of more trials than either of the two previous experiments. All trials were assumed not to contain P3 data and thus only true negative and false positive classifications were possible. Recognition for all three algorithms is shown in Figure 4-12. The peak picking algorithm performed the worst, with a low recognition of 74% (for forehead movement), a high recognition of 96% (for jaw clenching), and an average recognition of 88%. This result is due to the fact that large artifacts often create larger voltage differences in the P3 time window, causing more false positives.

The correlation routine and the routine using the robust Kalman filter performed in a similar manner with all results for both algorithms above 90% correct recognition. The correlation algorithm had the highest recognition rates for foot movement and jaw clenching (both 97%), the lowest recognition rate for vertical eye movements (93%), and an average recognition rate of 95%. The routine using the robust Kalman filter had the highest recognition rate for heavy breathing (98%), the lowest recognition rate for horizontal and vertical eye movements (both at 92%), and an average recognition rate of 94%.

Surprisingly, the results from the correlation algorithm and routine using the robust Kalman filter were better than the results on the original data, which had a false positive rate held at 10%. This difference could be due to the larger number of data trials used in this experiment. The difference also provides more support that the two algorithms perform well in the face of noise.

While most of the false positive averages show large artifacts in the vertical eye movement channel, a few show a distinct lack of artifact around the area of the P3. An example of this may be seen in the false positive average for the routine using the robust Kalman filter during heavy breathing (Figure 4-13). There is a slight peak around 400ms that does not show up at all in the vertical eye movement channel. Could this average contain a real P3? It is possible, but cannot be proven.

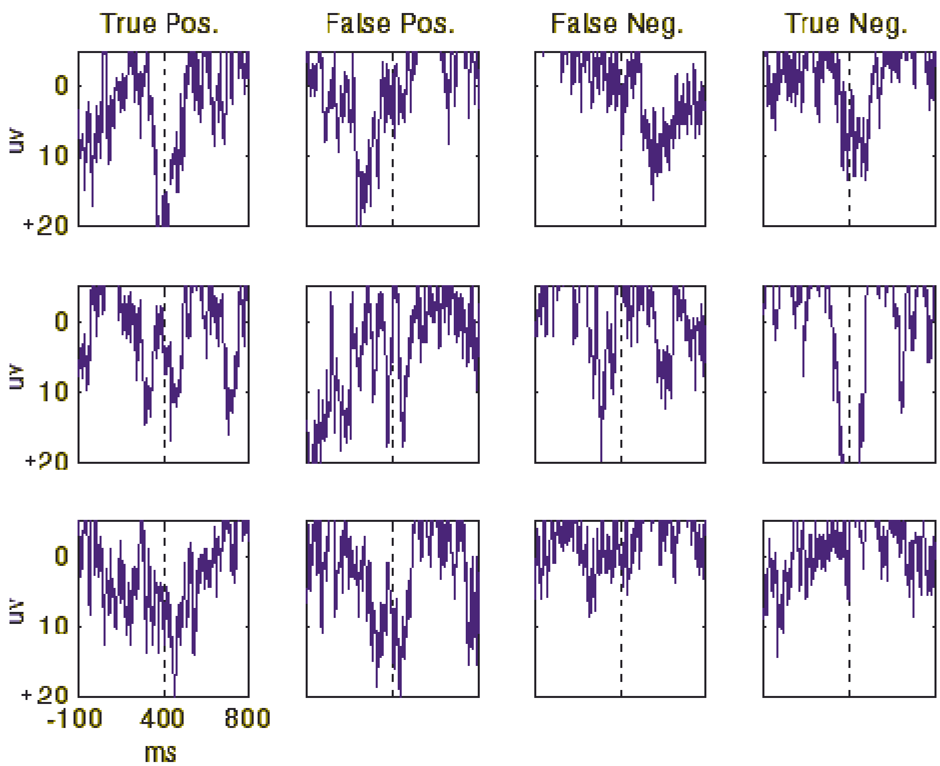


Figure 4-9: Single trials from subject 7 as recognized by the peak picking algorithm. These trials are from data with a maximum vertical eye movement voltage of over 50 microvolts.

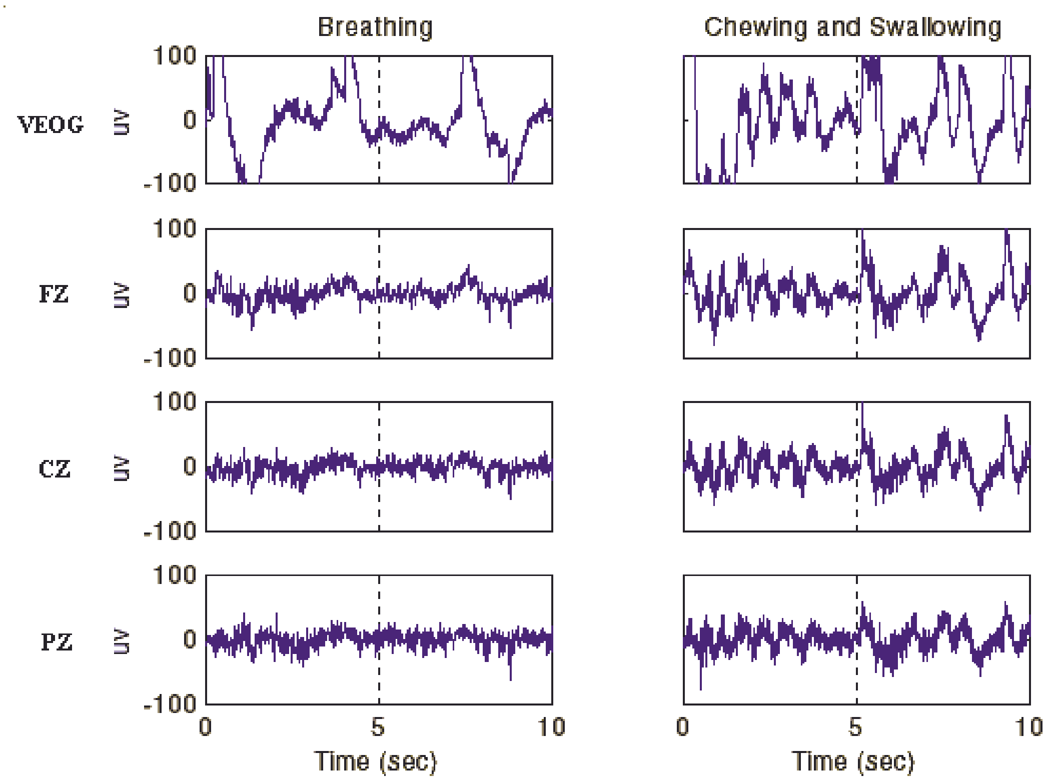


Figure 4-10: Ten seconds of artifactual raw EEG data demonstrating how heavy breathing and chewing/swallowing may affect other electrode sites.

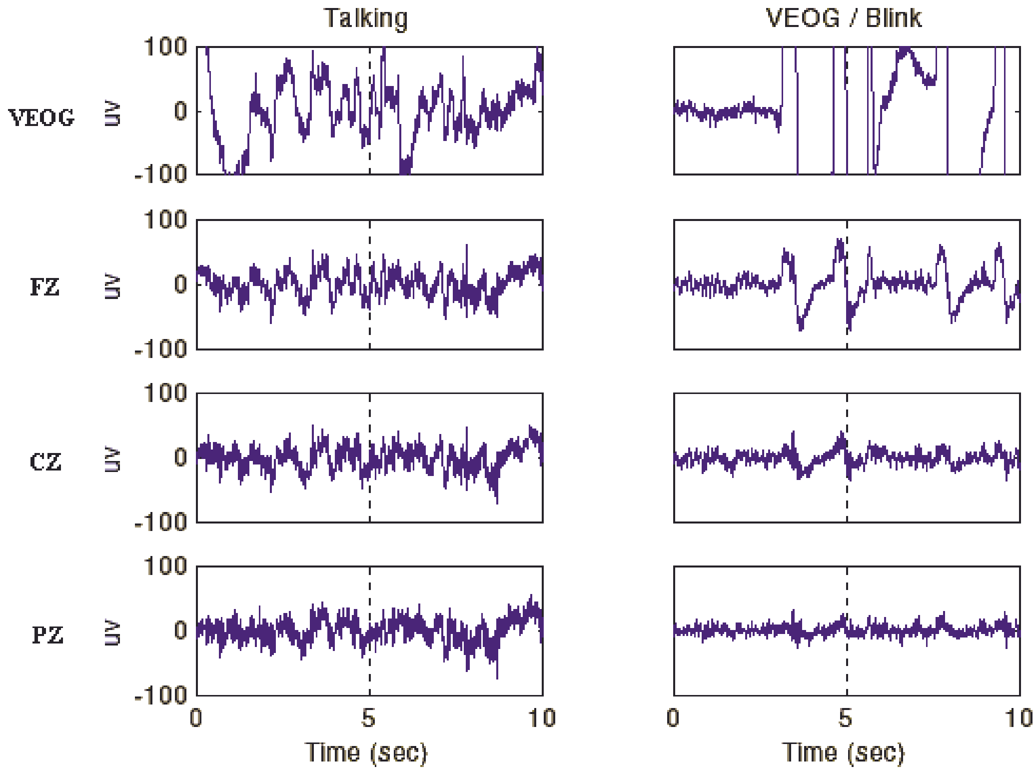


Figure 4-11: Ten seconds of artifactual raw EEG data showing how talking, vertical eye movements, and blinking may affect other electrode sites.

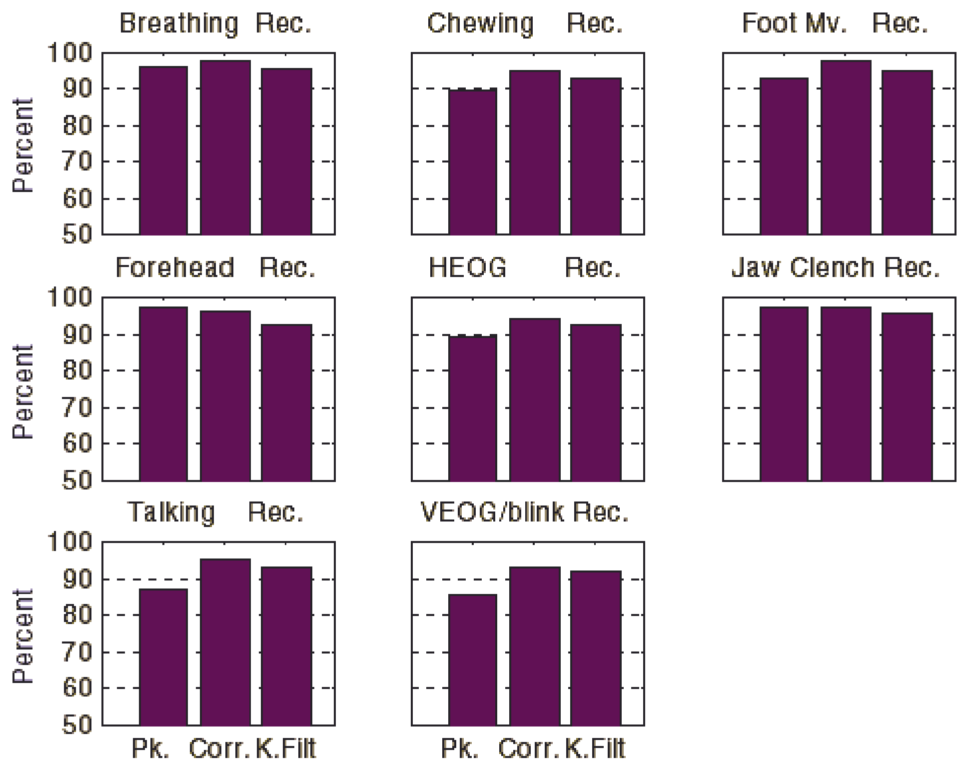


Figure 4-12: Recognition rates for the three algorithms over different sets of artifact data.

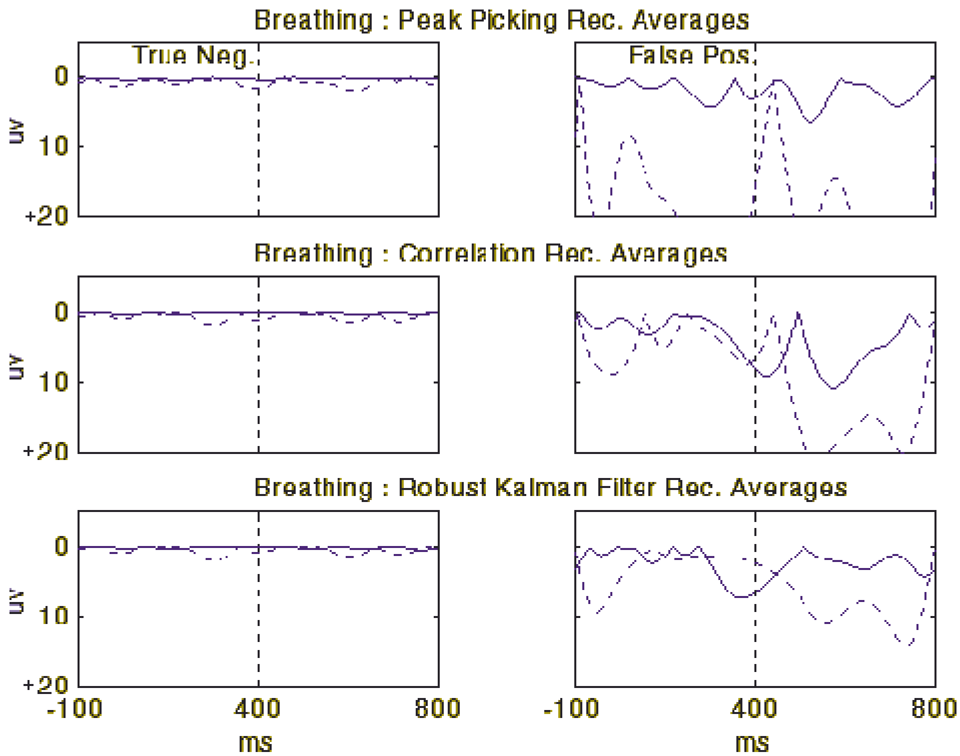


Figure 4-13: Recognition category averages for all three algorithms during the presence of heavy breathing artifacts.

5 Recording Evoked Potentials in a Virtual Environment

Virtual reality (VR) provides immersive and controllable experimental environments. It expands the bounds of possible evoked potential (EP) experiments by providing complex, dynamic environments in order to study cognition without sacrificing environmental control [6]. VR also serves as a safe dynamic testbed for brain-computer interface (BCI) research. However, there has been some concern about detecting EP signals in a complex VR environment. Signal recognition of evoked potentials remains a difficult and unsolved problem. Recognition difficulties lie in both the low signal-to-noise ratio available from an EEG signal as well as from possible outside contamination by artifacts. Individuals immersed in VR may move more, thereby making movement artifacts more common. There have also been concern regarding wearing a head-mounted display over electrodes and with possible immersive display interference. This chapter demonstrates that EPs can be recorded reliably in a virtual driving environment.

This dissertation shows that requiring subjects to stop or go at virtual traffic lights elicits a P3 component in an evoked potential. For a more in depth description of the P3 component of the EP, please see Section 1.4. In 1964, Walter reported an event related potential (ERP) event that showed a slowly increasing negative shift preceding an expected stimulus [87]. This ERP has since been named the “contingent negative variation” (CNV). A virtual traffic light indicating to a driver in the virtual world driver that he /she should reduce speed in anticipation of having to stop, might be expected to produce the CNV. This condition existed for the “slow down” lights in the virtual driving environment.

In order to determine the plausibility of single trial on-line P3 recognition in the VR driving environment, the P3 component was recognized at stoplights where subjects were required to stop and the absence of this signal was recognized at slow down lights. Binary recognition represents the most basic ability that would be necessary if a BCI were to be used in a virtual environment. Recognition results obtained after subjects freely drove a car simulator in a virtual town indicated that using a robust Kalman filter would enable the car simulator to be stopped at red lights with an average accuracy of 84.5% while results obtained when the simulator stop/go commands were actually controlled by the recognition algorithm were an average of 83% [5].

It may be noticed that the stoplight signals were classified as P3 or non-P3 after the system was told a light change had occurred. Thus, the signals were not detected. They were classified. Whenever a stoplight changed, a number was sent through a serial port to the data acquisition/analysis program. The number told the program that a signal had changed and that the recognition algorithm needed to be run in order to classify the signal type. One of the benefits of controlling items in a virtual environment, is that stimulus triggers (in this case stoplights) are easily detected. This may or may not be the case in a real environment. While the recognition results were not good enough to drive a real car in any environment, they do show the feasibility of controlling items in a virtual environment.

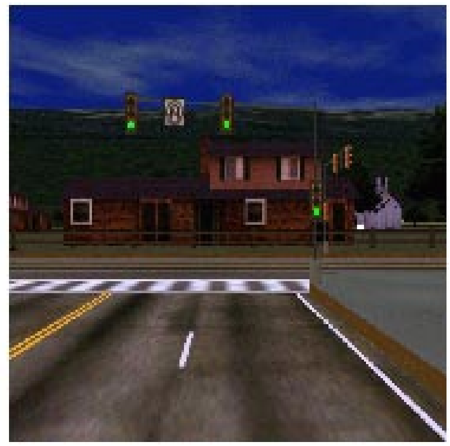


Figure 5-1: (Left) An individual demonstrates driving in the modified go cart while wearing an electrode cap and head-mounted display. (Right) A typical stoplight scene in the virtual environment.

5.1 Experimental Setup

The virtual reality interface was rendered on an Onyx Silicon Graphics machine with 4 processors and an Infinite Reality Graphics Engine. The environment was presented to the subject through a head-mounted display (HMD). Since electrical signals may easily interfere with EEG recordings during an experiment, the effects of wearing a VR4 HMD were tested and it was discovered that the noise levels inside of the VR helmet were comparable to noise levels while watching a laptop computer screen [4].

All subjects used a modified go cart in order to control the virtual car. A typical scene from the driving environment is shown in 5-1. Go cart driving was chosen, because it is more like a “natural” driving task than driving and stopping with a mouse. While this choice may cause a more artifacts in the signal collection (due to turning the steering wheel and braking), most of the actual artifacts in the data were discovered to be due to eye movements. Epochs with electro-oculographic (EOG) artifacts were removed for all off-line analysis in the following experiments. For on-line experiments, the EOG artifacts were regressed out of the signals of interest using the algorithm by Semlitsch [76].

Since the goal of the experiment was to examine the feasibility of detecting evoked potentials in a virtual environment for a brain-computer interface, a trigger signal containing information about the color of the light was sent over a serial port interface to the EEG acquisition system whenever a light changed. This allowed the acquisition system to sync information about the color of the light with acquired EEG data for later off-line analysis. The trigger also initiated the on-line recognition algorithm, so that EEG signals could be recognized in a timely fashion and on-line responses given. An epoch size from -100 ms to 1 sec was specified and the time course of a single epoch is shown in Figure 5-2 . Data were recorded continuously for future off-line analysis.

Eight electrode sites were arranged on each of the heads of five subjects with a linked mastoid reference. Sites FZ, CZ, CPZ, PZ, P3, P4, as well as lower and upper vertical EOG channels were used (see Figure 2-1 for a diagram of sites). Electrode impedances were between 2 and 5 kOhms for all subjects. The EEG signal was amplified using Grass amplifiers with an analog bandwidth from 0.1 to 100 Hz. Signals were then digitized at a rate of 500 Hz and stored to a computer.

5.2 The Stoplight Experiments

Previous P3 research has concentrated primarily on static environments, such as the continuous performance task [72]. In the visual continuous performance task, static images are flashed on a screen and the subject is told to press a button when a rare stimulus occurs or to count the number of occurrences of a rare stimulus. The nature of the rare, task relevant stimulus yields a P3. As an example, given picture of red and yellow stoplights, a P3 should occur if the red picture is less frequent than the yellow and subjects are told to press a mouse button only when they see the red stoplight picture.

A similar response should occur in a VR driving world if red stoplights are infrequent events and subjects are told to stop their virtual cars at the red light. This differs from the visual continuous performance task in two important ways:

1. In the visual continuous performance task, subjects sit passively and respond to stimuli. In the driving task, subjects control when the stimuli appear by where they drive. While driving is at first random in the environment, subjects quickly learn where the lights are and thus have control over their stimulus presentation.
2. Since subjects are actively involved and fully immersed in the virtual world, they make more eye and head movements.

The first difference makes the VR environment a more natural experimental environment. The second difference means that subjects will likely create more data artifacts due to extra movement. These artifacts were handled by first manipulating the experimental environment to reduce movements where important stimulus events occurred. This meant that all stoplights were placed at the end of straight stretches of road in order to avoid the artifacts caused by turning a corner. The eye movement reduction technique described by Semlitsch [76] was used in order to subtract a combination of the remaining eye and head movement artifacts.

5.2.1 Ordinary Light Colors

Five subjects each were instructed to slow down on yellow lights, stop for red lights, and go for green lights. These are expected traffic light colors. Subjects ranged in age from 19 to 52 and most had no previous experiences in a virtual environment. A schematic of the experiment may be seen in 5-2 . Subjects were allowed to drive in the environment before the experiment in order to become used to driving in VR.

In the USA, yellow lights precede only a red light. So that the yellow light did not indicate the next light color, this scheme was changed so that a yellow light preceded both green and red lights. This paradigm allowed the experiment to indicate if an EP other than the P3 could be reliably obtained. Since subjects expected the light to either change to green or red after turning yellow, a CNV was expected to occur. All stoplights turned to yellow when subjects were further than 30 virtual meters away from them. When the subject drove closer than 30 meters, the light then turned either red or green with equal probability. The rest of the light sequence followed normal stoplights, with the red light turning to the green after 3 seconds and remaining green while in the subject’s visual range.

Grand averages were calculated over red, green, and yellow light trials (see 5-3a). Epochs affected by obvious artifacts were removed by hand in order to make sure that existing movements were not causing a P3-like signal. Results show that a P3 EP occurs for both red and green lights. Since subjects expect to encounter stoplights at intersections while driving, the locations of stoplights at ordinary intersections produced a P3 mainly related to task relevance. Averaging back from the green/red light triggers to the yellow light trigger shows the existence of a CNV starting at approximately 2 seconds before the light changes to red or green.

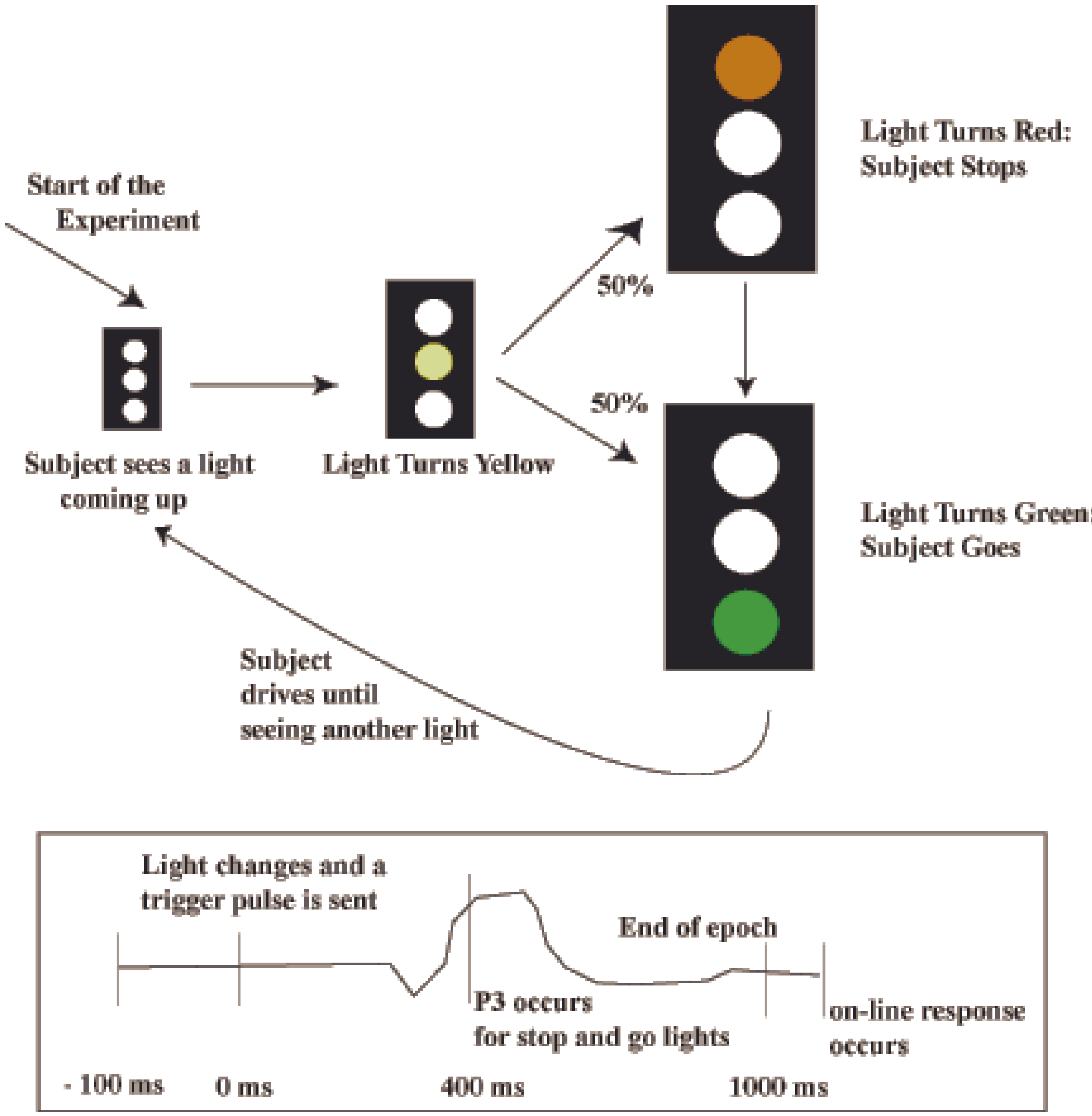


Figure 5-2 : The order of stoplight presentation and an example time course for a single trial in the ordinary light color experiment. In the alternative light color experiment, the presentation of lights changes, but they remain in the same physical location on the traffic light.

5.2.2 Alternative Light Colors

The P3 is related to task relevance and should not be related to light color, but color needed to be disambiguated as the source of the P3 in the experiment. Two subjects slowed down at green lights, stopped at yellow lights, and were instructed to keep going at red lights. In order to get used to this combination of colors, subjects were allowed to drive in the town for an extended period of time before the experiment.

The grand averages for each light color were calculated in the same manner as the averages for the ordinary light color experiment and are shown in 5-3b. As expected, a P3 signal existed for the stop condition and a CNV for the slow down condition. The go condition P3 was much noisier for these two subjects, although a slight positivity was noted at the appropriate time interval. Both subjects in the alternative light condition reported having difficulty learning the new light sequences.

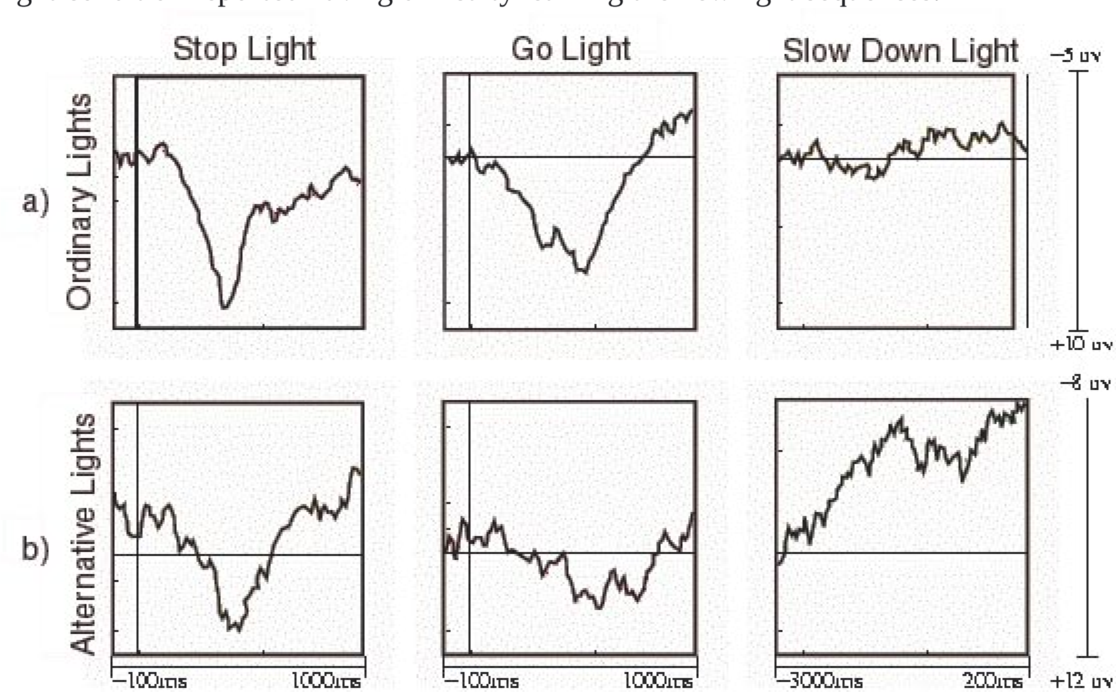


Figure 5-3: a) Grand averages for the red stop, green go, and yellow slow down lights. b) Grand averages for the yellow stop, red go, and green slow down lights. All slow down lights have been averaged back from the go/stop light trigger in order to show the existence of the CNV.

5.3 Signal Recognition of Single Epochs

While averages show the existence of the P3 component of the EP at stop lights and the absence of such at slow down lights, it was necessary to discover if the signal was clean enough for single trial recognition, as the quick feedback needed by a BCI depends on quick recognition. Three separate preprocessing methods were used: independent component analysis (ICA), a Kalman filter, and a robust Kalman filter. Classification was done using correlation with each of the preprocessing methods and by itself. This led to a total of four different signal processing algorithms.

Approximately, 90 yellow light and 45 red light trials from each subject were classified. A yellow light bias was allowed to enter recognition, because the yellow light represents an unimportant event in the environment. In a real BCI, unimportant events are likely to occur more than user-directed actions, making this bias justifiable.

All of the methods used data preprocessed with the method described in the previous section. The overall recognition results in Table 5-1 suggest that data preprocessed by both kinds of Kalman filter and ICA have a statistically significant advantage over correlation ($p<0.01$). The robust Kalman filter has a very small advantage over the regular Kalman filter and both Kalman filters have a slight advantage over ICA (not statistically significant).

In order to look at the reliability of the best algorithm two of the Subjects (S4 and S5) returned for another VR driving session. The results of this session using data preprocessed by the robust Kalman Filter trained on the first session are shown in 5-2. The recognition numbers for red and yellow lights between the two sessions were examined using correlation. Red light scores between the sessions correlated fairly highly - 0.82 for S4 and 0.69 for S5. The yellow light scores between sessions correlated poorly with both S4 and S5 at around -0.1. This indicates that the yellow light epochs tend to correlate poorly with each other due to the lack of a large feature such as the P3 to tie them together. Details of the comparisons are described below.

Table 5-1: Off-line data analysis recognition results for the 5 subjects in the normal light color stoplight task.

Subjects	Correlation % Correct			ICA % Correct			Kalman Filter/Robust Kalman Filter % Correct		
	Red	Yel.	Total	Red	Yel.	Total	Red	Yel.	Total
S1	81	51	64	76	77	77	54/55	85/86	76/77
S2	95	63	73	86	88	87	82/82	96/94	92/90
S3	89	56	66	72	87	82	61/74	85/85	77/81
S4	81	60	67	73	69	71	63/65	90/91	81/82
S5	63	66	65	65	79	74	78/78	92/92	87/87

Table 5-2: Results from two subjects returned on a different day to test the reliability of the previously trained robust Kalman filter. For this session, the recognition was done on-line with and the algorithm controlled the brakes of the virtual car.

Subjects	*Red Light % Correct*	*Yellow Light % Correct*	*Total % Correct*
S4	*73*	*90*	*85*
S5	*67*	*87*	*80*

In order to create a baseline from which to compare the performance of other algorithms, the correlation of all sample trials with the red and yellow light averages was calculated from the maximal electrode site for obtaining the P3 for each subject using the following formula:

$$\rho_{\mathbf{x},\mathbf{y}} = \frac{\mathrm{cov}(\mathbf{x},\mathbf{y})}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}}$$

where **x** represents a single EEG trial, **y** represents the red or yellow light average, cov(**x**,**y**) was the covariance of **x** and **y**, and σ was the standard deviation of the appropriate signal. **x** and **y** were both 1×500 vectors. Note that the covariance is a single value, because the **x** and **y** vectors both represent multiple instances of a single variable. If the highest correlation of a trial epoch with the red and yellow averages was greater than 0.0, then the signal was classified as that type of signal. If both signals correlated negatively or were equal to 0.0, the signal was counted as a yellow light signal. As can be seen in Table 5-1, the correct signal identification of red and yellow light epochs using this simple technique was quite high, although the correlations in general were poor with typical correlations around 0.25.

Since correlation was the baseline technique for recognition, all other algorithms were used in concert with correlation for classification. This means that ICA and the Kalman filter were used to separate the signal of interest from noise and other ongoing EEG activity while correlation was used to detect the signal. This gives a sense of how ICA and the Kalman filter could increase recognition through transforming the signals of interest. Recognition algorithms other than correlation could be used, but for an on-line recognition algorithm, correlation remains one of the fastest techniques available.

ICA has successfully been used in order to minimize artifacts in EEG data and has also proven useful in separating P3 component data from an averaged waveform [37], [86],[51]. Introduced by Comon [20], independent component analysis (ICA) approximates the factor analysis of multiple linearly mixed source signals through assuming that the sources are statistically independent from one another. Thus, it is assumed that *n* EEG data channels **x** are a linear combination of *n* statistically *independent* signals **s**:

$$\mathbf{x} = A\mathbf{s} \qquad \text{Eq. 2}$$

where **x** and **s** are $n\times1$ vectors and A is an $n\times n$ mixing matrix. An important difference from many algorithms is that ICA assumes that the EEG channels are the mixed sources. The algorithm processes data over channels rather than over a single trial across time. The eight electrode channels used in recording EEG data in the driving experiment meant that n was equal to 8.Given this source and mixed relationship, the goal of blind source separation is to recover the vector of sources ($\mathbf{s}$) when given only the n mixed signals $\mathbf{x}$ and with the knowledge that these sources are statistically independent from each other.

Statistically, this means that we would like to know the true probability distribution $P(\mathbf{x})$ from which the samples in our mixed signals have been drawn. While we cannot calculate this distribution, we can model it. Using the maximum-likelihood approach, the goal becomes to reduce the difference between our model of the probability density function (pdf) and the actual pdf. In order to achieve this using ICA, we must find a vector of parameters $\mathbf{w}$ that maximize the log-likelihood that a set of mixed signals $\mathbf{x}$ could have arisen from a random process in which the sources are linearly mixed. Discussed by Olshausen [59], this is formally equivalent to minimizing the Kullback-Leibler distance (G) between the actual joint probability of the signals $P^*(\mathbf{x})$ and our model of the joint probability $P(\mathbf{x} \mid \mathbf{w})$:

$$\begin{aligned} G(P^*(\mathbf{x}), P(\mathbf{x} \mid \mathbf{w})) &= \int P^*(\mathbf{x}) \log \frac{P^*(\mathbf{x})}{P(\mathbf{x} \mid \mathbf{w})} dx \\ &= H[P] - \int P^*(\mathbf{x}) \log P(\mathbf{x} \mid \mathbf{w}) dx \end{aligned}$$

This leaves us with the entropy of the fixed input distribution $P^*(\mathbf{x})$ minus the likelihood of $P^*(\mathbf{x})$ given $P(\mathbf{x} \mid \mathbf{w})$, and the $\mathbf{w}$ that minimizes G maximizes the likelihood. As we do not have access to the actual Kullback-Leibler distance G, we may still obtain an unbiased estimate of it by taking a sample x from $P^*(\mathbf{x})$,

$$G^*(P^*(\mathbf{x}), P(\mathbf{x} \mid \mathbf{w})) = H[P] - \log P(\mathbf{x} \mid \mathbf{w}) dx$$

In order to obtain a learning rule for the neural network architecture, we use stochastic gradient ascent:

$$\frac{dG^*}{d\mathbf{w}} = -(d / d\mathbf{w} \log P(\mathbf{x} \mid \mathbf{w})$$

Let W be an n-*by*-n matrix, and let $\mathbf{x} = W^{-1}\mathbf{s}$, where $\mathbf{s}$ is an n-dimensional vector of sources whose components s_j are drawn from n independent parameterized one-dimensional density functions $f_j(s_j \mid \mathbf{w}_j)$. The estimated density of $\mathbf{x}$ is denoted $P(\mathbf{x} \mid \mathbf{w})$ where $\mathbf{w}$ is a concatenation of the elements of W with the parameters $\mathbf{w}_1,\ldots,\mathbf{w}_n$ of the densities $f_1,\ldots, f_n$. Expanding $\log P(\mathbf{x} \mid \mathbf{w})$ from above, we obtain

$$\log P(\mathbf{x} \mid \mathbf{w}) = \log |W| + \sum_j \log f_j(s_j \mid \mathbf{w}_j)$$

This gives us the following formulas:

$$\frac{dG^*}{dW} = -W^{-T} - \left(\frac{f_j'(s_j \mid \mathbf{w}_j)}{f_j(s_j \mid \mathbf{w}_j)} \right)_j \mathbf{x}^T$$

$$\frac{dG^*}{d\mathbf{w}} = -\frac{f_j'(s_j \mid \mathbf{w}_j)}{f_j(s_j \mid \mathbf{w}_j)}$$

where $(\mathrm{expr(j)})_j$ denotes the column vector whose elements are $\mathrm{expr}(1),\ldots,\mathrm{expr}(n)$.

In the Bell and Sejnowski algorithm [10], $f_j(s_j \mid \mathbf{w}_j)$ is the derivative of the logistic function, $y = g(W\mathbf{x} + \mathbf{w}_0) = (1 + \exp^{-s})^{-1}$ and the parameter vector $\mathbf{w}_j$ holds the j^{th} component of a vector of bias terms giving us

$$\frac{dG^*}{dW} = -W^{-T} + (1 - 2y)\mathbf{x}^T$$

$$\frac{dG^*}{d\mathbf{w}} = 1 - 2y$$

The ICA Matlab package from Makeig et. al., which uses the Bell and Sejnowski algorithm [51] was used with default learning values for the experiment which follows. Other suggested density functions have been used, including one that takes the surrounding context of the signal into account [3] [62], but these have not been as highly used in processing EEG data. Since ICA uses higher order statistics in order to separate components, most ICA programs, including the ICA Matlab package used, remove first and second order statistics from the mixed data in order to train the algorithm. This is known as prewhitening the data and speeds convergence time [9].

The experiment used ICA in order to try and separate the background EEG signal from the P3 signal. After training the *W* matrix, the source channel showing the closest P3-like signal for the red light average data was chosen as the signal with which to correlate individual epochs. The trained *W* matrix was also used to find the sources of the yellow light average. The red and yellow light responses were then correlated with individual epoch trials in the manner of the correlation experiment. Training on both the average red light signal and the separate red light epochs was tried and it was found that training on the first 7 P3 epochs led to better classification in all cases. The training data consisted of the first 7 P3 epochs while the testing data consisted of the rest of the epochs. Results are shown in Table 5-1.

While the strong assumption of independent sources is what enables ICA to unmix various signals, this assumption may also cause problems when ICA is used. There is no reason to assume that the brain is made up of 8 individual and statistically independent sources (the number of sources assumed in the experiment). Since the number of mixed signals and sources need to be the same, increasing the number of independent sources would mean increasing the number of electrodes used to record the mixed signals. This is infeasible for large numbers of sources. Even if there were only 8 mixed sources, there is no reason to assume that all sources were active at the time of the experiment. EEG signals are known to be notoriously non-stationary over time as different regions of the brain become activated or inhibited. The assumption that all sources are active during training may lead to the noticeable result that ICA may "create" a P3-like signal from various statistics in different EEG channels when the signal does not in reality exist. This may have caused ICA to perform poorly in recognizing the yellow lights as they do not contain a P3 signal.

The third experiment used data preprocessed by the the Kalman filter framework formulated by Rao, and described in detail in Chapter 4. The Kalman filter assumes a linear model similar to the one of ICA in Eq. 2, but assumes the EEG output $\mathbf{x}$ over time is the observable output of a generative or measurement matrix A and an internal state vector $\mathbf{s}$ of *Gaussian* sources. This is different from ICA, which does not look at data over time, but only over channels.

In our experiments, both the internal state matrix $\mathbf{s}$ and the measurement matrix A were learned by training them on the average red light and yellow light signals (each a 500×1 input vector). The signal is measured from the start of the trial, which is known since it is triggered by the light change. In order to speed up training, a simplifying assumption that the Kalman gain was equal to 0.6 was made. The decay used was equal to 0.3. After training, the signal estimate for each epoch was correlated with the red and yellow light signal estimates.

The results show that the data preprocessed by the Kalman filter yields recognition results that are a little better than the results obtained by preprocessing with ICA. The Kalman filter technique may be viewed as template matching, which suggests that most epochs are relatively similar to the overall average. Correlations for this technique were much higher than the other techniques and red light correlations below a threshold ranging from 0.8 to 0.91 (depending on the subject) were assumed to be so poor that they were classified as yellow light matches.

The last experiment used a robust version of the previously described Kalman filter. Robustness comes from ignoring parts of the EEG signal that fall outside a standard deviation of 1.0 from the signals on which the Kalman filter was trained. Thus, a robust filter should treat sudden noisy activity as an *occlusion* of the EEG signal. As can be seen from the results in Table 5-1, the results did not differ significantly from the Kalman filter results, indicating that reducing eye movement signals from the ongoing EEG signal removed most of the large artifactual data.

5.4 Discussion

It has been shown that single trial EP signals obtained while subjects drive through a VR driving environment may be accurately classified as either P3 or non-P3. Our results suggest that building a brain computer interface using the P3 component of the EP, as in the P3 character recognition system proposed by Farwell and Donchin [25], would prove feasible in a dynamic VR environment.

Variations between different subject sessions may be due to slightly different electrode placement or impedance. Training on-line during the first few data epochs will help remove these factors in order to give a better indication of algorithm reliability.

Current results show that the slow down light epochs have a high variance, making these lights difficult to classify reliably. The P3 epochs are more reliable. In the context of a BCI, the unreliability of slow down (non-P3) lights may give an indication of how well the algorithm will perform when trying to recognize an intended action (such as a button press) versus a non-intended action. Current results indicate that an intended action should be easy to recognize, but that there will be problems with false positive recognition.

6 Usability and Control in a Virtual Apartment

Due to the difficulty of processing small and noisy EEG signals, little work has been done on making brain-computer interfaces usable. As an example of usability difficulties encountered by BCI users, consider the Thought Translation Device. It was the only means of communication for several locked-in patients and the communication rate was two characters per minute [12]. While the creators of the Thought Translation Device worked hard to make the device possible, only two out of five locked-in patients were able and willing to use the system. Usability issues, including how a specific application may affect the EEG signals it uses for control, are especially important for BCIs.

Developing usable BCIs is a challenge for system designers. The complexity of the hardware, real-time processing demands of the software, and small number of test users can leave little time for usability testing and tuning. This is changing. Moore and Kennedy overview human-computer interface and training issues for an implanted BCI [55]. The user moved a cursor by imagining movements in his left hand. This resulted in a communication rate of about three letters per minute. In work with brain injury patients, Cole et al. note that system developers' design processes must change [19]. They note: "...our work with brain injury patients has shown that patient-system performance is extremely sensitive to ... minor design parameters: furthermore, that the brain injury survivor needs to be viewed as a relatively sensitive component, while the computer system design needs to be the most flexible." In fact, about two thirds of user interface changes and three quarters of the functionality were requested by patients or clinicians. They conclude "it is clear that at least some ... changes would not have been suggested by those with systems expertise [developers] because those changes were either counter-intuitive or violated accepted guidelines."

Virtual reality may prove useful for training individuals to use a BCI, for providing complex and controllable experimental environments for those improving BCIs, and for motivational reasons [8]. Experiments were performed to analyze the robustness of the P3 signal over virtual and non-virtual environments. Subjects were able to control several items in the virtual apartment shown in Figure 6-1. They could turn a light on/off, a stereo (on/off), a television set (on/off), and could choose to say hi/bye to a graphics character. Control performance under three conditions was looked at: in an immersive virtual apartment while wearing a head-mounted display (VR condition), while looking at the same apartment scene on a computer monitor (MONITOR condition), and while looking at a virtual monitor in the head-mounted display (FIXED DISPLAY condition). The goal of looking at these conditions was to determine if there were performance or qualitative experience differences between the different conditions. The basic questions asked were:

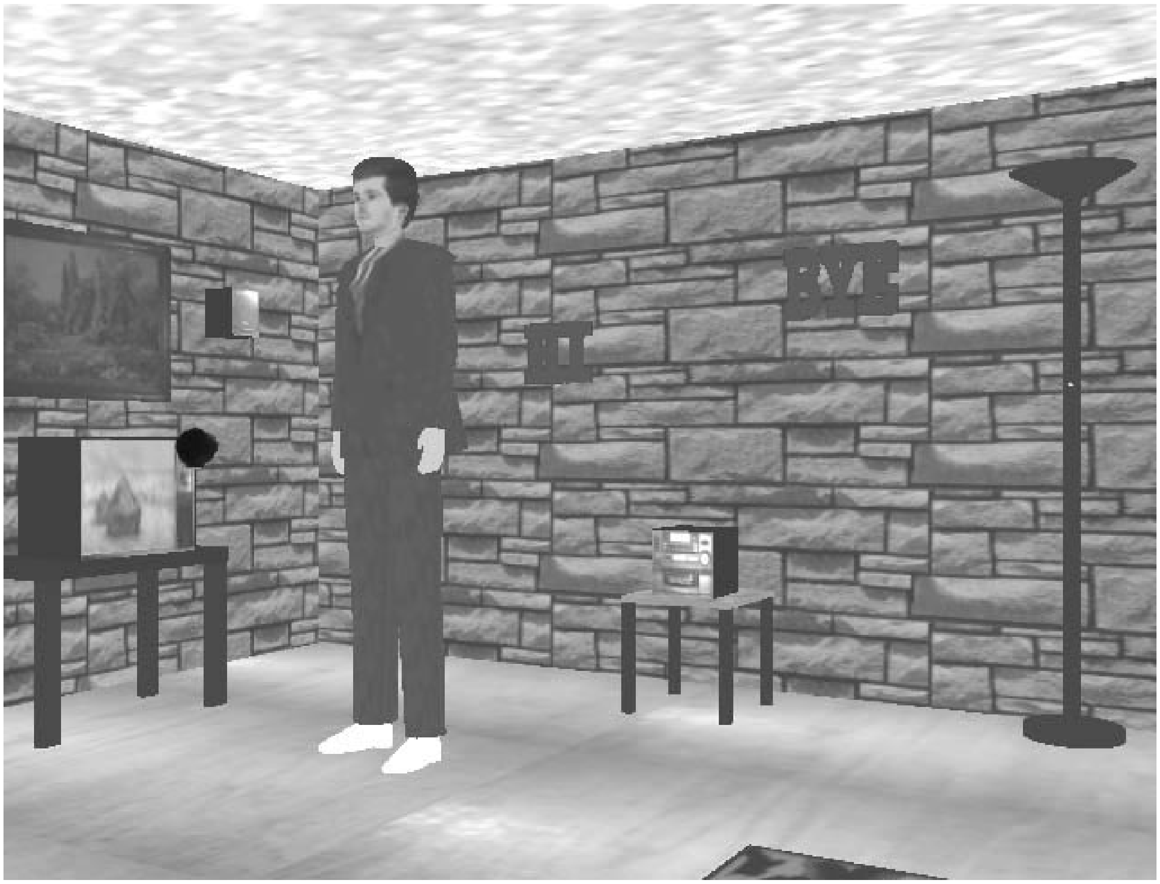


Figure 6-1: A sample scene from the virtual apartment. In this scene, a red light on the television set is blinking.

1. Are there significant differences between the evoked potentials obtained in the different conditions?
2. Are there significant performance differences between the different conditions?
3. Are there performance differences over time?
4. Are there qualitative experience differences between the different conditions?

The results obtained indicate there are no significant differences between evoked potentials in the three main conditions. This indicates the robustness of the P3 signal over different environments. There are slight performance differences among the conditions, with the only significant difference being between the MONITOR and FIXED DISPLAY conditions ($p < 0.05$). Subjects overall perform significantly better in the MONITOR condition. There are some differences in performance over time, but they have not been proven significant in this experiment.

Subjects' self-reported qualitative experiences did not necessarily match their objective performance. Six out of nine subjects liked the VR environment better while only one of these subjects performed the best in this environment. The possible ramifications of this, as well as future work, will be discussed.

6.1 Experimental Setup

The virtual reality interface was rendered on an Onyx Silicon Graphics machine with 4 processors and an Infinite Reality Graphics Engine. The environment was presented to the subject through a head-mounted display (HMD) during the FIXED DISPLAY and VR conditions. Since electrical signals may easily interfere with EEG recordings during an experiment, we tested the effects of wearing a VR4 HMD and discovered that the noise levels inside of the VR helmet were comparable to noise levels while watching a laptop screen [4].

All subjects sat in a chair to view the apartment environment. A typical scene from the apartment environment is shown in Figure 6-1. For on-line recognition and analysis, EOG artifacts were regressed out of the signals of interest using the algorithm by Semlitsch [76].

In order to trigger a P3 evoked potential response, a transparent bubble existed on every controllable item. This transparent bubble occasionally blinked red in order to cause a P3 response for the control goal. Stimuli were presented at a rate of one red flash per second. A numerical code for the particular bubble flash was then sent to the BCI backend over a serial port. An epoch size from -100 ms to 1500 msec was specified for a total epoch size of 1600 ms. The data were recorded continuously and saved to a file.

Seven electrode sites were arranged on the heads of nine subjects with a linked mastoid reference. Sites FZ, CZ, PZ, P3, P4, as well as a lower and an upper vertical electro-oculographic (EOG) channels were used (see Figure 2-1 for a diagram of sites). The EEG signal was amplified using Grass amplifiers with an analog bandwidth from 0.1 to 100 Hz and electrode impedances were between 2 and 10 kOhms for all subjects. Signals were then digitized at a rate of 512 Hz and stored to a computer.

6.2 The Experiment

The experiment consisted of four tasks with items two through four occurring in a randomized block order:

1. The subject sits quietly and counts the number of red bubble flashes located on the light. This occurs for around 5 minutes.
2. VR condition: The subject receives a goal, such as "Turn on the light", at the bottom of the screen and attempts to achieve the goal by counting the number of red flashes on that goal while fully immersed in a virtual apartment. This condition lasts for approximately 5 minutes.
3. MONITOR condition: The subject receives a goal at the bottom of the screen and attempts to achieve the goal by counting the number of red flashes on that goal while looking at the virtual apartment on a twenty-one inch monitor. This condition lasts for approximately 5 minutes.
4. FIXED DISPLAY condition: The subject receives a goal at the bottom of the screen and attempts to achieve the goal by counting the number of red flashes on that goal while looking at the virtual apartment on a fixed screen inside of the HMD. This condition lasts for approximately 5 minutes.

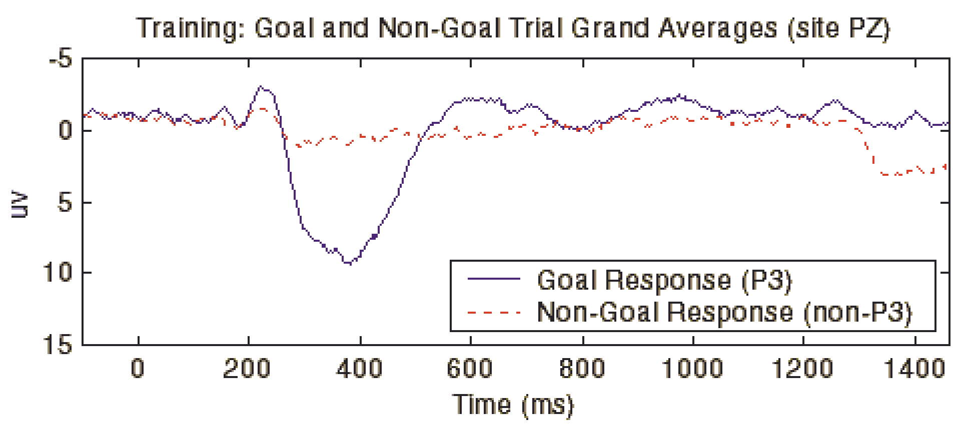


Figure 6-2: Grand averages over all 9 subjects during the training task. The 'Goal' refers to the red flashes that the subject was counting while 'non-Goal' refers to all other red flashes occurring in the virtual apartment during the counting task.

The first task is used to train a signal processing algorithm on a particular subject's P3 response. A total of 300 stimulus presentations are shown with approximately 60 of these being from the button flash on the light. Please see Chapter 4 for a discussion of the signal processing during the experiment. This task is very close to the traditional continuous performance task. In the visual continuous performance task, static images are flashed on a screen and the subject is told to respond when a rare stimulus occurs or to count the number of occurrences of a rare stimulus. This makes the stimulus both rare and task relevant in order to evoke a P3. As an example, given red and yellow stoplight pictures, a P3 should occur if the red picture is less frequent than the yellow and subjects are told to press a mouse button only when they see the red stoplight picture. In a similar manner, even though the apartment scene did not change, red flashes on the light should evoke a P3 if the flashes are counted (making the stimulus task relevant). Other red flashes in the environment should not evoke a P3, as they are not task relevant. As may be seen in Figure 6-2, this was the case for the training task.

The next three tasks were quite similar in that they all required the subject to control different aspects of the apartment. During each control task, the following sequence of events occur:

1. The goal is randomly chosen.
2. The subject tries to achieve this goal up until a maximum of approximately 50 presentations of the goal stimulus.
3. The subject obtains the goal and an action in the virtual apartment is taken. For instance, when the goal has been to turn off the light, the action will be to make the room dark when the goal has been achieved.
4. The next goal is chosen randomly with replacement.

This sequence of events occur until a total of 250 stimulus trials have been presented. Whenever a subject obtained a goal, visual feedback was given and the subject was given a new goal. The visual feedback for the light was to have the room lighten when the light was turned on and darken when it was turned off. Television feedback caused a picture to appear/disappear on the TV when it was controlled. Saying Hi caused the graphical character to appear and saying Bye caused the graphical figure to disappear. The Hi goal only occurred when the character was absent from the scene and the Bye goal when the character was in the scene. The feedback for the stereo consisted of having musical notes appear above the stereo when it was turned on and disappear when the stereo was turned off.

The FIXED DISPLAY condition was chosen to represent a condition between truly immersive VR and staring at a computer monitor. It takes place inside of the HMD, but has a fixed screen like a computer monitor. In addition, the FIXED DISPLAY is most like the condition that patients who could only make eye movements would have, as head movement does not change the location of the screen, since the HMD is on the subject’s head.

6.3 Results

The results from the previously described experiment were used to answer several different questions. Overall, most differences between conditions were small. Subjects varied widely in both performance and in maximal P3 amplitude obtained. The results from individual subjects may be seen in Appendix C.

6.3.1 Different Evoked Potentials in Different Conditions?

The results obtained indicate there are no significant differences between evoked potentials in the three main conditions. Figures 6-3, 6-4, and 6-5 show the individual grand averages between the different conditions. The surprising fact is that the grand averages are almost the same between the conditions as may be seen in Figure 6-6. The training average differed slightly from the other grand averages, perhaps because the counting task was different from the controlling task.

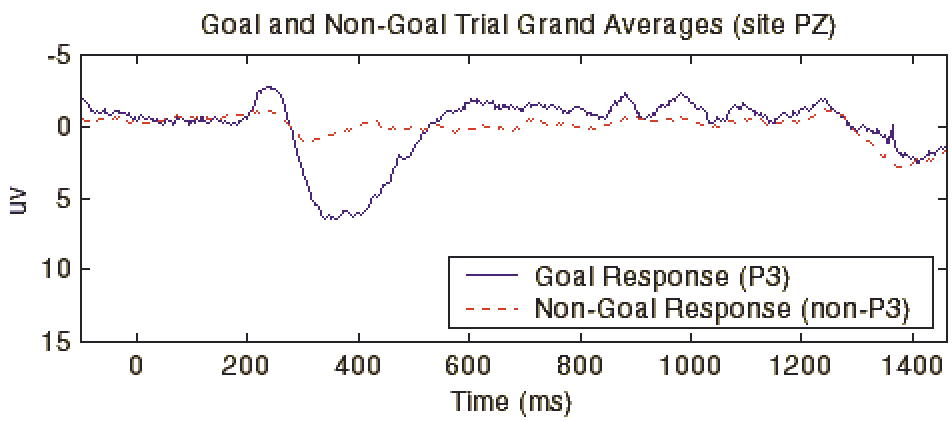


Figure 6-3: Grand average over all subjects for the FIXED DISPLAY condition.

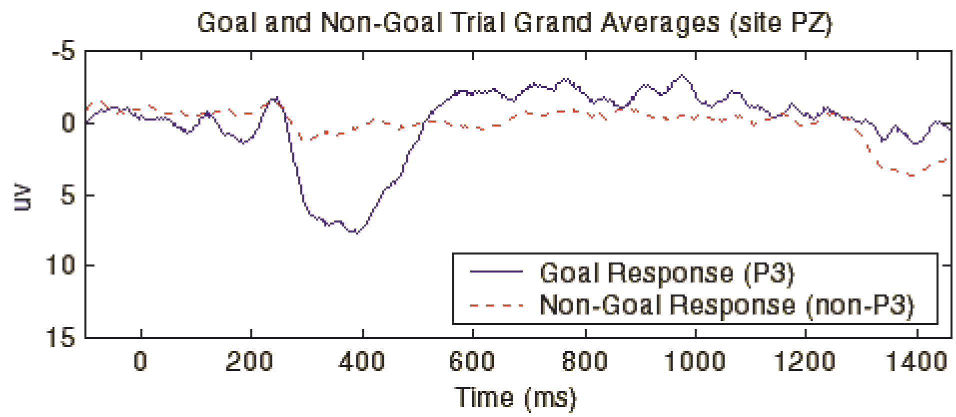


Figure 6-4: Grand average over all subjects for the MONITOR condition.

These results uphold the hypothesis that the act of placing a subject in a VR environment has no significant negative effects on evoked potential recording. This does not mean that subjects will not perform differently in a virtual environment if the task requires more movement than the ordinary EP experiment. This particular experiment did not require extensive physical interaction with the VR apartment and so the EP's obtained were very similar to those obtained while watching a computer monitor.

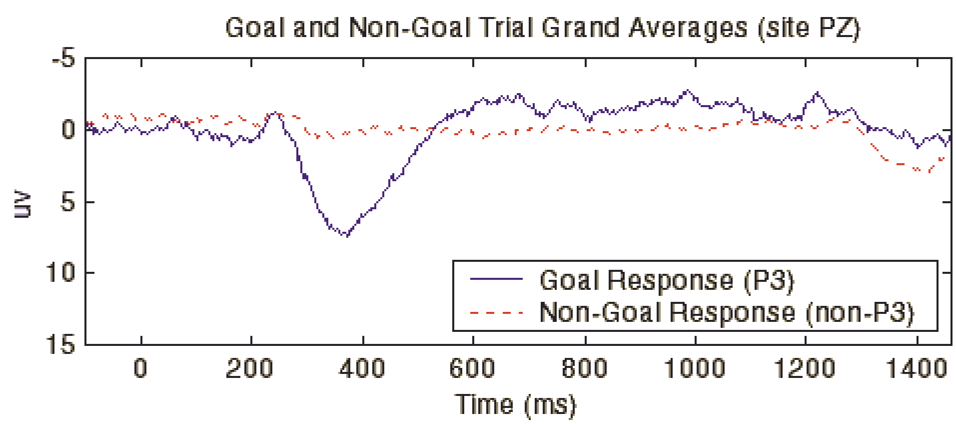


Figure 6-5: Grand average over all subjects for the VR condition.

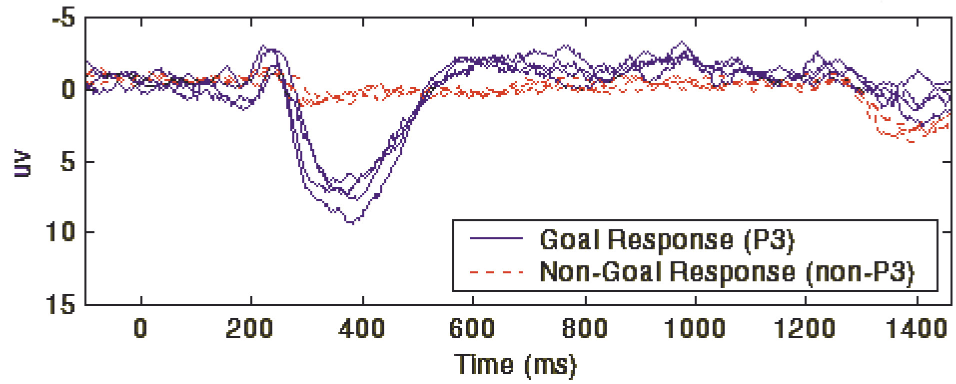


Figure 6-6: Superimposed grand averages for training (the largest Goal average), the FIXED DISPLAY condition, the VR condition, and the MONITOR condition.

6.3.2 Different Performance in Different Conditions?

It may be argued that if there are no noticeable differences in EP's between the conditions, there should be no noticeable differences in performance between the conditions. Since the signal recognition algorithm used was not limited to just recognizing the P3, it was possible that factors other than the P3 could increase recognition and performance. Therefore, performance was considered separately from the EP analysis.

The individual performances of all subjects under different conditions are shown in Table 6-1. While there are no large performance differences, the MONITOR condition did have performance that was better than the FIXED DISPLAY performance ($p < 0.05$). Since the VR condition was neither as good as the MONITOR performance nor as bad as the FIXED DISPLAY performance, it is possible that the limited sample size could be causing the performances to look more different than they actually are.

Table 6-1: Average number of tasks/minute accomplished by each subject.

	Subjects									
Conditions	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*Ave*
VR	*3.8*	*3.7*	*4.5*	*3.3*	*1.9*	*1.7*	*2.6*	*3.1*	*0.9*	*2.83*
MONITOR	*4.5*	*5.6*	*6.1*	*3.5*	*1.2*	*1.2*	*3.3*	*2.9*	*1.3*	*3.29*
FIXED SCREEN HMD	*3.0*	*3.2*	*4.3*	*3.8*	*2.4*	*1.7*	*3.3*	*2.7*	*0.7*	*2.79*

6.3.3 Different Performance over Time?

There are differences in performance over time, but they appear to be related to individual subjects and have not been proven significant in this experiment. Table 6-2 shows the performance of all subjects over time. Some subjects do show a marked increase in performance over time, while others show no increase or may even show a decrease. A trend appears in the average performance, but this trend is due to the few subjects that show a marked performance increase.

The performance of subjects is related to the maximum amplitude of the P3 signal (correlation 0.64) and when some subjects perform better, it may be noticed in plots of the grand averages. This is especially noticeable in a graph of the grand averages over the task order (see Figure 6-7).

Table 6-2: Average number of tasks/minute accomplished over time.

	Subjects									
Conditions	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*Ave.*
First Condition	*3.0*	*3.2*	*4.3*	*3.5*	*1.2*	*1.7*	*3.3*	*3.1*	*0.9*	*2.69*
Second Condition	*4.5*	*3.7*	*4.5*	*3.3*	*2.4*	*1.7*	*3.3*	*2.9*	*1.3*	*3.07*
Third Condition	*3.8*	*5.6*	*6.1*	*3.8*	*1.9*	*1.2*	*2.6*	*2.7*	*0.7*	*3.16*

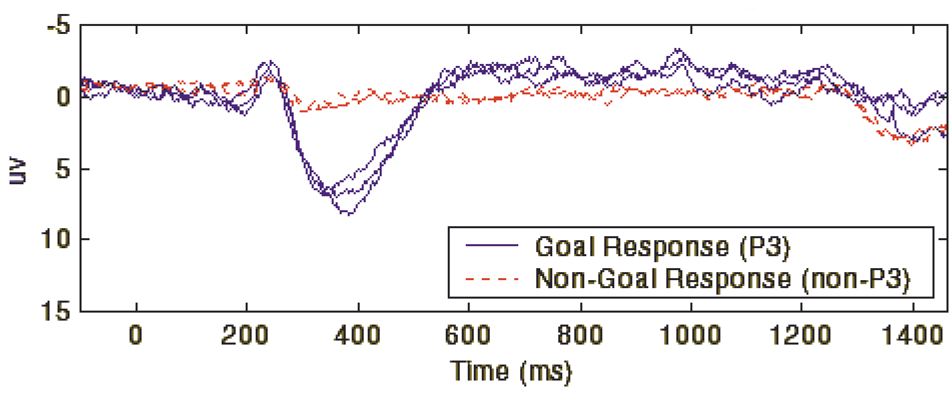


Figure 6-7: Superimposed grand averages for the fourth task, third task, and second task. The third task shows the largest Goal peak followed by the third task and then the second task.

Six of the nine subjects performed better on their last task than the first one. When asked why they might have performed better, subjects made comments like, "I learned to relax and just let the system work", and, "I learned to make the blinking light more relevant to myself.” Subjects could very well have improved by relaxing, as tightening muscles causes muscle-related noise and may decrease the performance of the signal recognition algorithm.

6.3.4 Qualitative Subject Experience Differences Between Conditions?

Subjects’ self-reported qualitative experience did not necessarily match their objective performance. Six out of nine subjects preferred the VR environment while only one of these subjects performed the best in this environment. All subjects rated the FIXED DISPLAY condition the lowest.

Subjects did not like that they could not move their head in order to concentrate on different items in the FIXED DISPLAY condition. This implies that even when subjects maintain eye control, they may not want to use a visual interface if they can’t move their heads. Of the subjects who did not like the VR condition, all of them complained of the HMD fitting in an uncomfortable manner over their electrodes and all of them were among the worst performers.

Some subjects complained that some of the objects were harder to control than others. One of the largest complaints was that the Hi and Bye commands were difficult to activate. In order to investigate this claim, the overall number of tries necessary to achieve a command were investigated for each of the three main conditions. These plots are shown in Figures 6-8, 6-9, and 6-10. There are no large differences, but the Hi and Bye commands do show a larger amount of variation in all conditions.

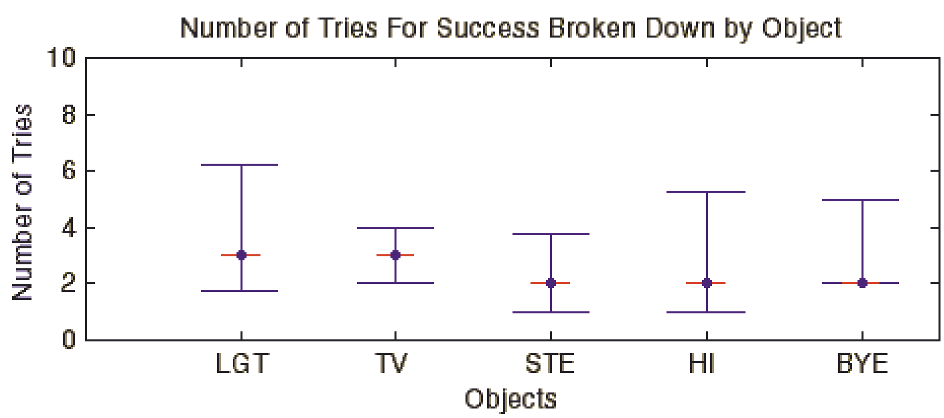


Figure 6-8: The number of tries necessary to achieve the labeled object goal in the FIXED DISPLAY condition. For each object, the top line represents the upper quartile, the bottom line represents the lower quartile, the middle line represents the median, and the plus signs represent data points outside of the box. Occasionally, the lower quartile and the upper quartile are the same and so only one line is drawn.

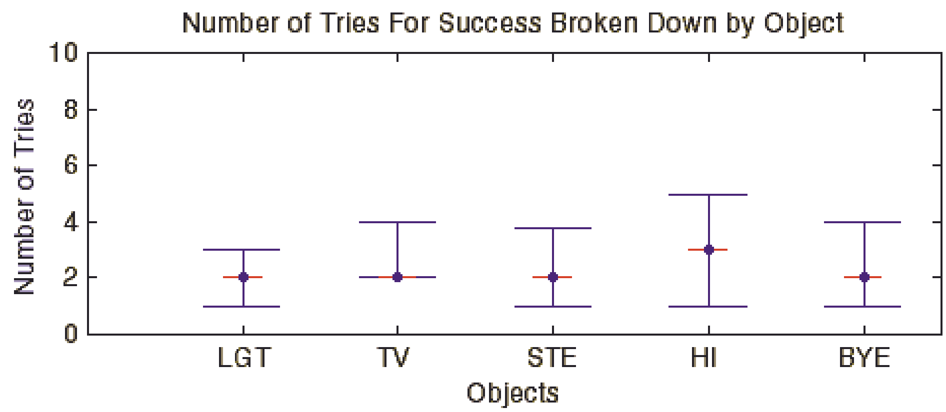


Figure 6-9: The number of tries necessary to achieve the labeled object goal in the MONITOR condition. For each object, the top line represents the upper quartile, the bottom line represents the lower quartile, the middle line represents the median, and the plus signs represent data points outside of the box. Occasionally, the lower quartile and median are the same and so only one line is drawn.

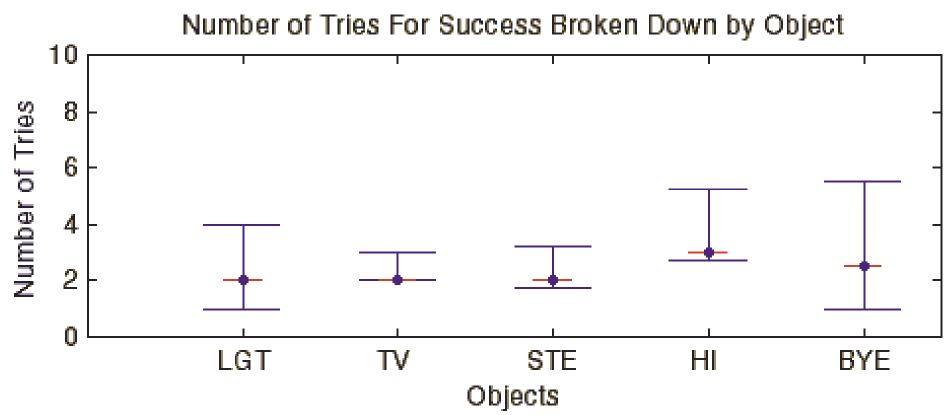


Figure 6-10: The number of tries necessary to achieve the labeled object goal in the VR condition. For each object, the top line represents the upper quartile, the bottom line represents the lower quartile, the middle line represents the median, and the plus signs represent data points outside of the box. Occasionally, the lower quartile and median are the same and so only one line is drawn.

6.4 Discussion and Future Work

The goal of the experiment was to determine the robustness of the P3 component of the evoked potential across several different environments including an immersive virtual environment and a computer monitor. The results show there are no significant differences between responses obtained from within an immersive virtual environment and on a computer monitor. The subjective experiences of subjects mirror this performance, but most subjects preferred the VR condition, even though they did not perform the best on it. This could be due to the novelty of the VR environment. The FIXED DISPLAY condition is the most similar condition to the visual interfaces commonly used by handicapped individuals unable to move their heads, and subjects performed the worst on this condition. The poor results on this condition provide evidence that other types of visual interfaces should be explored, or even that other modalities should be tested. This is one of the goals of future work.

In addition, a subject suggested that flashing red buttons were “obnoxious” and that better results might be obtained for more transparent bubbles of different colors. A pilot experiment using differently colored, almost transparent bubbles was tried on three of the subjects during the same experimental session as the other tasks. The performance results of these subjects are shown in Table 6-3. The task with different colored bubbles is labeled “MONITOR with Diff. Colors” as it is basically the MONITOR condition without red colored flashing bubbles.

Table 6-3: Average number of tasks/minute accomplished by each subject.

	Subjects	*6*	*8*	*9*
Conditions				
MONITOR		*1.2*	*1.3*	*2.9*
MONITOR with Diff. Colors		*1.8*	*1.7*	*5.0*

It may be seen that all subjects who tried this task performed better than they did on the MONITOR condition, but the results are not statistically significant due to the small number of subjects. This may be due to the following:

1. The novelty of the new environment caused larger P3’s, that in turn made signal recognition easier.
2. There is something better about the different colored bubbles that make them easier for subjects to user for control.

In order to make an application usable, it is normally good not to have novel occurrences in the application. It is possible that exploiting the known novelty effects of the P3 may make signal recognition easier and make the interface more interesting. Many subjects complain that the control tasks are boring and novel flashes might make the application more interesting for users, since they have to wait for stimuli to respond to. While locked-in patients would be expected to have a very high level of motivation to accomplish the given tasks independent of the user interface, a more interesting user interface might make the tasks more enjoyable to accomplish. This needs to be addressed in future work.

7 Conclusions

Winning isn't everything. The desire to win is everything. In fact, it's the only thing!

- *Vince Lombardi*

In creating this document, I initially believed that just to have a working signal recognition algorithm or a working BCI would mean "winning". Along the way, the desire to create a BCI that not only worked, but also allowed users to enjoy working with it crept in. As Vince Lombardi says above, the desire to accomplish or win is more important than the actual product derived. It is this desire that keeps a researcher working until well into the early morning when the last results would be sufficient for a paper.

The role of the computer in a BCI is important and as such the computer processing surrounding the BCI must improve. Improvements may be made in the areas of system design, signal processing, user applications, and in usability. Until recently, most improvements were made in the area of signal processing and in special hardware to acquire the EEG data. This is changing as researchers have accomplished the goal of creating more effective acquisition and signal processing systems.

7.1 Thesis Summary

This dissertation has explored several issues in creating a BCI.Results from several experiments suggest the following:

- A brain-computer interface should be designed for flexibility. Important areas for flexibility include signal processing, and user applications.
- Virtual environments may be used in evoked potential BCI applications.
- Signal recognition only accounts for part of the performance and usability in a BCI system. A trade-off between recognition accuracy and time often occurs and it may be preferable to choose a worse recognition accuracy in order to maximize the system throughput for a user. Even with different throughput levels, the user may believe the slower system is better to use.
- The presence of signal artifacts decreases recognition, but not necessarily by adding false positives. When an artifact swamps the signal of interest, more false negatives occur and throughput decreases.

A flexible system was developed and described. Flexibility in user applications is necessary, because more research needs to be done in order to determine the best types of applications for a BCI. Signal processing flexibility is also important, because new preprocessing and recognition algorithms may increase signal recognition and lead to better BCI performance. The BCI in this document has been created in order to make signal recognition and user applications easy to change or update.

Recognizing that virtual reality may prove useful for training individuals to use a BCI, for providing complex and controllable experimental environments for those improving BCIs, and for motivational reasons, several experiments have been performed in virtual environments. This dissertation has shown that evoked potentials may be reliably obtained in complex virtual environments and that the P3 component of the evoked potential is robust over a variety of different environments.

Several signal processing routines were compared. Some routines may lead to better signal recognition, but at the cost of increased time. It may be better to choose a quicker routine that leads to a lower recognition rate. It was shown that all the algorithms used were fairly robust under a variety of situations and that when the raw electrophysiological signals where obscured by artifacts, the combinations of preprocessing and recognition routines tended to produce false negative rather than false positive results. Even with a lower recognition rate, users may like a more entertaining system such as a virtual environment over watching a computer monitor.

7.2 Future Work

Many paths to future work exist in BCIs. Signal processing algorithms need to be improved. One way of doing this might be to take several of the algorithms presented in Chapter 4 and combine them to yield an algorithm with better recognition abilities without a change in the amount of time taken to recognize the signal.

The system can and should be improved over time. User applications must improve in order to make individuals want to learn to use a BCI. Even with these improvements, there is room for a lot of work in optimizing the BCI system for users. One question of primary importance in this task is how much the individual should adapt to the system vs. how much the computer should adapt to the user. Current systems assume that the human should do most of the learning and current BCI users have been trained in lengths of time up to a year. Future systems should allow more variability in training with a heavier weight on computer adaptability.

The known problems of the BCI system should be solved and the program released as an open source program. In this way, researchers will be able to use the program in their experiments. In order to achieve this goal, better software testing abilities need to be added as well the ability to display EEG data on-line.

With a flexible BCI, it is possible to use multiple sensory inputs for control. Multiple EEG signals may be recognized in order to expand the processing capabilities of a BCI system. As an example, mu waves, a type of EEG signal related to motor function, may be used for cursor control while using the P3 component as a mouse click. Another modality such as eye tracking may be used in order to control the computer. Different users may need to use different muscles or EEG signals for optimal use.

As mentioned in Chapter 6, the flashing red buttons on controllable virtual apartment items were perceived as "obnoxious". This led to trying a different combination of colored buttons that were more transparent. A pilot experiment using differently colored, almost transparent bubbles was tried on three of the subjects during the same experimental session as the other virtual apartment tasks. Performance results indicated that subjects performed better on the tasks with the more transparent bubbles. It is currently unknown whether or not these results were caused by the novelty of differently colored buttons or for another as of yet unknown reason.

In order to make an application usable, it is normally good *not* to have novel occurrences in the application. It is possible that exploiting the known novelty effects of the P3 may make signal recognition easier and make the interface more interesting. Many subjects complain that the control tasks are boring and novel flashes might make the application more interesting for users, since they have to wait for stimuli to respond to. This needs to be addressed in future work.

8 Bibliography

[1] H. Al-Nashi, “A maximum likelihood method for estimating EEG evoked potentials", *IEEE Trans. on Biomed. Eng.*, 33:12, pp. 1087—1095, 1986.

[2] Jean-Dominique Bauby, *The Diving Bell and the Butterfly: A Memoir of Life in Death*, Trans. by Jeremy Leggatt, Alfred A. Knopf, New York, 1997.

[3] J.D. Bayliss, J.A. Gualtieri, and R.F. Cromp, “Analyzing hyperspectral data with Independent Component Analysis”, *Proc. SPIE Applied Image and Pattern Recognition Wrkshp.*, October, 1997.

[4] J.D. Bayliss and D.H. Ballard, "The Effects of Eye Tracking in a VR Helmet on EEG Recording", TR 685, University of Rochester National Resource Laboratory for the Study of Brain and Behavior, May, 1998.

[5] J.D. Bayliss and D.H. Ballard, "Recognizing Evoked Potentials in a Virtual Environment", Advances in Neural Information Processing Systems 12, 2000.

[6] J.D. Bayliss and D.H. Ballard, "Single Trial P300 Epoch Recognition in a Virtual Environment”, Neurocomputing, v.32-33, pp. 637—642, 2000.

[7] J.D. Bayliss and D.H. Ballard, "A Virtual Reality Testbed for Brain-Computer Interface Research”, IEEE Trans. On Rehabilitation Engineering, 8:2, 2000.

[8] J.D. Bayliss and B. Auernheimer, “Using a Brain-Computer Interface in Virtual and Real Worlds”, Proc. of HCI Internat’l, v.1, pp. 312—316, 2001.

[9] A.J. Bell and T.J. Sejnowski, “Fast blind separation based on information theory”, *Proc. Intern. Symp. On Nonlinear Theory and Applications (NOLTA)*, Las Vegas, December 1995.

[10] A.J. Bell and T.J. Sejnowski, “An information-maximization approach to blind separation and blind deconvolution”, *Neural Computation*, v.7, pp. 1129—1159, 1995.

[11] J.M. Belsh and P.L. Schiffman (editors), *Amyotrophic lateral sclerosis: diagnosis and management for the clinician*, Futura Publishing Co., Inc., Armonk, NY, 1996.

[12] N. Birbaumer, N. Ghanayim, T. Hinterberger, I. Iversen, B. Kotchoubey, A. Kubler, J. Perelmouter, E. Taub, H. Flor, "A Spelling Device for the Paralysed", *Nature*, v.398, pp. 297—298, 1999.

[13] Grady Booch, *Object-Oriented Analysis and Design*, *Second Edition*, The Benjamin/Cummings Pub. Co., Inc, Redwood City, CA, 1994.

[14] *Brain-Computer Interface Technology: Theory and Practice: First International Meeting Program and Papers*, The Rensselaerville Institute, Rensselaerville, New York, June 16-20, 1999.

[15] J.K. Chapin and G. Gaal, “Robotic Control from Realtime Transformation of Multi-neuronal Population Vectors”, *Brain-Computer Interface Technology: Theory and Practice: First International Meeting Program and Papers*, The

Rensselaerville Institute, Rensselaerville, New York, pp. 54, June 16-20, 1999.
[16] J.K. Chapin, K.A. Moxon, R.S. Markowitz, and M.A.L. Nicoleslis, “Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex”, *Nature Neurosci., 2:7,* pp. 664—670, 1999.
[17] R.M. Chapman and H.R. Bragdon, Evoked responses to numerical and non-numerical visual stimuli while problem solving, *Nature*, v.203, pp. 1155—1157, 1964.
[18] J. Cohen and J. Polich, “On the number of trials needed for P300”, *Intern’l J. of Psychophys.*, v.25, pp. 249—255, 1997.
[19] E. Cole, P. Dehdashti, L. Petti and M. Anger, “Participatory design for sensitive interface parameters: Contributions of traumatic brain injury patients to their prosthetic software”, *CHI94 Conference Companion*, ACM, 1994.
[20] P. Comon, “Independent Component Analysis: A new concept”, *Signal Proc.*, v.36, pp. 287—314, 1994.
[21] Cyberlink System, http://www.brainfingers.com, 2001.
[22] E. Donchin, “Discriminant analysis in average evoked response studies: the study of single trial data”, Electroenceph. Clin. Neurophysiol, v.27, pp. 311—314, 1969.
[23] S. Devulapalli, "Non-linear component analysis and classification of EEG during mental tasks", *Masters Thesis at Colorado State University*, 1996.
[24] The Doctor's Guide to ALS Information and Resources, *http://www.pslgroup.com/ALS.HTM/#Disease*, 1998.
[25] L.A. Farwell and E. Donchin, “Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials”, Electroenceph. Clin. Neurophysiol., pp. 510—523, 1988.
[26] T. Fernandez, T. Harmony, M. Rodriguez, J. Bernal, and J. Silva, "EEG activation patterns during the performance of tasks involving different components of mental calculation", *Electroenceph. Clin. Neurophysiol.*, v.94, pp. 175—182, 1995.
[27] M. Fowler, K. Scott, and G, Booch, *UML Distilled: Second Ed.*, Addison-Wesley, Reading, Mass., 1999.
[28] W.J. Freeman, *Induced rhythms of the brain*, Birkhaeser Boston Inc., 1991.
[29] W.J. Freeman, Chaos in the brain: Possible roles in biological intelligence, *International Journal of Intelligent Systems*, 10, pp. 71—88, 1995.
[30] Pierre Gloor, Hans Berger On the Electroencephalogram of Man: The Fourteen Original Reports on the Human Electroencephalogram, N. Elsevier Science Publishers, Amsterdam, 1969.
[31] C. Guger, A. Schlögl, D. Walterspacher, and G. Pfurtscheller, "Design of an EEG-based brain-computer interface (BCI) from standard components running in real-time under Windows," *Biomed. Technik*, v.44, pp. 12-16, 1999.
[32] A.J. Hanlan, *Autobiography of dying*, Doubleday, Garden City, NY, 1979.

[33] B. Hjorth, On-line transformation of EEG scalp potentials into orthogonal source, *Electroenceph. Clin. Neurophysiol.*, 39, pp. 111—118, 1975.

[34] R.E. Isaacs, D. J. Weber, and A.B. Schwartz, “Real-time control of a cortical neural prothesis”, *Brain-Computer Interface Technology: Theory and Practice: First International Meeting Program and Papers*, The Rensselaerville Institute, Rensselaerville, New York, pp. 61—62, June 16-20, 1999.

[35] H.H. Jasper, "The Ten-Twenty electrode system of the international federation," *Electroencephalogram and Clinical Neurophysiology*, v.10, pp. 371--375, 1958.

[36] K.S. Jones, M.S. Middendorf, G. Calhoun, and G. McMillan, "Evaluation of an Electroencephalographic-based Control Device", *Proc. of the 42nd Annual Mtg of the Human Factors and Ergonomics Society*, pp. 491—495, 1998.

[37] T.P. Jung, C. Humphries, T. Lee, S. Makeig, M.J. McKeown, V. Iragui, and T.J. Sejnowski, "Extended ICA Removes Artifacts from Electroencephalographic Recordings", *Advances in Neural Information Processing Systems*, v.10, 1998.

[38] T-P Jung, S. Makeig, M. Westerfield, J. Townsend, E. Courchesne, T.J. Sejnowski, “Independent component analysis of single-trial event-related potentials”, *ICA99: First Internat’l Wrkshp. on Independent Component Analysis and Signal Separation*, pp. 173—178, 1999.

[39] J. Kalcher, D. Flotzinger, Ch. Neuper, S. Golly, and G. Pfurtscheller, “Graz brain-computer interface II: towards communication between humans and computers based on online classification of three different EEG patterns",*Medical and Biological Engineering and Computing*, v.34, pp. 382—388, 1996.

[40] K. Kaneko, Globally coupled chaos violates the law of large numbers, *Physical Review Letters*, 65, 1391—1394, 1990.

[41] K. Kaneko, Mean field fluctuation in network of chaotic elements, *Physica D*, 55, 368—384, 1992.

[42] P.R. Kennedy and R.A.E. Bakay, “Restoration of neural output from a paralyzed patient by a direct brain connection”, *NeuroReport*, v.9, pp. 1707—1711, 1998.

[43] P.R. Kennedy and R.A.E. Bakay, “Direct control of a computer from the human central nervous system”, *Brain-Computer Interface Technology: Theory and Practice: First International Meeting Program and Papers*, The Rensselaerville Institute, Rensselaerville, New York, pp. 65—70, June 16-20, 1999.

[44] K.L. Kilgore, P.H. Peckham, G.B. Thrope, M.W. Keith, and K.A. Gallaher-Stone, "Synthesis of Hand Grasp Using Functional Neuromuscular Stimulation", *IEEE Trans. on Biomedical Engineering*, 36:7, pp. 761—770, 1989.

[45] A. Kralj, T. Bajd, and R. Turk, "Enhancement of gait restoration in spinal injured patients by functional electrical stimulation", *Clinical Orthopaedics and Related Research*, v.233, pp. 34—43, 1988.

[46] J.R. LaCourse and F.C. Hludik Jr., "An Eye Movement Communication-Control System for the Disabled", *IEEE Trans. on Biomedical Engineering,* 37:12, pp. 1215—1220, 1990.

[47] D. Lange and G. F. Inbar, "A robust parametric extimator for single-trial movement related brain potentials", *IEEE Trans. On Biomed. Eng.*, 43:4, pp. 341—346, 1996.

[48] S. Lin, Y. Tsai, and C Liou, "Conscious mental tasks and their EEG signals", *Medical and Biol. Engineering and Comput.*, v.31, pp. 421—425, 1993.

[49] D.J. McFarland, G.W. Neat, R.F. Read, and J.R. Wolpaw, "An EEG-based method for graded cursor control", *Psychobiology*, 21:1, pp. 77—81, 1993.

[50] D. J. McFarland, A. T. Lefkowicz, and J. R. Wolpaw, "Design and operation of an EEG-based brain-computer interface with digital signal processing technology", *Behav. Research Methods, Instruments, and Computers*, 29:3, pp. 337--345, 1997.

[51] S. Makeig, T. Jung, A.J. Bell, D. Ghahremani, and T.J. Sejnowski, "Blind Separation of Auditory Event-related Brain Responses into Independent Components", *Proc. Nat'l Acad. Sci. USA*, v.94, pp. 10979—10984, 1997.

[52] A.H. Malcolm, *This far and no more: a true story*, Times Books, New York, 1987.

[53] E.B. Marsolais and R. Kobetic, "Development of a practical electrical stimulation system for restoring gait in the paralyzed patient", *Clinical Orthopaedics and Related Research*, v.233, pp. 64—74, 1988.

[54] M.S. Middendorf, G. McMillan, G. Calhoun, and K.S. Jones, "Brain-computer interfaces based on the steady-state visual-evoked response", *Brain-Computer Interface Technology: Theory and Practice: First International Meeting Program and Papers*, The Rensselaerville Institute, Rensselaerville, New York, pp. 78—82, June 16-20, 1999.

[55] M.M. Moore and P.R. Kennedy, "Human factors issues in the Neural Signals direct brain-computer interface", *Proceedings of ASSETS'00*, ACM, 2000.

[56] W.T. Nelson, L.J. Hettinger, J.A. Cunningham, and M.M. Roe, "Navigating through virtual flight environments using brain-body-actuated control", *Proceedings. IEEE 1997 Virtual Reality Annual Intern. Symposium*, pp. 30—37, 1997.

[57] D. A. Norman, *The design of everyday things*, Doubleday, New York, 1990.

[58] P.L. Nunez, *Neocortical Dynamics and Human EEG Rhythms*, Oxford University Press, New York, 1995.

[59] B.A. Olshausen, "Learning linear, sparse, factorial codes, *MIT AI-memo 1580*, 1996.

[60] Marco Onofrj, Donato Melchionda, Astrid Thomas, and Tommaso Fulgente, "Reappearance of event-related P3 potential in locked-in syndrome", *Cognitive Brain Research*, v.4, pp. 95—97, 1996.

[61] C. W. Palmer, A. J. Derbyshire, and W. Lee, “A method of analyzing individual cortical responses to auditory stimuli”, *Electroenceph. Clin. Neurophysiol.*, v.20, pp. 204—206, 1966.

[62] B.A. Pearlmutter, L.C. Parra, “A context-sensitive generalization of ica”, *Proc. ICONIP ’96*, 1996.

[63] M. Peltoranta and G. Pfurtscheller, “Neural network based classification of non-averaged event-related EEG responses”, *Med. and Biol. Eng.. and Comput.*, v.32, pp. 189—198, 1994.

[64] G. Pfurtscheller and A. Aranibar, "Event-related cortical desynchronization detected by power measurements of scalp EEG", *Electroenceph. Clin. Neurophysiol.*, v.42, pp. 817—826, 1977.

[65] G. Pfurtscheller, D. Flotzinger, W. Mohl, and M. Peltoranta, "Prediction of the side of hand movements from single-trial multi-channel EEG data using neural networks", *Electroenceph. Clin. Neurophysiol.*, v.82, pp.313—315, 1992.

[66] G. Pfurtscheller, D. Flotzinger, and J. Kalcher, "Brain-computer Interface -- a new communication device for handicapped persons", *J. of Microcomputer Applications*, v.16, pp. 293—299, 1993.

[67] G. Pfurtscheller, J. Kalcher, Ch. Neuper, D. Flotzinger, and M. Pregenzer, "On-line EEG classification during externally-paced hand movements using a neural network-based classifier", *Electroenceph. Clin. Neurophysiol.*", v.99, pp. 416—425, 1996.

[68] G. Pfurtscheller, D. Flotzinger, M. Pregenzer, J. Wolpaw, and D. McFarland, "EEG-based Brain Computer Interface (BCI)", *Medical Progress through Technology*, v.21, pp. 111-121, 1996.

[69] J. Polich, "P300 Clinical Utility and Control of Variability", *J. of Clinical Neurophysiology*, 15:1, pp. 14—33, 1998.

[70] J. B. Polikoff, H. T. Bunnell, and W. J. Borkowski Jr., “Toward a P300-based computer interface”, *Proc. of the RESNA '95 Annual Conf.*, RESNAPRESS, Arlington, Va, pp. 178—180, 1995.

[71] R. P.N. Rao, Visual Attention during Recognition, *Advances in Neural Information Processing Systems 10*, 1998.

[72] H.E. Rosvold, A.F. Mirsky, I. Sarason, E.D. Bransome Jr., and L.H. Beck, "A Continuous Performance Test of Brain Damage", *J. Consult. Psychol.*, v.20, pp. 343—350, 1956.

[73] George Santayana, *The life of reason; or, The phases of human progress*, Collier Books, New York, 1962.

[74] B. Shneiderman, Designing the user interface: Strategies for effective human-computer interaction, 3rd ed., Addison Wesley, Reading, Mass., 1998.

[75] A. Schwartz, *Killing me softly: the inspiring story of a champion of the poor*, Alba House, New York, 1993.

[76] H.V. Semlitsch, P. Anderer, P Schuster, and O. Presslich, "A solution for reliable and valid reduction of ocular artifacts applied to the P300 ERP", *Psychophys.*, v.23, pp. 695—703, 1986.

[77] F.H. Lopes da Silva, W.Storm van Leeuwen, and A. Remond, *Handbook of Electroencephalography and Clinical Neurophysiology: Volume 2, Clinical Applications of Computer Analysis of EEG and other Neurophysiological Signals*, N. Elsevier Science Publishers}, Amsterdam, 1986.

[78] R. Spehlmann, *Spehlmann's EEG Primer*, N. Elsevier Science Publishers, Amsterdam, 1991.

[79] E.E. Sutter, “The brain response interface: communication through visually-induced electrical brain responses", *Journal of Microcomputer Applications*, v. 15, pp. 31—45, 1992.

[80] K. M. Spencer, O. Karni, R. Wijesinghe, and E. Donchin, “The mental prothesis: assessing the speed of a P300-based brain-computer interface”, *Brain-Computer Interface Technology: Theory and Practice: First International Meeting Program and Papers*, The Rensselaerville Institute, Rensselaerville, New York, pp. 65—70, June 16-20, 1999.

[81] S. Sutton and M. Braren and J. Zublin and E. John, Evoked potential correlates of stimulus uncertainty, *Science*, v.150, pp. 1187—1188, 1965.

[82] Clemens Szyperski, *Component Software: Beyond Object-Oriented Programming*, ACM Press, New York, 1998.

[83] N. Toda, N. Murai, and S. Usui, A measure of nonlinearity in time series using neural network prediction model, *Artificial Neural Networks 2*, 2, 1117—1120, 1992.

[84] T.M. Vaughan, J.R. Wolpaw, and E. Donchin, "EEG-Based Communication: Prospects and Problems", *IEEE Trans. on Rehabilitation Engineering*, 4:4, pp. 425—430, 1996.

[85] Jacques J. Vidal, Toward Direct Brain-Computer Communication, In L. J. Mullins (ed.) *Annual Review of Biophysics and Bioengineering*, Annual Reviews Inc, pp. 157—180, 1973.

[86] R. Vigario, "Extraction of ocular artifacts from eeg using independent component analysis", *Electroenceph. Clin. Neurophysiol.*, v.103, pp. 395—404, 1997.

[87] W.G. Walter, "The contingent negative variation: an electrocortical sign of significant association in the human brain", *Science*, v.146, p. 434, 1964.

[88] J.R. Wolpaw, D.J. McFarland, G.W. Neat, and C.A. Forneris, "An EEG-based brain-computer interface for cursor control", *Electroenceph. Clin. Neurophysiol.*, v.78, pp. 252—258, 1991.

[89] J.J. Wright and D.T.J. Liley, Dynamics of the brain at global and microscopic scales: Neural networks and the EEG, *Behavioral and Brain Sciences*, 19, pp. 285—320, 1996.

Appendix A
Recognition data for low artifact conditions

Data was obtained eight subjects in an environmental control experiment. This data has been used for off-line analysis over three different algorithms: peak picking, correlation, and an algorithm that used a robust Kalman filter for preprocessing data and correlation for recognition. An ROC curve representing how each algorithm performs for different parameter values is shown for each subject. The number of trials used to produce each set of averages for peaking picking, correlation, and the algorithm using the robust Kalman filter are shown in Tables A-1, A-2, and A-3 (respectively).

The averages for each algorithm result category (true positive, false negative, false positive, and true negative) are also shown for each subject. Subject plots are labeled S1 for subject 1, S2 for subject 2, etc. The solid line in each average plot represents the appropriate type of trial average at site PZ, while the dotted line represents the average vertical eye movement voltage. All averages are filtered at 30 Hz for display purposes.

The data from heavy breathing is shown in Chapter 5. Please see this chapter for a complete discussion of results.

Table A-1: The number of trial used for each of the subject averages in the peak picking algorithm. Subject 7 has fewer trials because only trials with a maximum vertical eye channel difference of less than 50 microvolts were used.

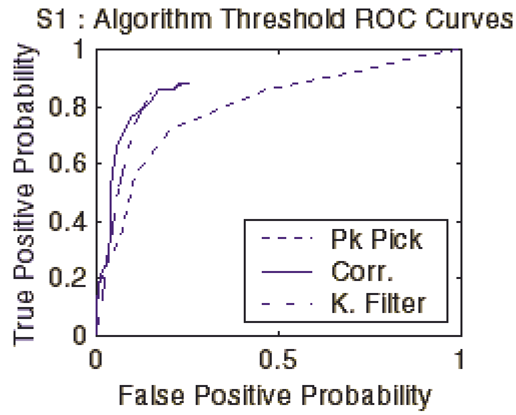
	True Pos.	*False Pos.*	*False Neg.*	*True Neg.*
Subject 1	*28*	*23*	*22*	*201*
Subject 2	*32*	*24*	*31*	*215*
Subject 3	*24*	*19*	*22*	*178*
Subject 4	*15*	*18*	*27*	*164*
Subject 5	*21*	*22*	*46*	*203*
Subject 6	*10*	*23*	*57*	*203*
Subject 7	*3*	*7*	*9*	*62*
Subject 8	*22*	*20*	*18*	*172*

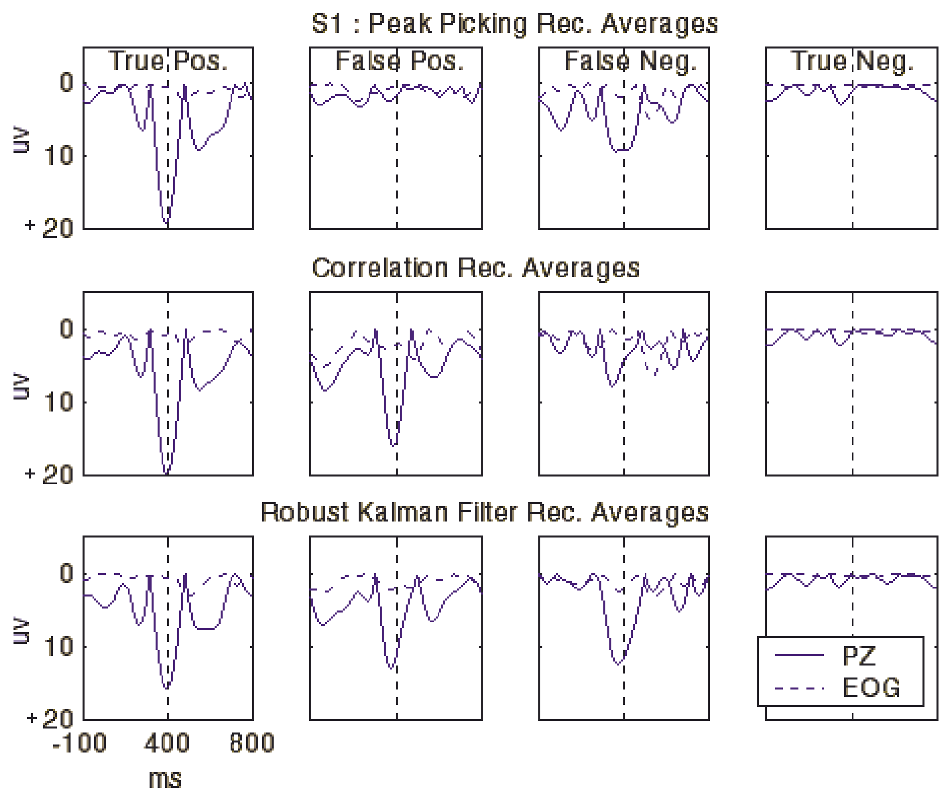
Table A-2: The number of trial used for each of the subject averages in the correlation algorithm. Subject 7 has fewer trials because only trials with a maximum vertical eye channel difference of less than 50 microvolts were used.

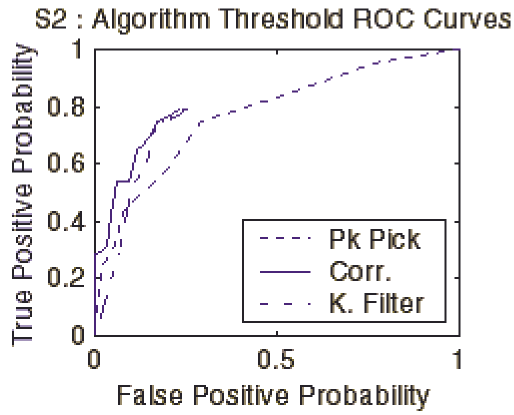
	True Pos.	*False Pos.*	*False Neg.*	*True Neg.*
Subject 1	*38*	*22*	*12*	*202*
Subject 2	*36*	*23*	*27*	*216*
Subject 3	*36*	*20*	*10*	*177*
Subject 4	*21*	*18*	*21*	*164*
Subject 5	*29*	*23*	*38*	*202*
Subject 6	*29*	*23*	*38*	*203*
Subject 7	*7*	*7*	*5*	*62*
Subject 8	*27*	*19*	*13*	*173*

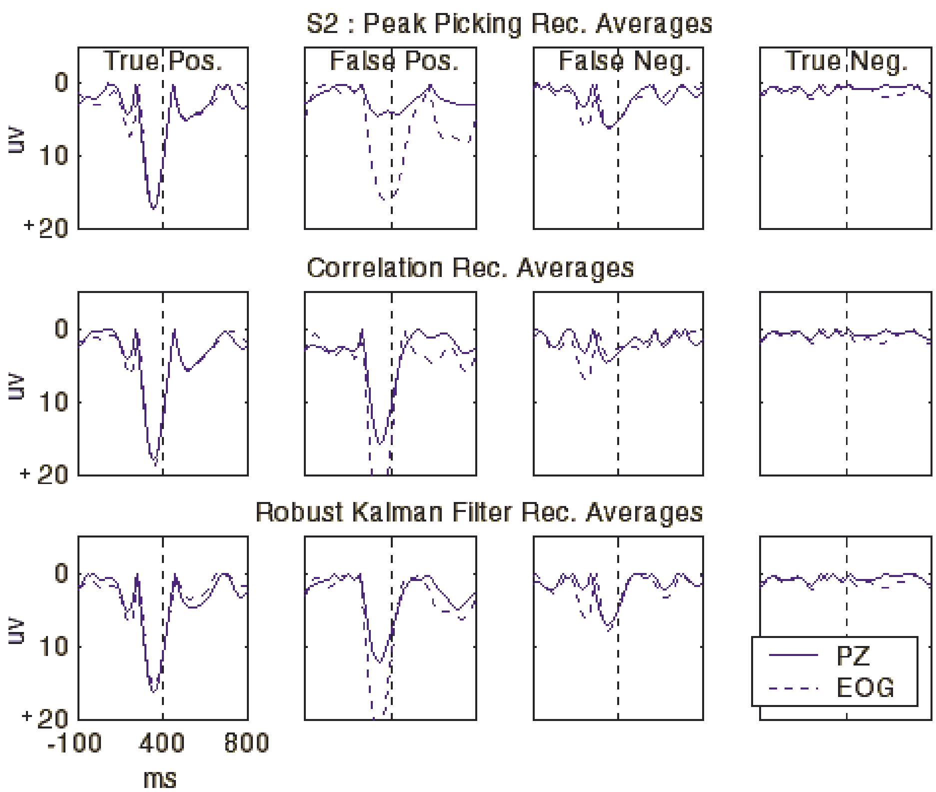
Table A-3: The number of trial used for each of the subject averages in the algorithm using the robust Kalman filter. Subject 7 has fewer trials because only trials with a maximum vertical eye channel difference of less than 50 microvolts were used.

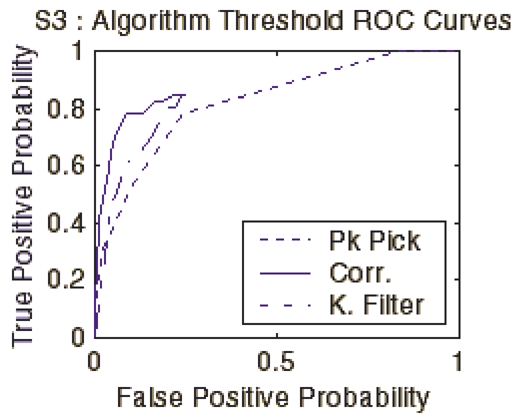
	True Pos.	*False Pos.*	*False Neg.*	*True Neg.*
Subject 1	*36*	*22*	*14*	*202*
Subject 2	*32*	*24*	*31*	*215*
Subject 3	*29*	*20*	*17*	*177*
Subject 4	*23*	*18*	*19*	*164*
Subject 5	*26*	*22*	*41*	*203*
Subject 6	*21*	*23*	*46*	*203*
Subject 7	*7*	*7*	*5*	*62*
Subject 8	*10*	*19*	*30*	*173*

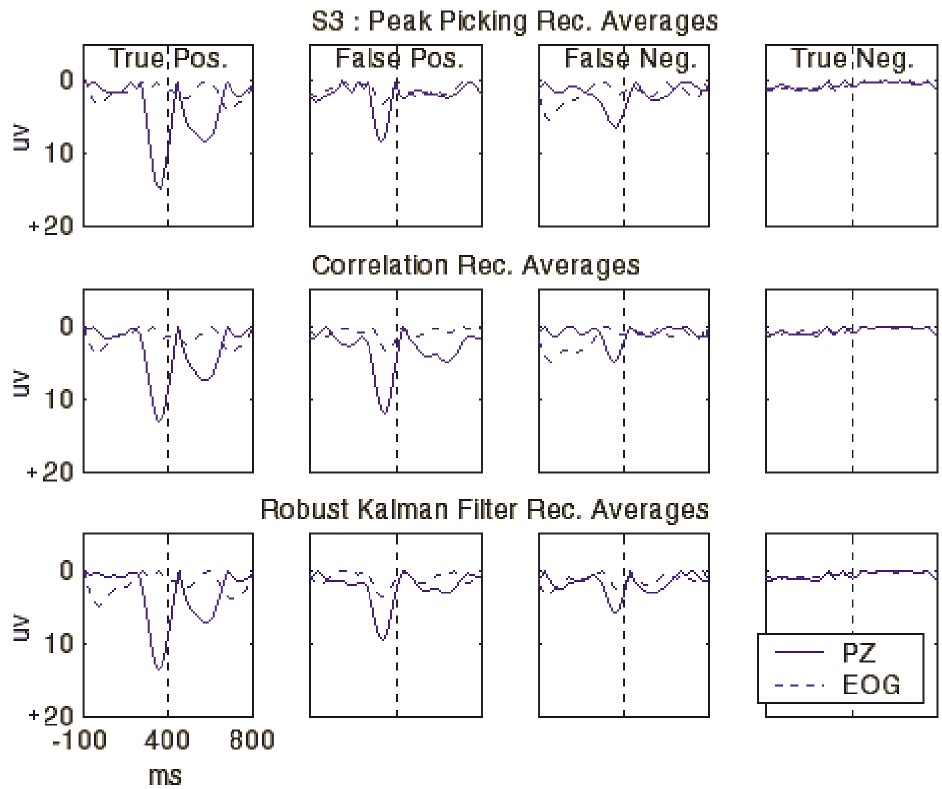

S1 : Algorithm Threshold ROC Curves
True Positive Probability
False Positive Probability
Pk Pick
Corr.
K. Filter
0
0.2
0.4
0.6
0.8
1
0.5

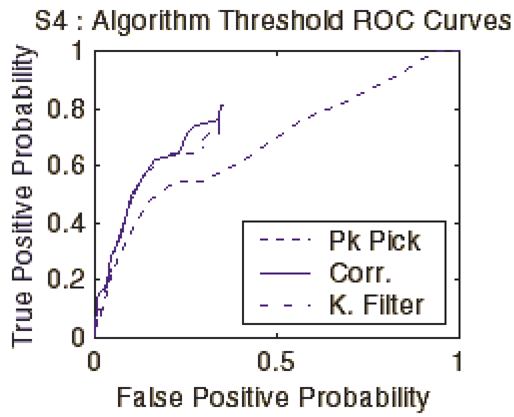

S1 : Peak Picking Rec. Averages
True Pos.
False Pos.
False Neg.
True Neg.
Correlation Rec. Averages
Robust Kalman Filter Rec. Averages
uv
0
10
+ 20
-100
400
800
ms
PZ
EOG

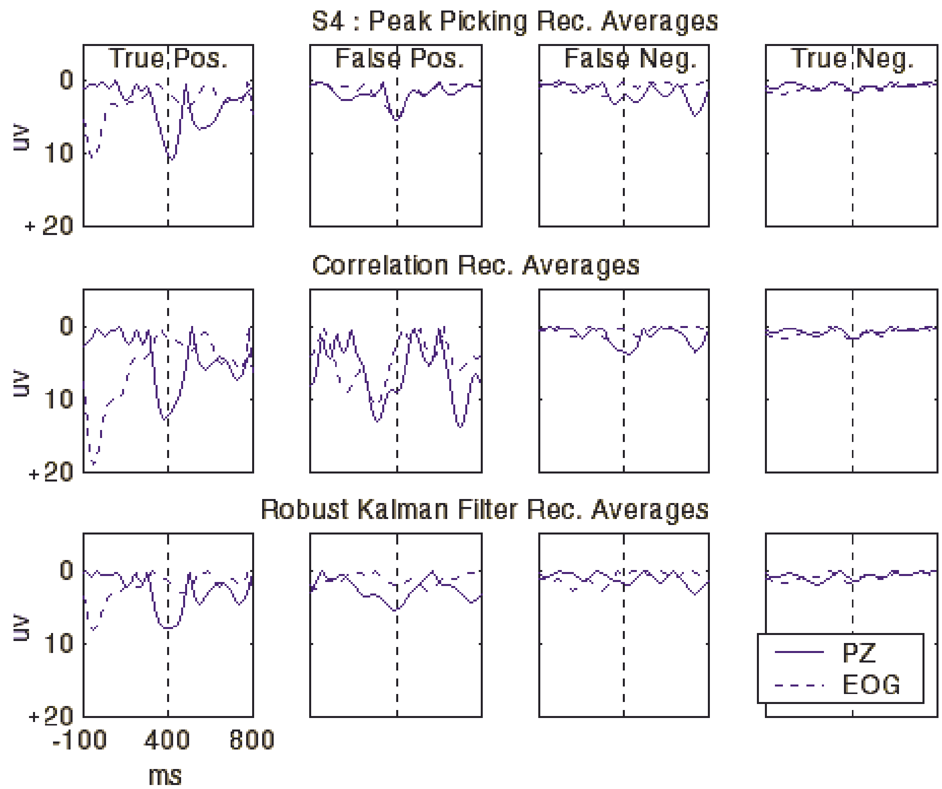

S2 : Algorithm Threshold ROC Curves
True Positive Probability
False Positive Probability
Pk Pick
Corr.
K. Filter
0
0.2
0.4
0.6
0.8
1
0.5

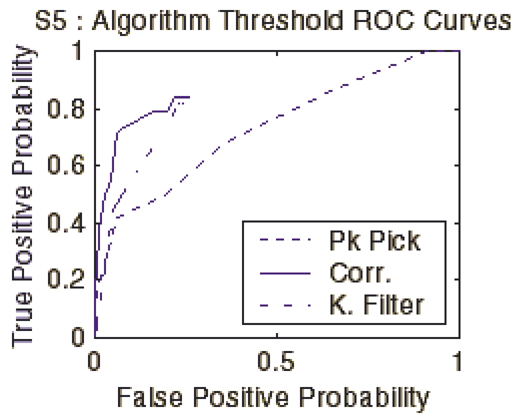

S2 : Peak Picking Rec. Averages
True Pos.
False Pos.
False Neg.
True Neg.
Correlation Rec. Averages
Robust Kalman Filter Rec. Averages
uv
0
10
+20
-100
400
800
ms
PZ
EOG

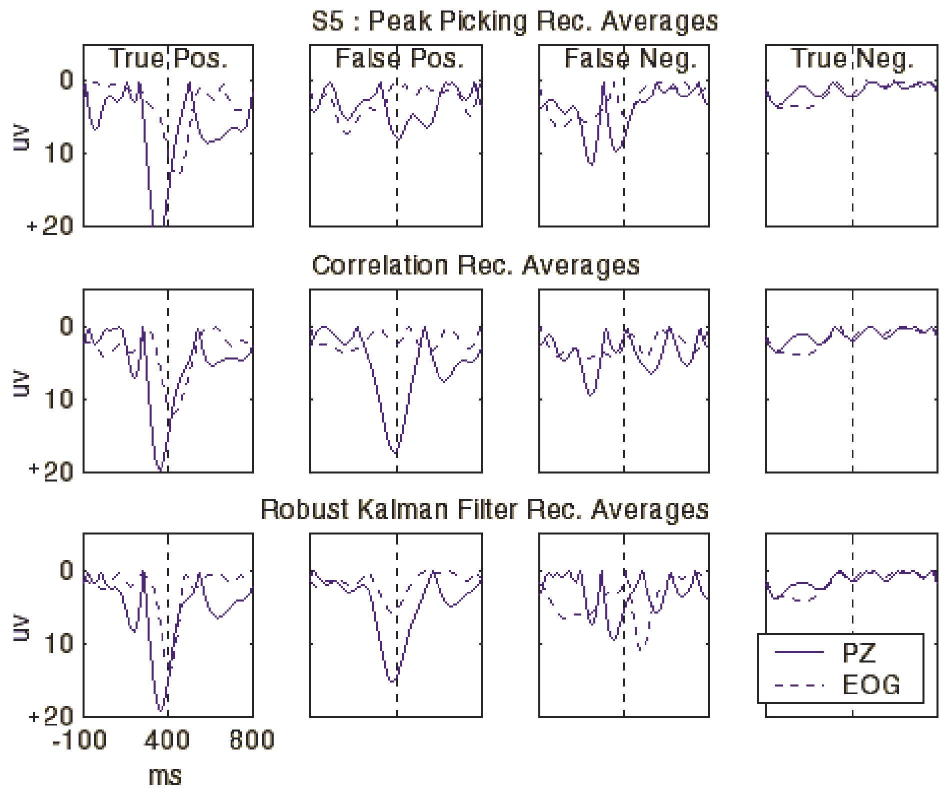

S3 : Algorithm Threshold ROC Curves
True Positive Probability
False Positive Probability
0
0.2
0.4
0.6
0.8
1
0
0.5
1
Pk Pick
Corr.
K. Filter

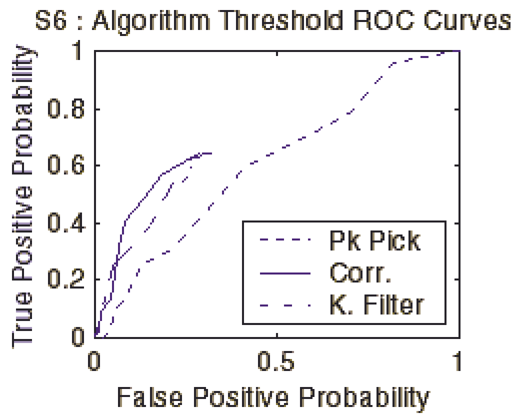

S3 : Peak Picking Rec. Averages
True Pos.
False Pos.
False Neg.
True Neg.
uv
0
10
+20
Correlation Rec. Averages
Robust Kalman Filter Rec. Averages
-100
400
800
ms
PZ
EOG

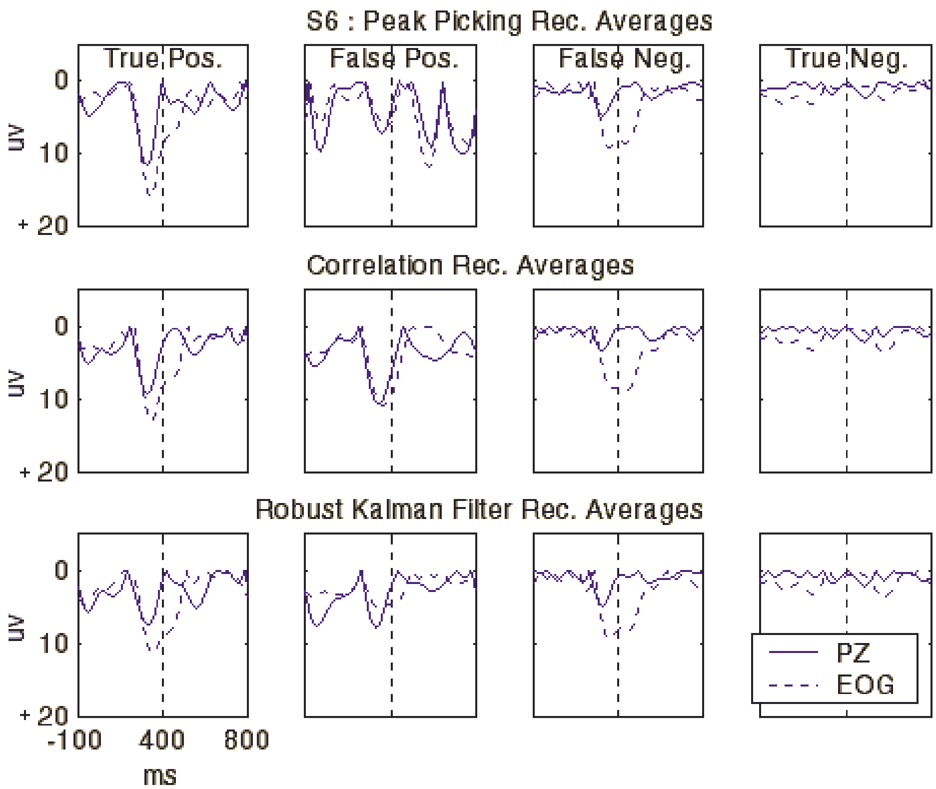

S4 : Algorithm Threshold ROC Curves
True Positive Probability
False Positive Probability
Pk Pick
Corr.
K. Filter
0
0.2
0.4
0.6
0.8
1
0.5

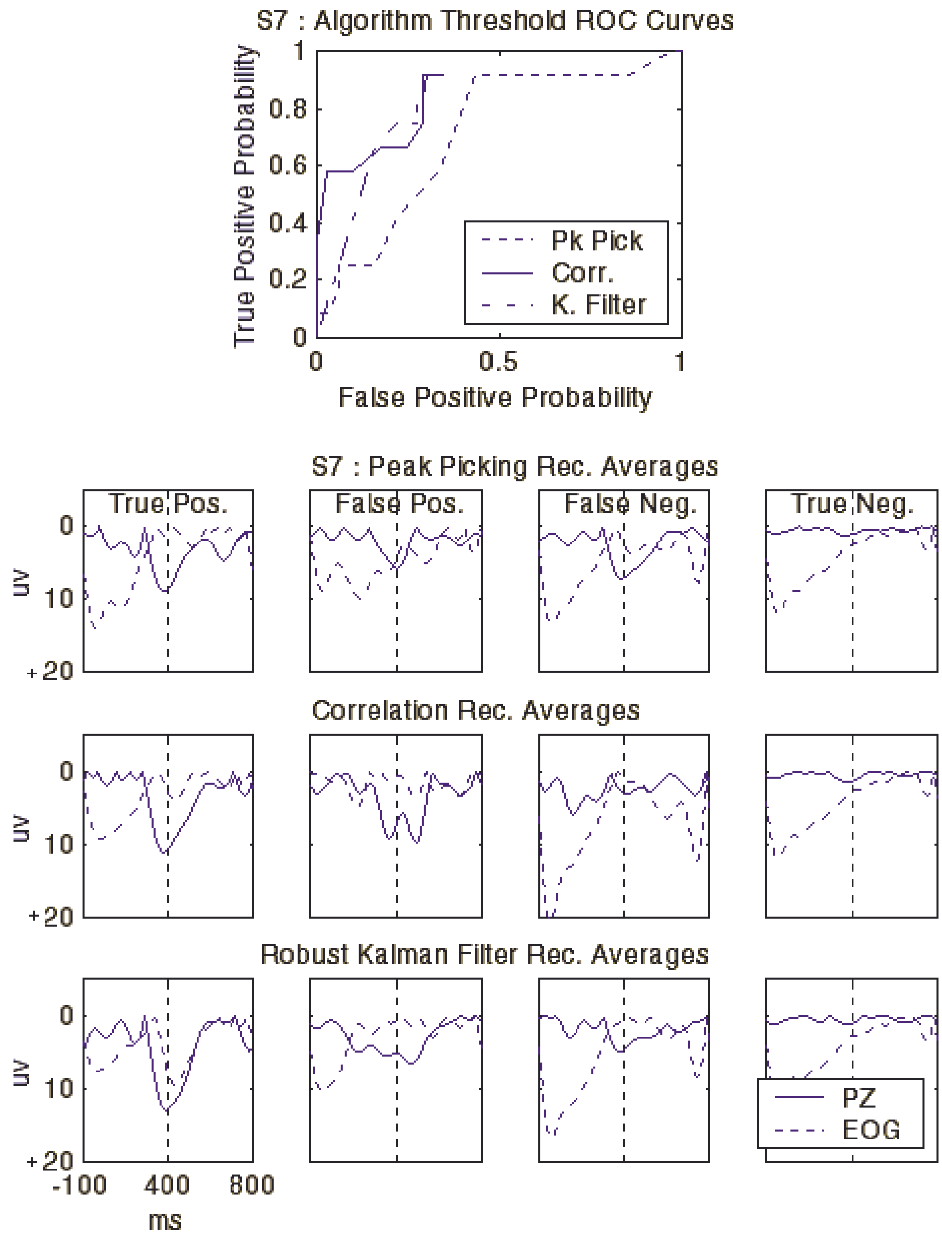

S4 : Peak Picking Rec. Averages
True Pos.
False Pos.
False Neg.
True Neg.
Correlation Rec. Averages
Robust Kalman Filter Rec. Averages
uv
0
10
+20
-100
400
800
ms
PZ
EOG

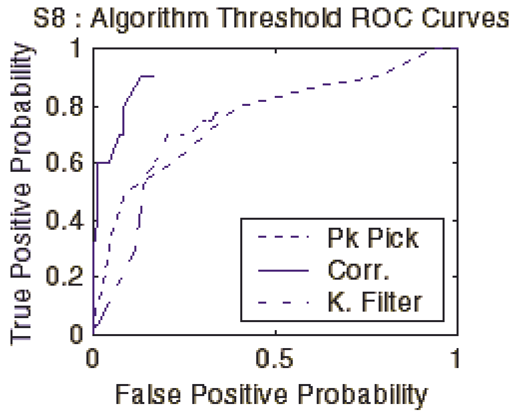

S5 : Algorithm Threshold ROC Curves
True Positive Probability
False Positive Probability
0
0.2
0.4
0.6
0.8
1
0
0.5
1
Pk Pick
Corr.
K. Filter

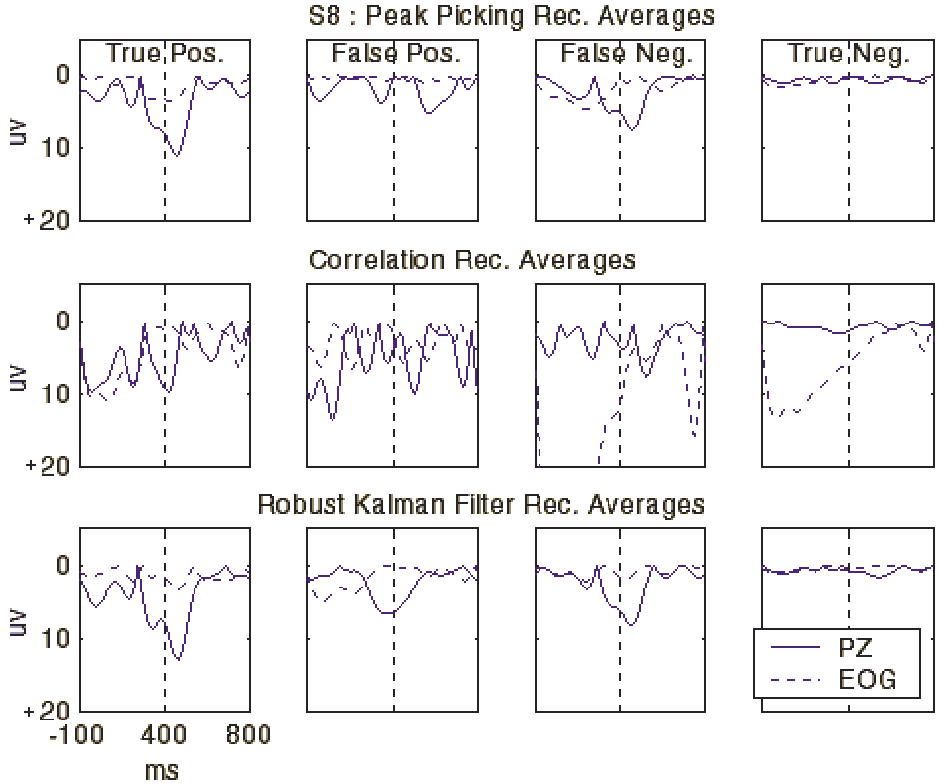

S5 : Peak Picking Rec. Averages
True Pos.
False Pos.
False Neg.
True Neg.
Correlation Rec. Averages
Robust Kalman Filter Rec. Averages
uv
0
10
+20
-100
400
800
ms
PZ
EOG


S6 : Algorithm Threshold ROC Curves
True Positive Probability
False Positive Probability
0
0.2
0.4
0.6
0.8
1
0
0.5
1
Pk Pick
Corr.
K. Filter


S6 : Peak Picking Rec. Averages
True Pos.
False Pos.
False Neg.
True Neg.
Correlation Rec. Averages
Robust Kalman Filter Rec. Averages
uv
0
10
+ 20
-100
400
800
ms
PZ
EOG


S7 : Algorithm Threshold ROC Curves
True Positive Probability
False Positive Probability
0
0.2
0.4
0.6
0.8
1
0
0.5
1
Pk Pick
Corr.
K. Filter
S7 : Peak Picking Rec. Averages
True Pos.
False Pos.
False Neg.
True Neg.
Correlation Rec. Averages
Robust Kalman Filter Rec. Averages
uv
0
10
+20
-100
400
800
ms
PZ
EOG


S8 : Algorithm Threshold ROC Curves
True Positive Probability
False Positive Probability
Pk Pick
Corr.
K. Filter
0
0.2
0.4
0.6
0.8
1
0.5


S8 : Peak Picking Rec. Averages
True Pos.
False Pos.
False Neg.
True Neg.
Correlation Rec. Averages
Robust Kalman Filter Rec. Averages
uv
0
10
+20
-100
400
800
ms
PZ
EOG

Appendix B

Recognition for Artifact

Data was obtained from a subject performing several different kinds of artifacts. This data has been used for off-line analysis over three different algorithms: peak picking, correlation, and an algorithm that use a robust Kalman filter for preprocessing data and correlation for recognition. Examples taken from the artifact data are shown in Figures B-1, B-2, B-3, and B-4. The averages for each algorithm result category (true negative and false positive) are shown for each kind of artifact used. It was assumed that all data represented true negatives, since no stimuli were presented in order to obtain a P3.

The number of trials used to produce each set of averages for peaking picking, correlation, and the algorithm using the robust Kalman filter are shown in Tables B-1, B-2, and B-3 (respectively). The solid line in each average plot represents the appropriate type of trial average at site PZ, while the dotted line represents the average vertical eye movement voltage. All averages are filtered at 30 Hz for display purposes.

The data from heavy breathing is shown in Chapter 4. Please see this chapter for a complete discussion of results.

Table B-1: The number of trials used for each of the specific artifact averages in the peak picking algorithm.

	True Neg.	*False Pos.*	*Percentage Correct*
Breathing	*159*	*8*	*95*
Chewing	*497*	*81*	*86*
Foot Mv.	*373*	*56*	*87*
Forehead Mv.	*569*	*195*	*74*
Horizontal EOG	*694*	*52*	*93*
Jaw Clenching	*467*	*18*	*96*
Talking	*641*	*111*	*85*
Vertical EOG	*619*	*111*	*85*

Table B-2: The number of trials used for each of the specific artifact averages in the correlation algorithm.

	True Neg.	*False Pos.*	*Percentage Correct*
Breathing	*162*	*5*	*97*
Chewing	*545*	*33*	*94*
Foot Mv.	*414*	*15*	*97*
Forehead Mv.	*734*	*30*	*96*
Horizontal EOG	*701*	*45*	*94*
Jaw Clenching	*470*	*15*	*97*
Talking	*716*	*36*	*95*
Vertical EOG	*678*	*52*	*93*

Table B-3: The number of trials used for each of the specific artifact averages in the algorithm using the robust Kalman filter for preprocessing.

	True Neg.	*False Pos.*	*Percentage Correct*
Breathing	*164*	*3*	*98*
Chewing	*547*	*31*	*95*
Foot Mv.	*405*	*24*	*94*
Forehead Mv.	*713*	*51*	*93*
Horizontal EOG	*690*	*56*	*92*
Jaw Clenching	*457*	*28*	*94*
Talking	*701*	*51*	*93*
Vertical EOG	*669*	*61*	*92*

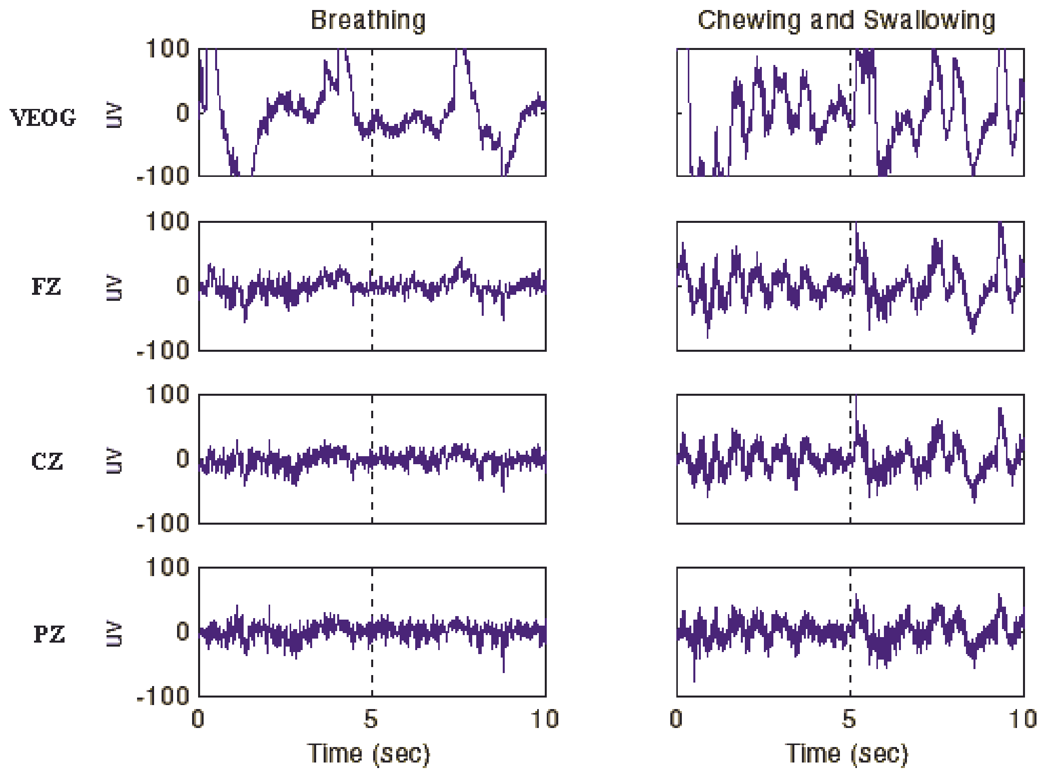


Figure B-1: Examples of artifacts obtained from heavy breathing and chewing. In addition to the heavy breathing, the subject tended to move his/her whole body frame while inhaling and exhaling. This accounts for the large movements seen in the vertical eye movement channel.

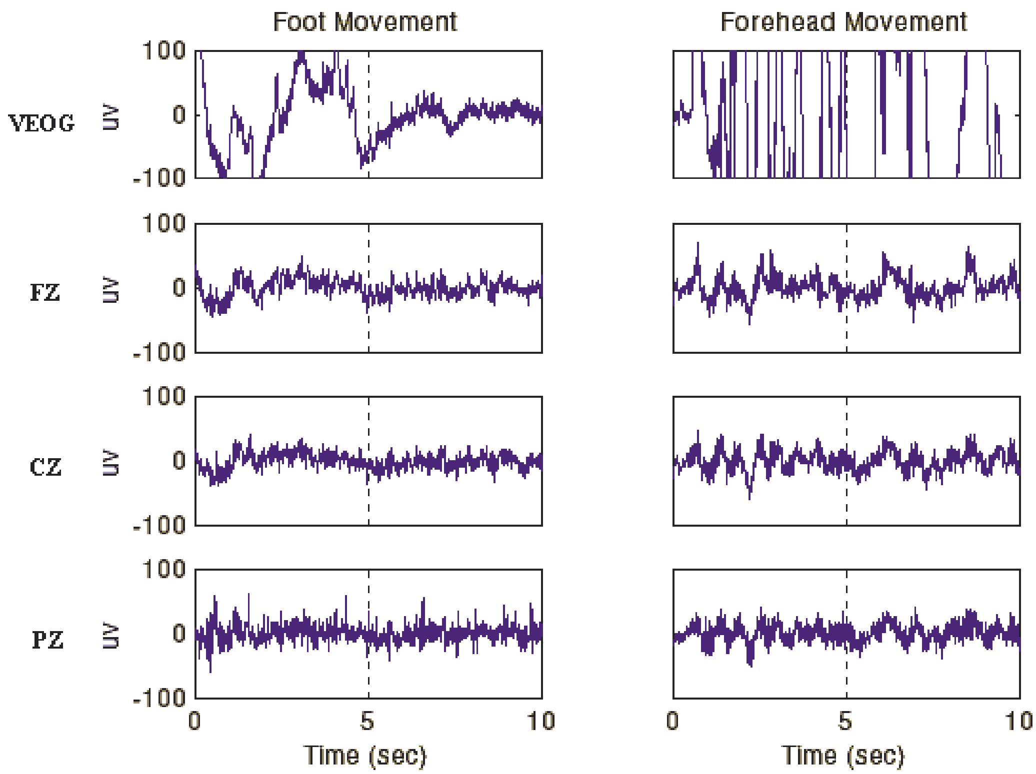


Figure B-2: Examples of artifact data for foot and forehead movements.

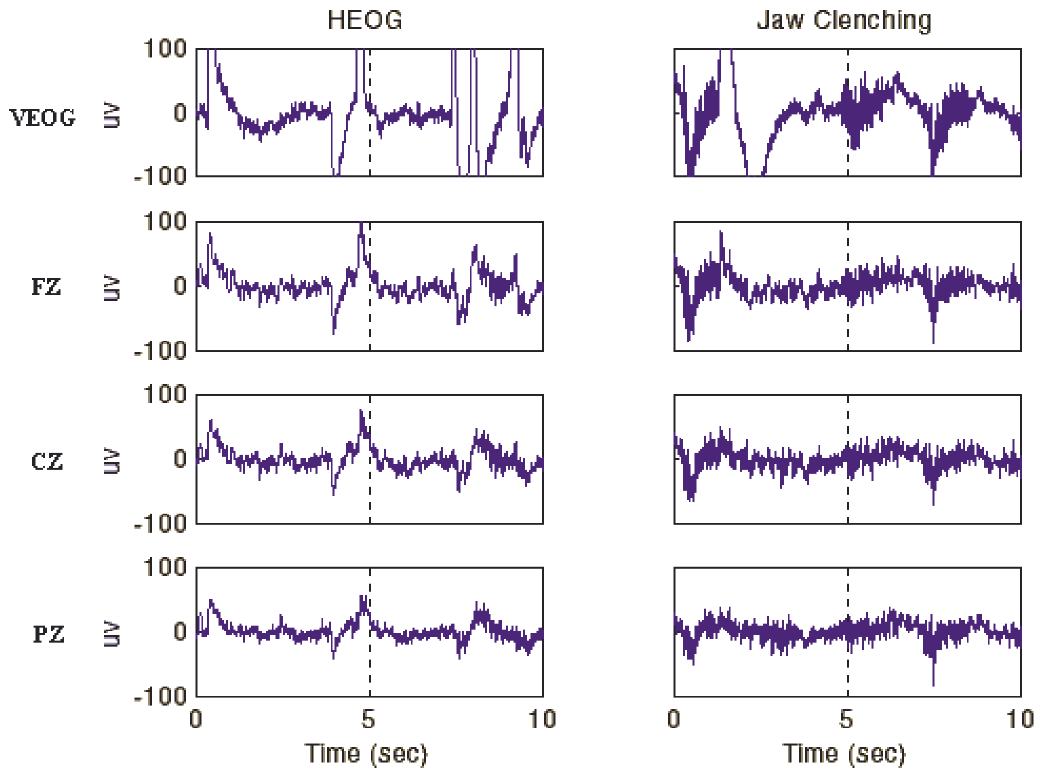


Figure B-3: Examples of artifact recordings containing horizontal eye movements and jaw clenching.

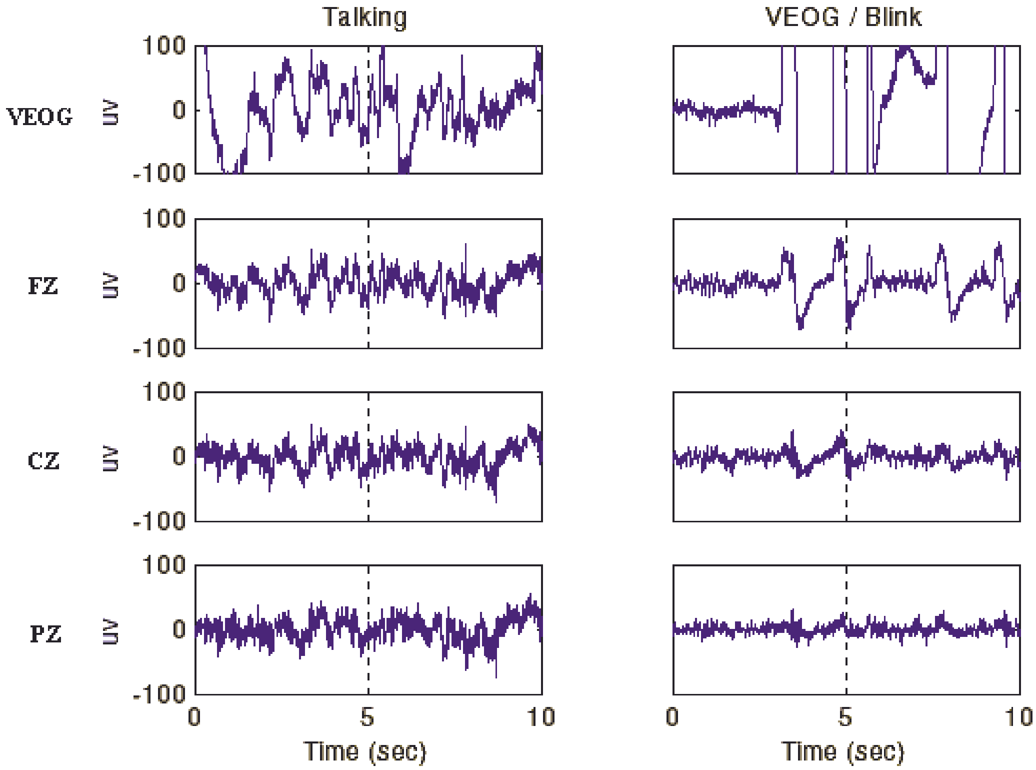


Figure B-4: Examples of artifact recordings of talking, blinking, and vertical eye movements.

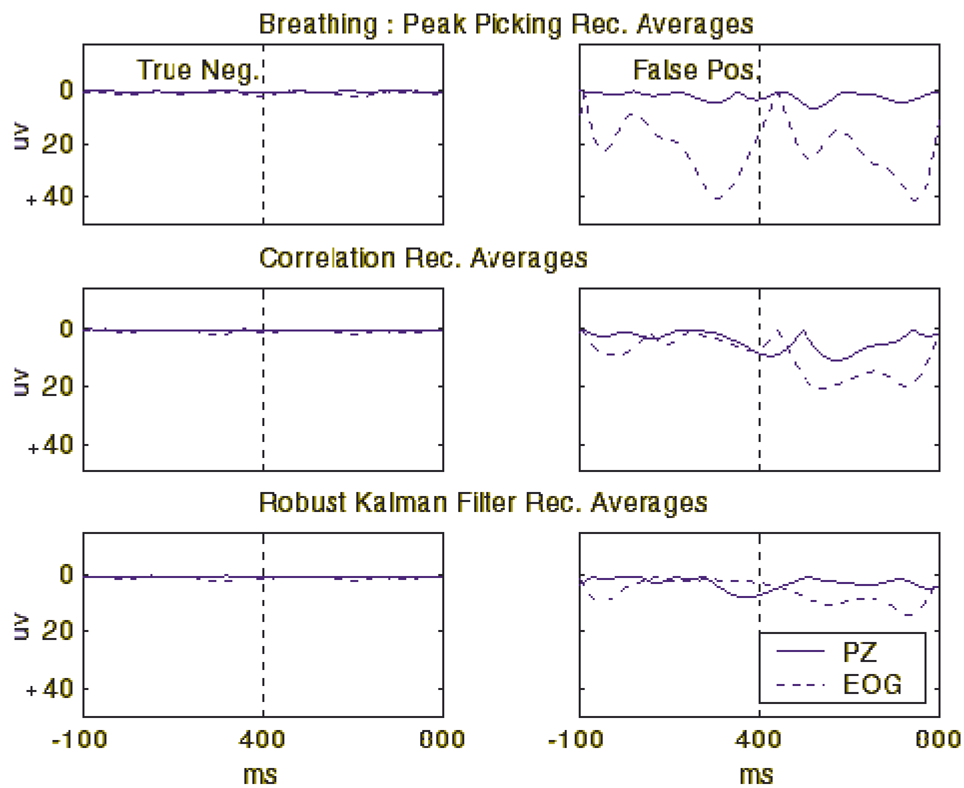


Figure B-5: True negative and False positive recognition averages while the subject breathed heavily.

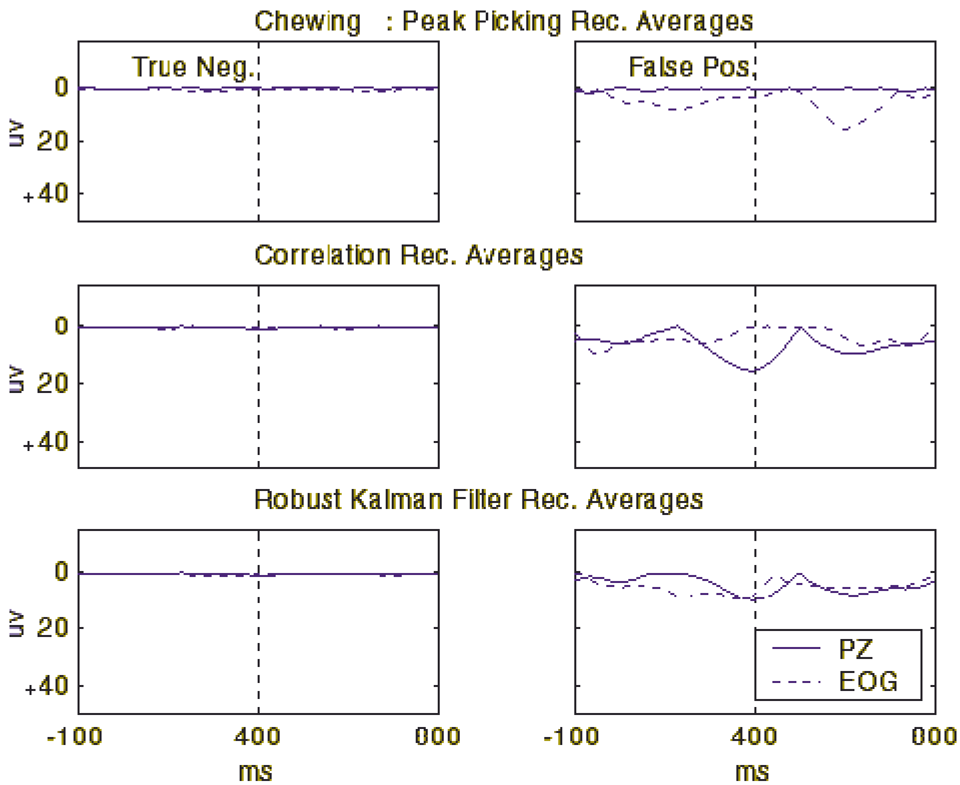


Figure B-6: True negative and false positive recognition averages while the subject chewed and swallowed.

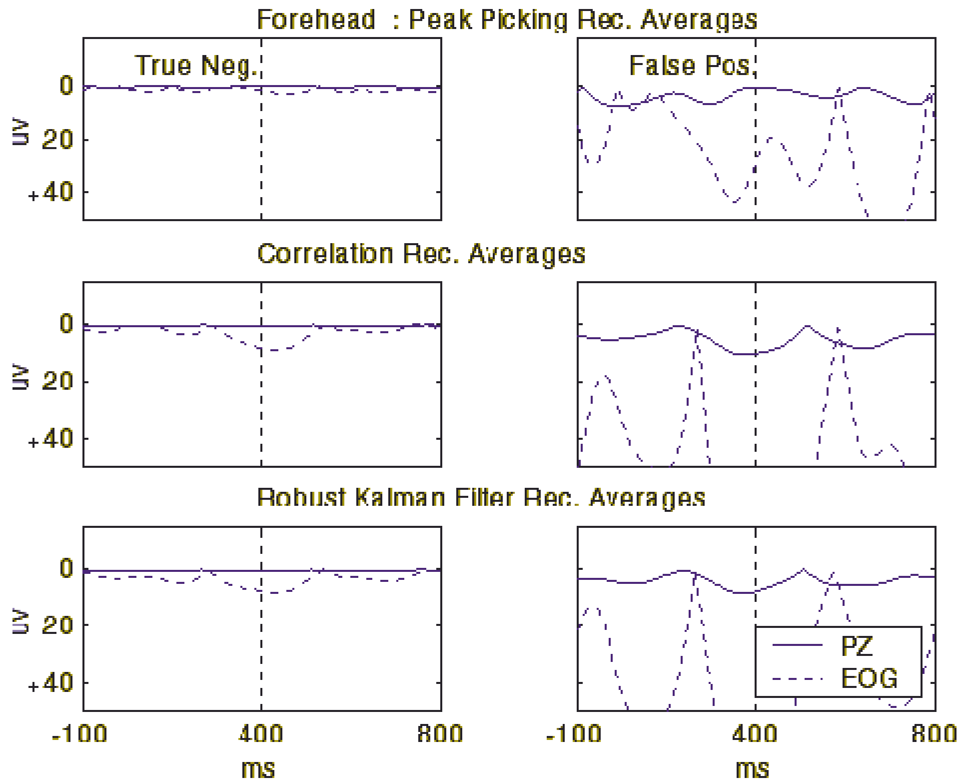


Figure B-7: True negative and false positive recognition averages for forehead movements.

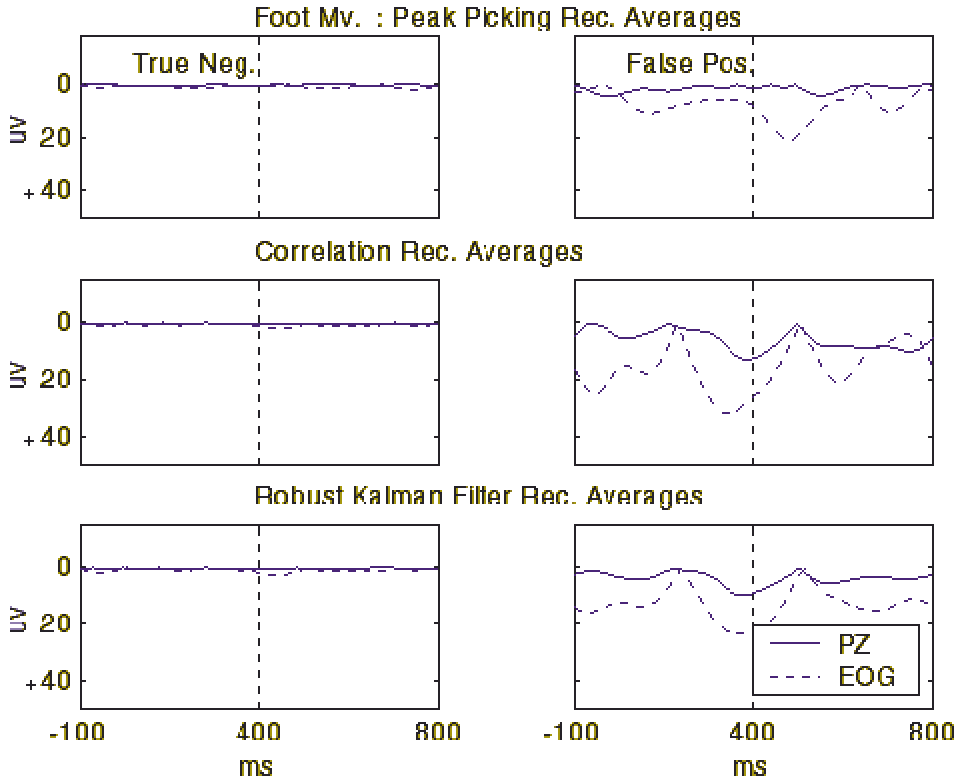


Figure B-8: True negative and false positive recognition averages while the subject moved one or both feet.

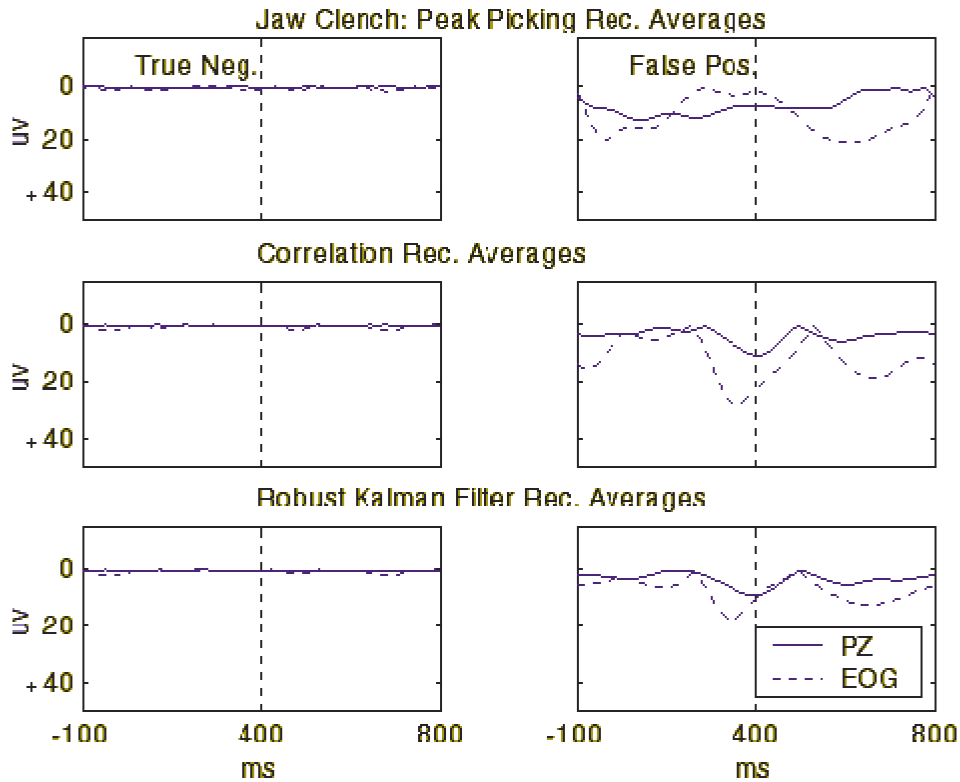


Figure B-9: True negative and false positive recognition averages while the subject performed jaw clenching.

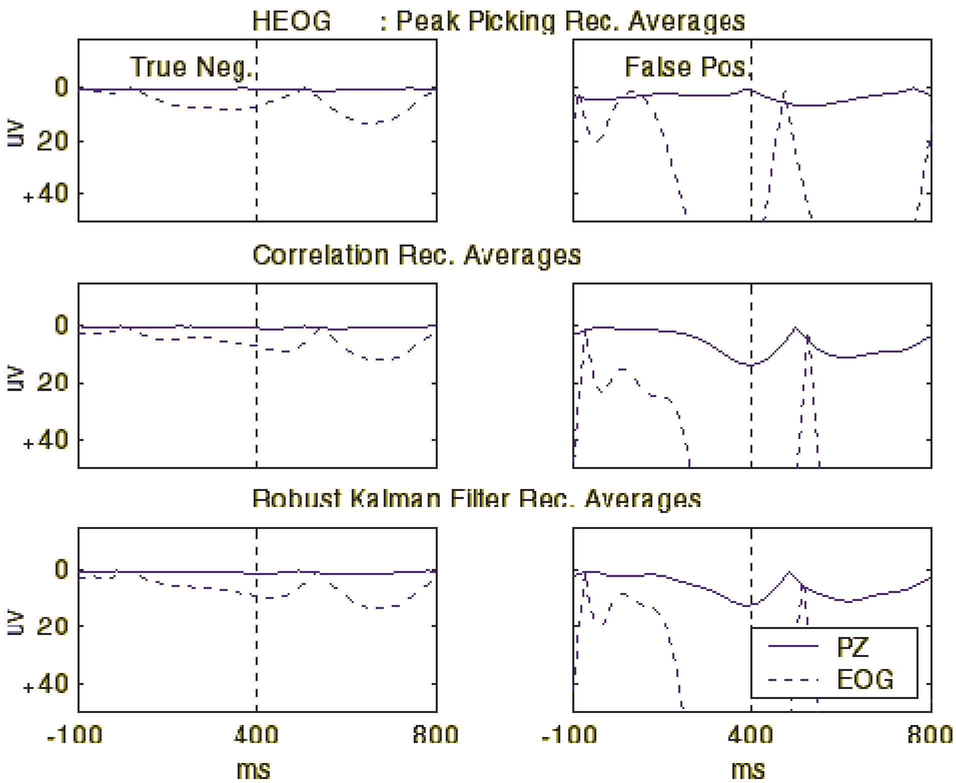


Figure B-10: True negative and false positive recognition averages for horizontal eye movements.

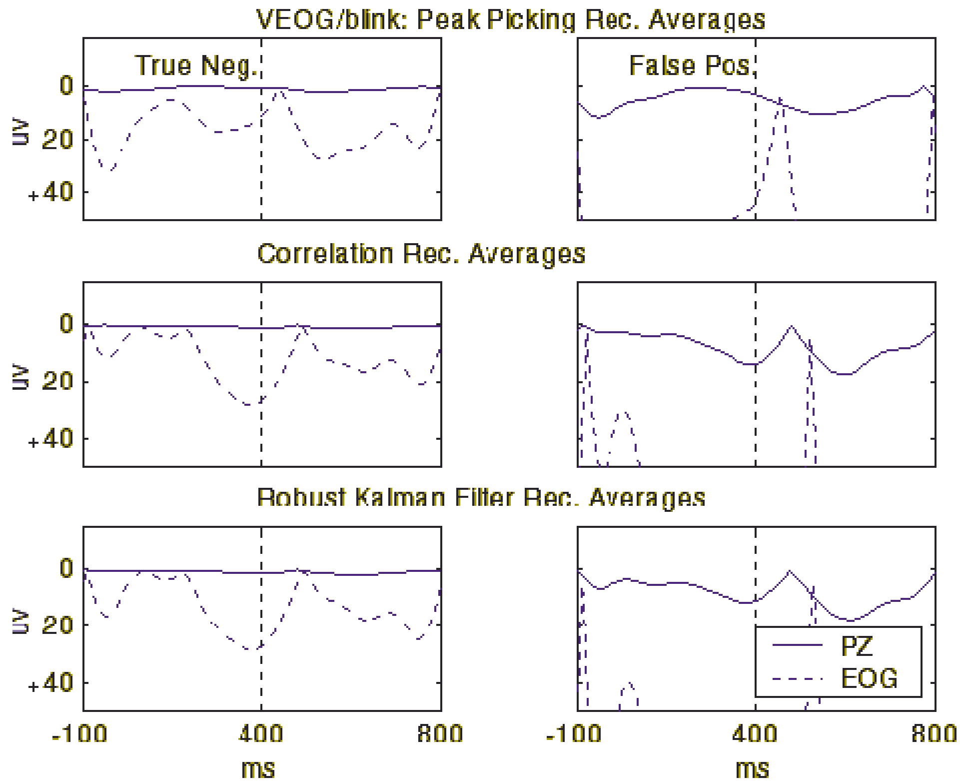


Figure B-11: True negative and false positive recognition averages for vertical eye movements and blinking.

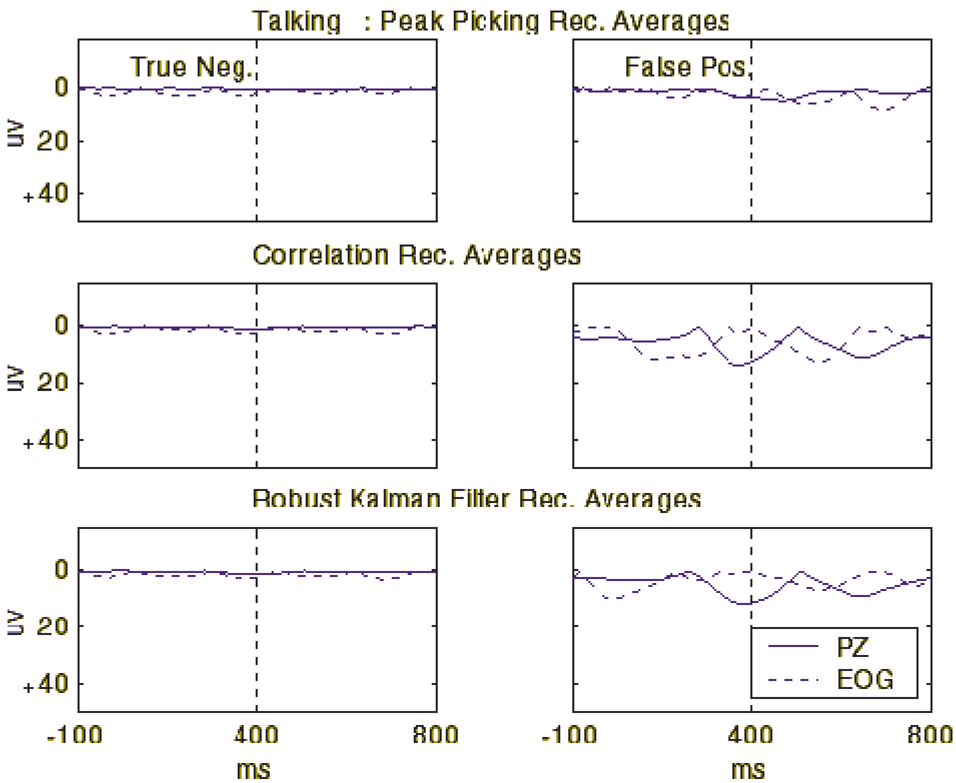


Figure B-12: True negative and false positive recognition averages during talking.

Appendix C
Individual Subject Data for the Virtual Apartment Experiment

Data was obtained for nine subjects in an environmental control experiment. While overall results are shown in Chapter 6, results from individual subjects are shown in this appendix. Please see the main chapter for a discussion of the experiment and main results. The great degree of variance between different subjects becomes apparent by looking through the results for each subject. Some subjects show prototypical P3 signals while others do not.

In order to demonstrate the variability between trials, the first 40 trials for the Goal (where a P3 signal should occur) and non-Goal (where a P3 signal should not occur) light flashes are shown for each of the three main conditions and for each subject. The main conditions are VR, SCREEN, and FIXED DISPLAY. Grand averages for goal and non-goal signals are also shown for each subject. These averages have not been filtered at 30 Hz for display purposes and so look noisier than the filtered averages from previous chapters. In addition, the data is shown without preprocessing to remove eye artifacts, so that the extent of eye artifacts among subjects may be seen. The total number of trials in each average is shown in Table C-1. Goal trials occur approximately 20% of the time.

An indication of the average number of tries per successful control is given in bar charts of tries per object for each subject. These charts show the median number of tries/misses as well as the upper/lower quartiles and any data outliers. Tables C-2, C-3, and C-4 show the number of goals (true positives in the application) achieved by each subject as this has an effect on each bar chart. Tables C-5, C-6, and C-7 show the total number of misses (false positives in the application) for each subject.

Table C-1: The number of trials used in subject averages.

	Training		*Fixed Display*		*Monitor*		*VR*	
	Goal	*Non-Goal*	*Goal*	*Non-Goal*	*Goal*	*Non-Goal*	*Goal*	*Non-Goal*
Subject 1	*57*	*245*	*34*	*132*	*42*	*140*	*49*	*140*
Subject 2	*63*	*239*	*37*	*137*	*30*	*133*	*34*	*144*
Subject 3	*62*	*239*	*49*	*143*	*41*	*128*	*45*	*144*
Subject 4	*66*	*235*	*40*	*148*	*34*	*143*	*35*	*151*
Subject 5	*69*	*234*	*29*	*149*	*40*	*145*	*40*	*169*
Subject 6	*61*	*240*	*41*	*164*	*23*	*106*	*44*	*160*
Subject 7	*51*	*251*	*28*	*141*	*37*	*135*	*41*	*154*
Subject 8	*56*	*244*	*37*	*137*	*45*	*145*	*27*	*120*
Subject 9	*63*	*238*	*51*	*173*	*34*	*126*	*37*	*173*

Table C-2: The number of goals achieved by each subject for the FIXED DISPLAY condition.

	Light	*TV*	*Stereo*	*Bye*	*Hi*	*Total*
Subject 1	*3*	*3*	*1*	*2*	*2*	*11*
Subject 2	*7*	*1*	*5*	*4*	*2*	*19*
Subject 3	*5*	*6*	*3*	*3*	*1*	*18*
Subject 4	*3*	*3*	*3*	*5*	*2*	*16*
Subject 5	*2*	*0*	*2*	*1*	*0*	*5*
Subject 6	*3*	*1*	*1*	*2*	*0*	*7*
Subject 7	*3*	*3*	*3*	*3*	*1*	*13*
Subject 8	*5*	*1*	*1*	*2*	*2*	*11*
Subject 9	*2*	*0*	*0*	*0*	*1*	*3*

Table C-3: The number of goals achieved by each subject for the MONITOR condition.

	Light	TV	Stereo	Bye	Hi	Total
Subject 1	*3*	*4*	*4*	*5*	*3*	*19*
Subject 2	*5*	*5*	*3*	*7*	*2*	*22*
Subject 3	*6*	*6*	*6*	*4*	*4*	*26*
Subject 4	*3*	*2*	*2*	*6*	*1*	*14*
Subject 5	*2*	*0*	*0*	*2*	*1*	*5*
Subject 6	*1*	*0*	*1*	*1*	*1*	*4*
Subject 7	*5*	*3*	*2*	*2*	*1*	*13*
Subject 8	*3*	*2*	*4*	*2*	*1*	*12*
Subject 9	*1*	*0*	*1*	*1*	*1*	*4*

Table C-4: The number of goals achieved by each subject for the VR condition.

	Light	TV	Stereo	Bye	Hi	Total
Subject 1	*5*	*3*	*4*	*2*	*2*	*16*
Subject 2	*6*	*4*	*8*	*6*	*1*	*25*
Subject 3	*4*	*4*	*6*	*3*	*2*	*19*
Subject 4	*2*	*3*	*2*	*5*	*1*	*13*
Subject 5	*2*	*1*	*1*	*2*	*1*	*7*
Subject 6	*2*	*1*	*1*	*1*	*2*	*7*
Subject 7	*2*	*0*	*1*	*3*	*5*	*11*
Subject 8	*0*	*1*	*1*	*3*	*0*	*5*
Subject 9	*0*	*4*	*1*	*4*	*1*	*10*

Table C-5: The number of false positives incurred by each subject for the FIXED DISPLAY condition.

	Light	TV	Stereo	Bye	Hi	Total
Subject 1	*6*	*3*	*2*	*1*	*2*	*14*
Subject 2	*8*	*5*	*7*	*2*	*3*	*25*
Subject 3	*1*	*2*	*1*	*2*	*4*	*10*
Subject 4	*3*	*3*	*2*	*2*	*4*	*14*
Subject 5	*3*	*5*	*4*	*6*	*4*	*22*
Subject 6	*1*	*4*	*2*	*2*	*4*	*13*
Subject 7	*6*	*1*	*3*	*3*	*5*	*18*
Subject 8	*1*	*3*	*2*	*0*	*3*	*9*
Subject 9	*2*	*2*	*1*	*6*	*0*	*11*

Table C-6: The number of false positives incurred by each subject for the MONITOR condition.

	Light	*TV*	*Stereo*	*Bye*	*Hi*	*Total*
Subject 1	*5*	*1*	*3*	*3*	*1*	*13*
Subject 2	*6*	*5*	*3*	*3*	*12*	*29*
Subject 3	*4*	*4*	*2*	*0*	*3*	*13*
Subject 4	*2*	*1*	*4*	*3*	*6*	*16*
Subject 5	*2*	*11*	*1*	*6*	*7*	*27*
Subject 6	*1*	*3*	*2*	*1*	*1*	*8*
Subject 7	*4*	*3*	*6*	*0*	*5*	*18*
Subject 8	*1*	*2*	*2*	*3*	*5*	*13*
Subject 9	*3*	*2*	*3*	*0*	*1*	*9*

Table C-7: The number of false positives incurred by each subject for the VR condition.

	Light	*TV*	*Stereo*	*Bye*	*Hi*	*Total*
Subject 1	*4*	*4*	*3*	*2*	*0*	*13*
Subject 2	*2*	*6*	*6*	*10*	*6*	*30*
Subject 3	*2*	*4*	*0*	*0*	*5*	*11*
Subject 4	*2*	*2*	*4*	*0*	*5*	*13*
Subject 5	*5*	*3*	*1*	*3*	*0*	*12*
Subject 6	*2*	*2*	*2*	*4*	*4*	*14*
Subject 7	*5*	*5*	*3*	*1*	*1*	*15*
Subject 8	*0*	*1*	*1*	*3*	*0*	*5*
Subject 9	*0*	*4*	*1*	*4*	*1*	*10*

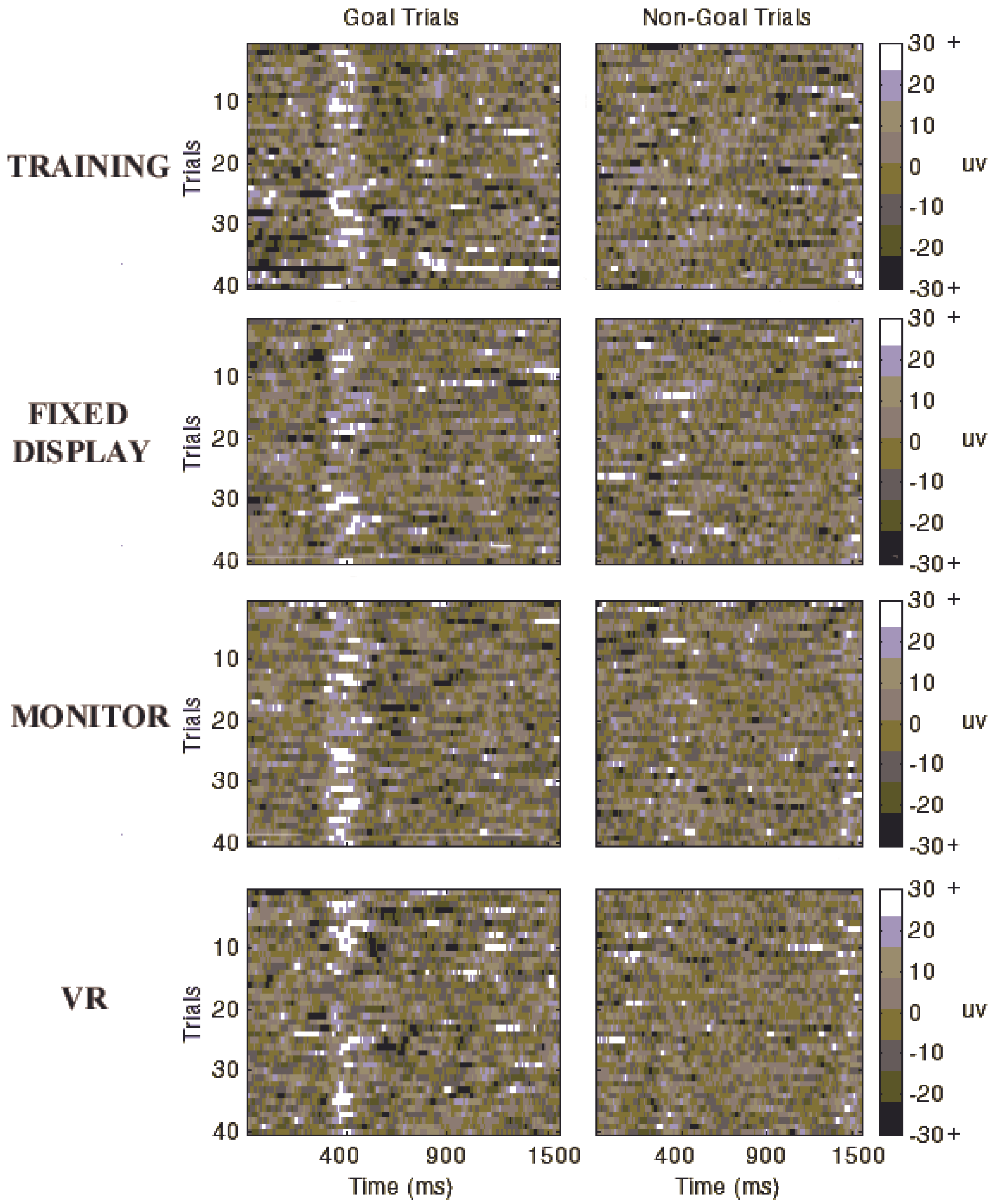


Figure C-1: Subject 1's goal and non-goal trials for each condition shown in the order the condition was presented.

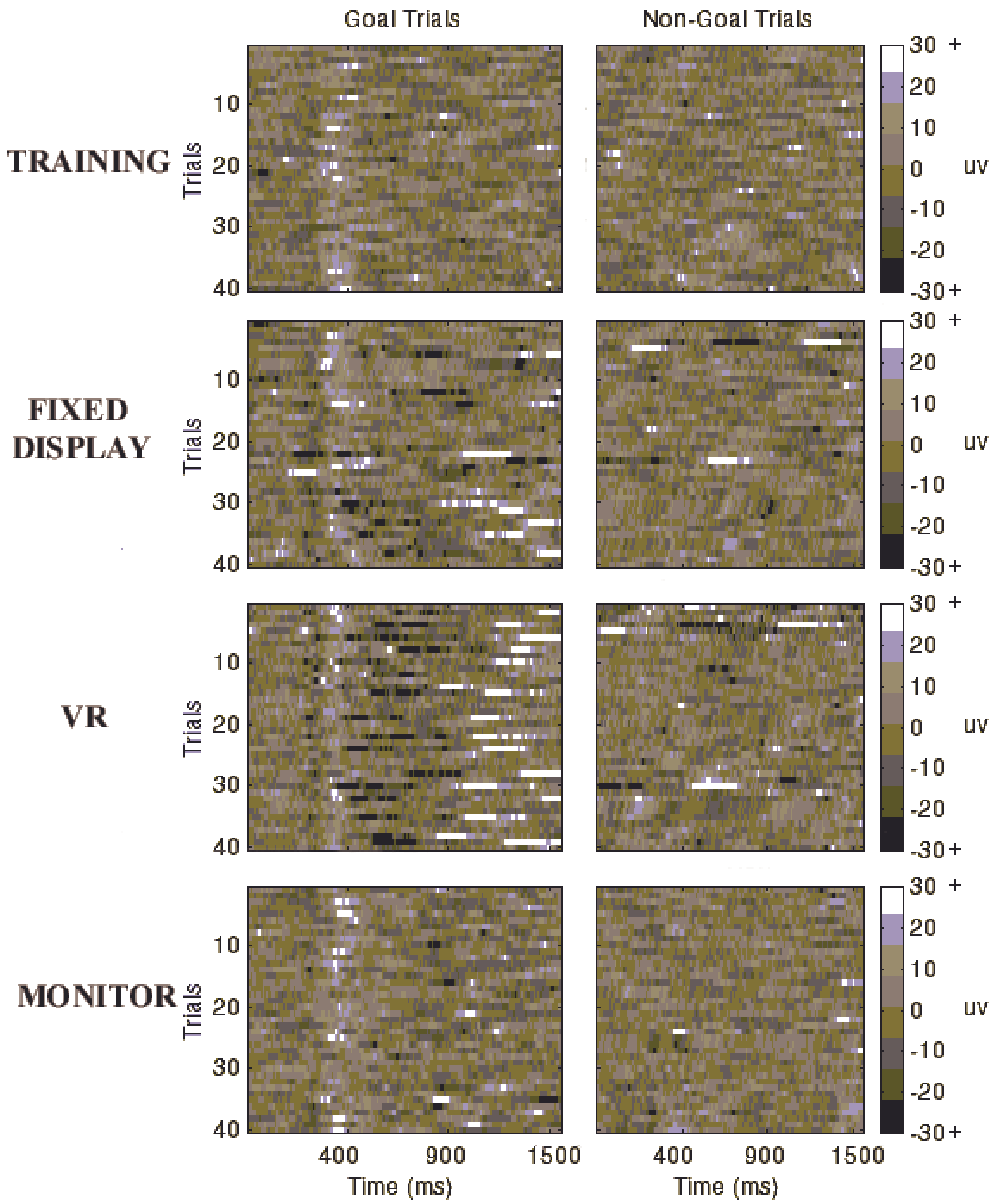


Figure C-2: Subject 2's goal and non-goal trials under different conditions and shown in the order of task presentation.

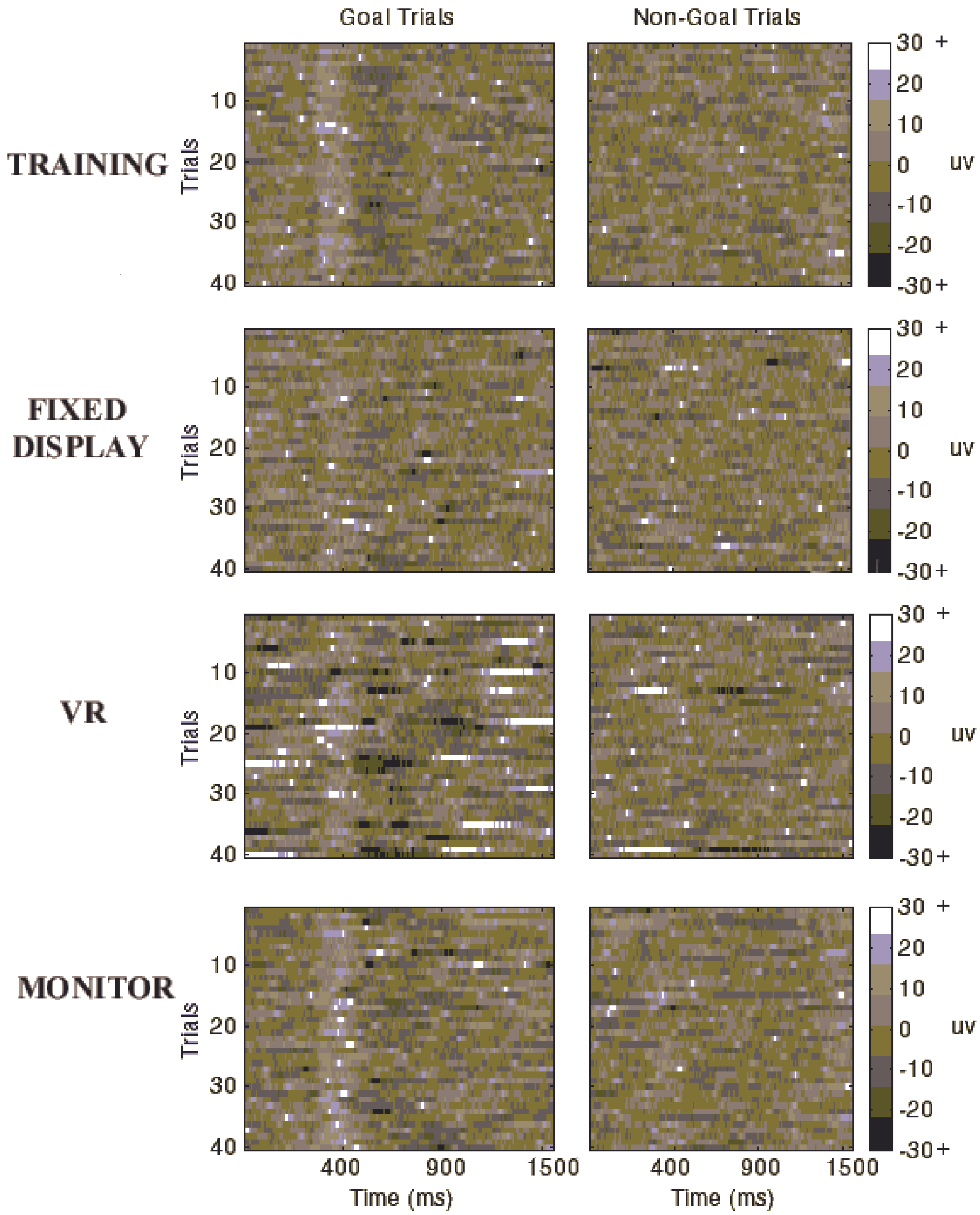


Figure C-3: Subject 3's goal and non-goal trials under different conditions and shown in the order of task presentation.

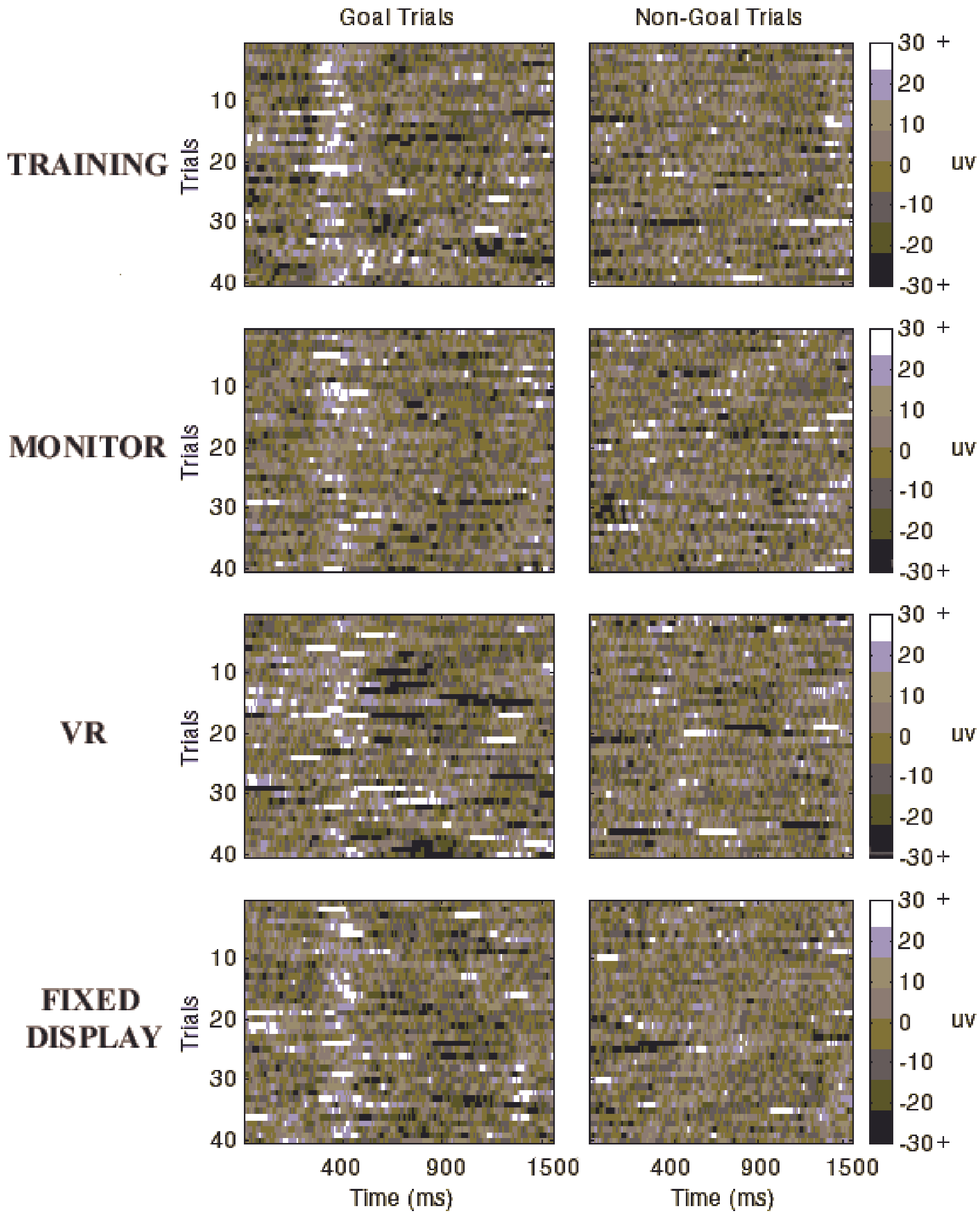


Figure C-4: Subject 4's goal and non-goal trials under different conditions and shown in the order of task presentation.

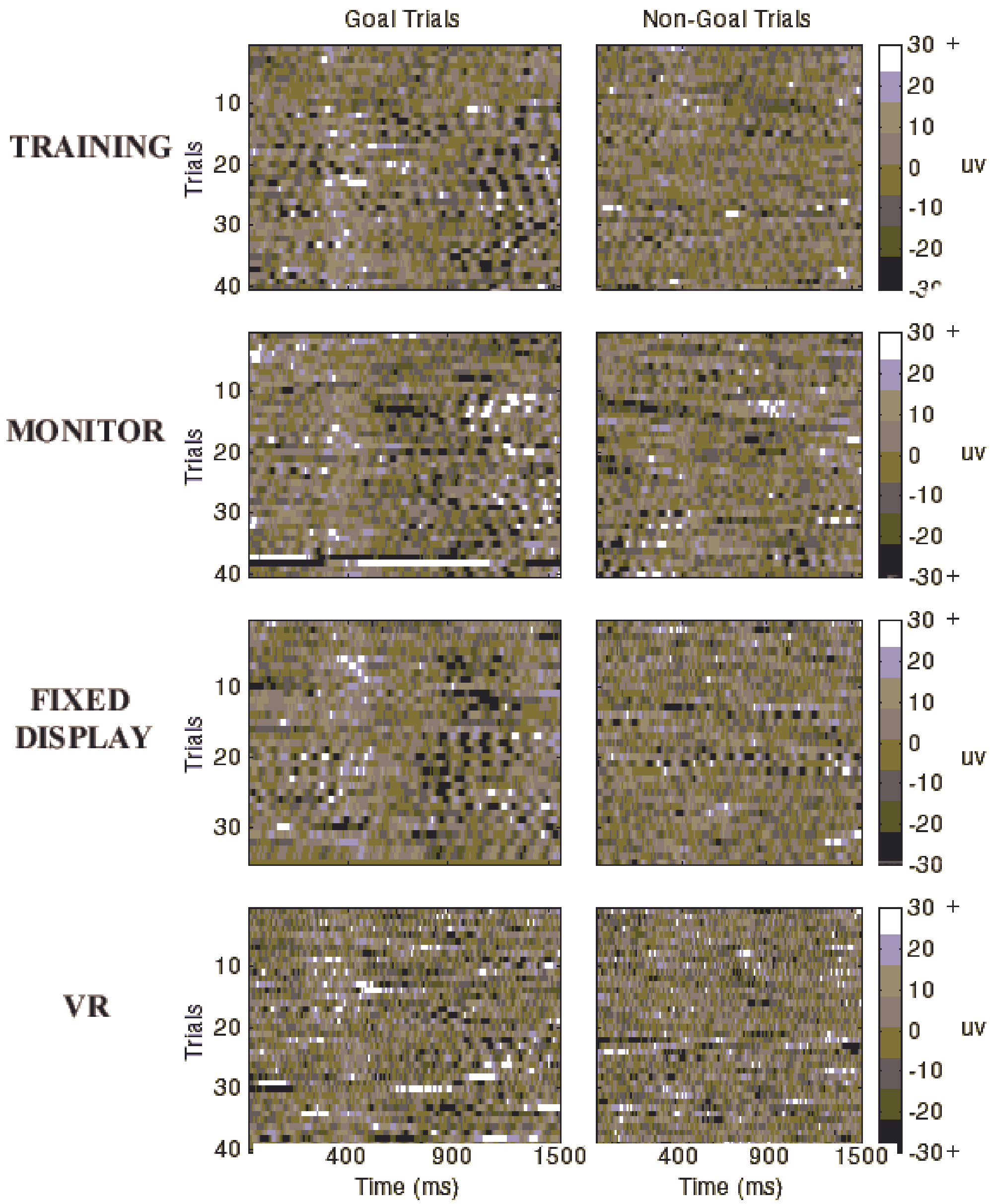


Figure C-5: Subject 5's goal and non-goal trials under different conditions and shown in the order of task presentation.

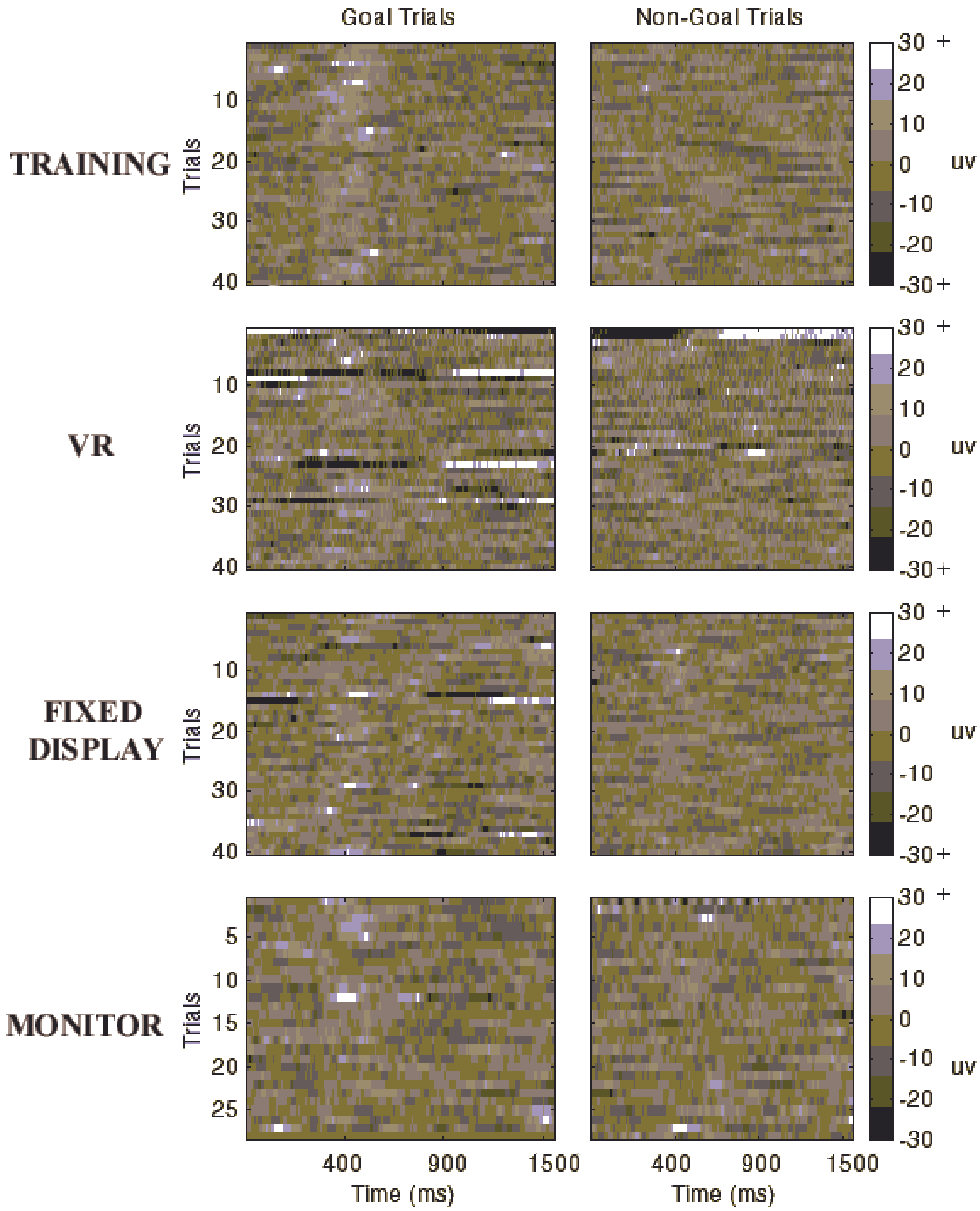


Figure C-6: Subject 6's goal and non-goal trials under different conditions and shown in the order of task presentation.

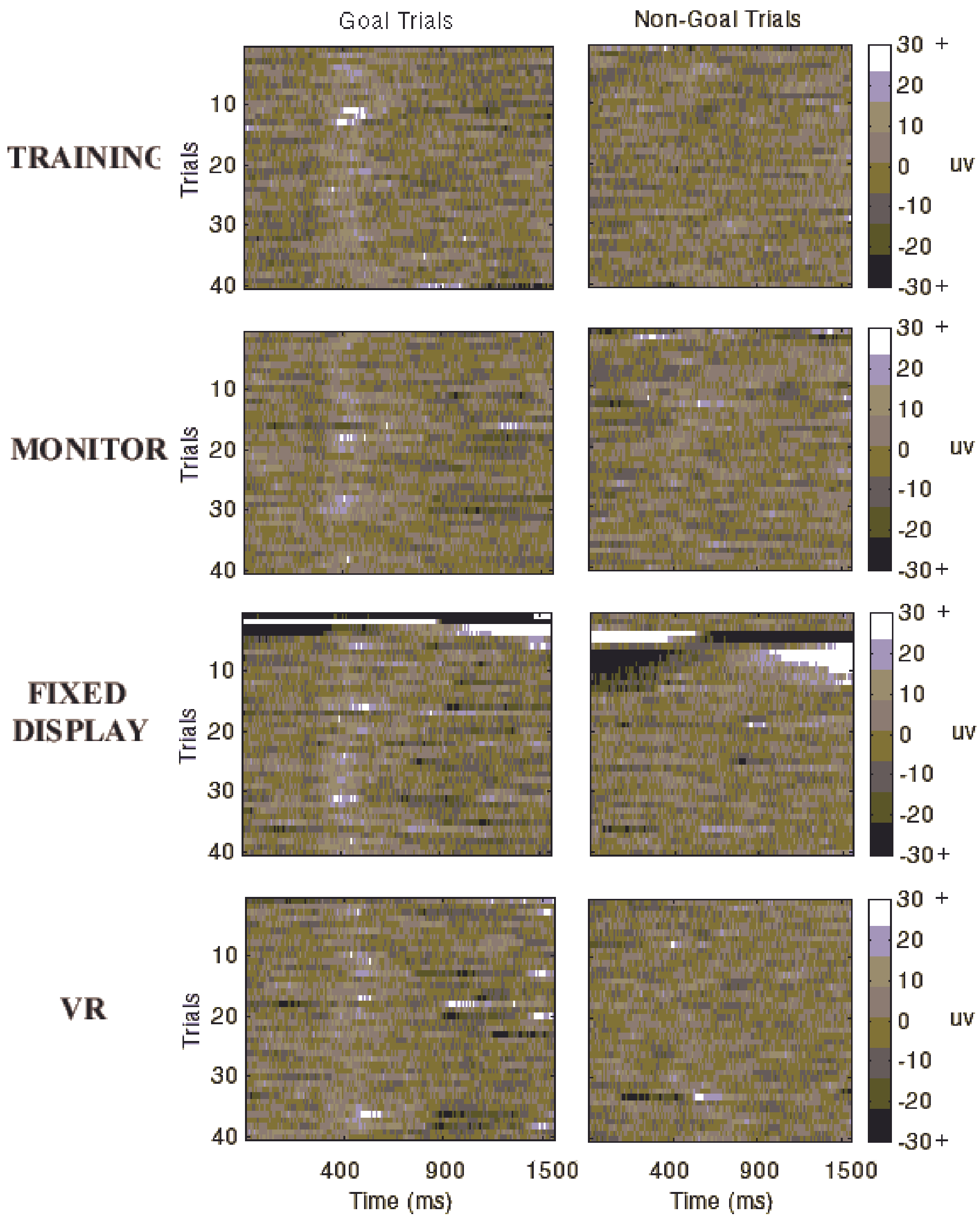


Figure C-7: Subject 7's goal and non-goal trials under different conditions and shown in the order of task presentation.

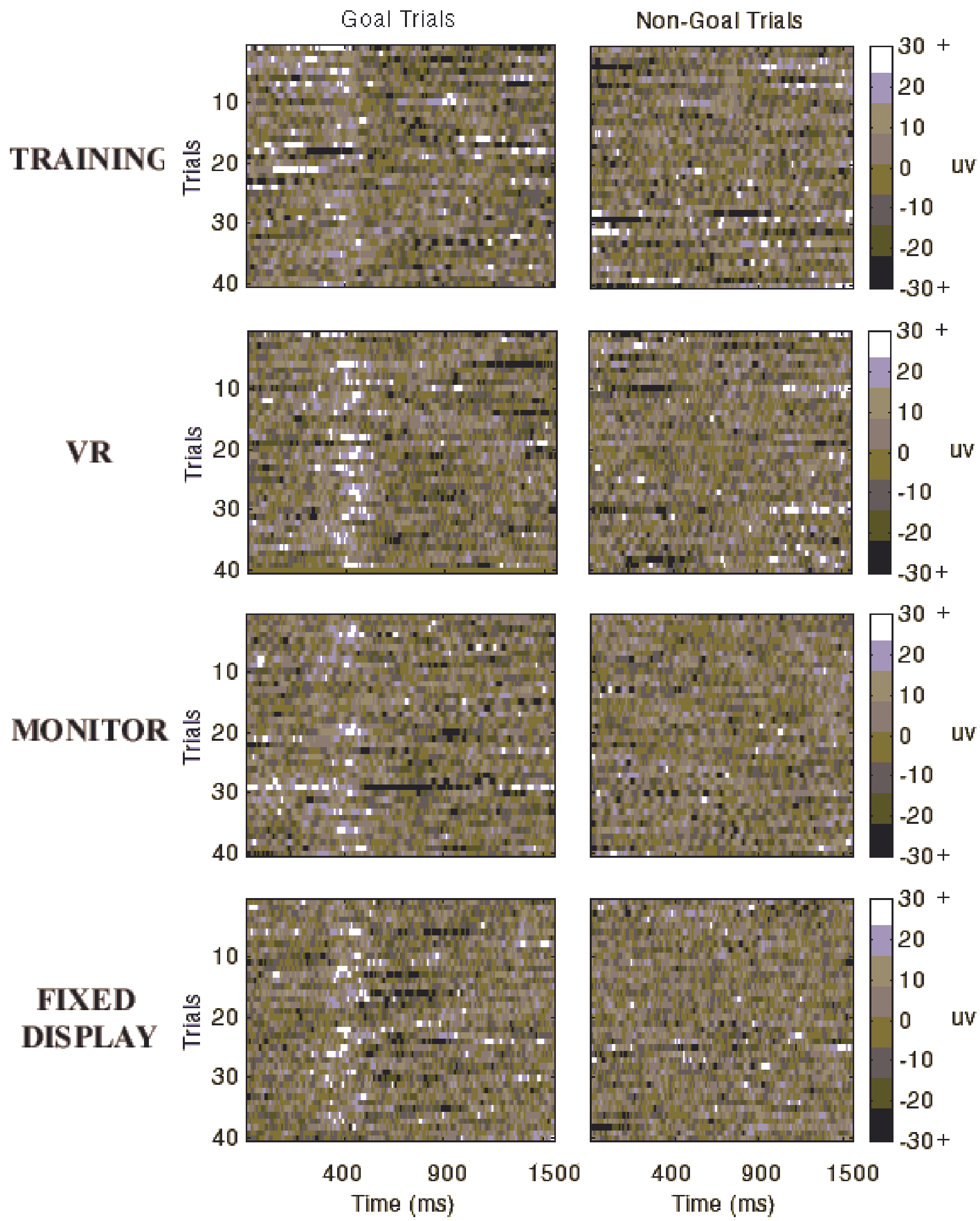


Figure C-8: Subject 8's goal and non-goal trials under different conditions and shown in the order of task presentation.

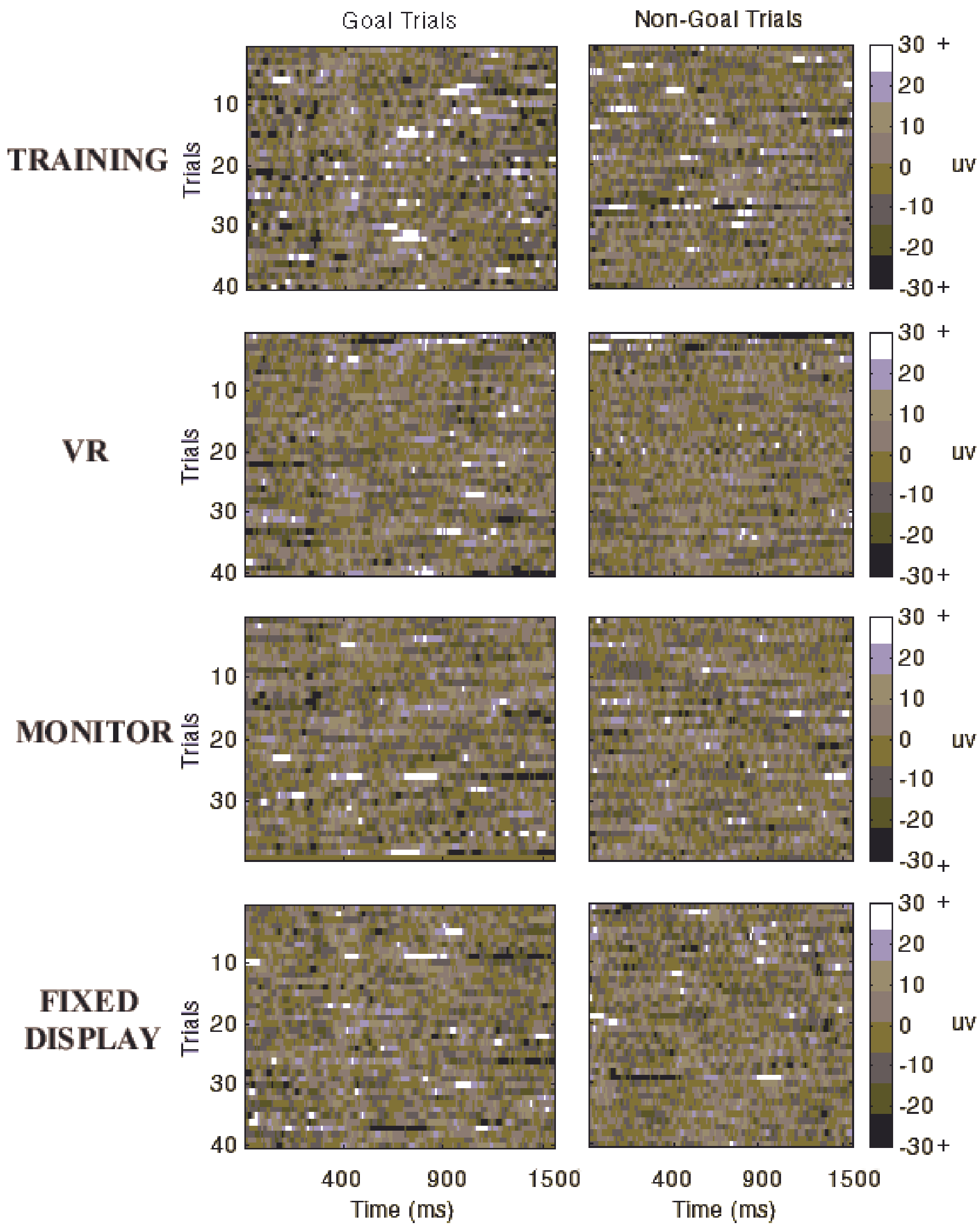


Figure C-9: Subject 9's goal and non-goal trials under different conditions and shown in the order of task presentation.

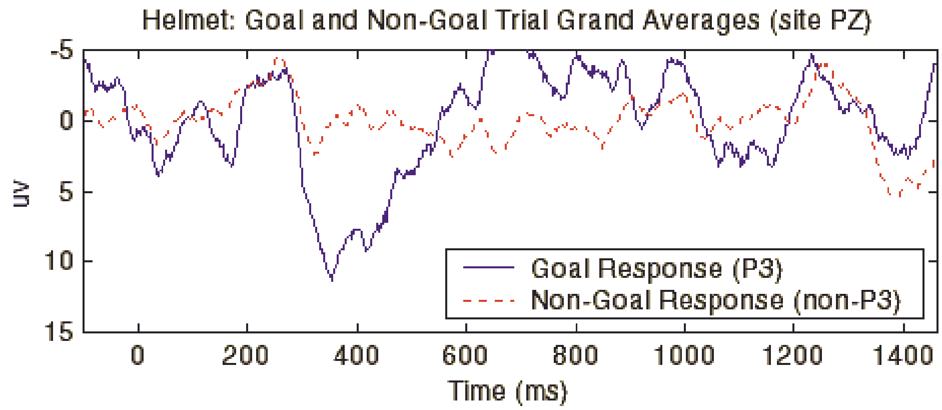


Figure C-10: Subject 1's goal and non-goal grand averages for the FIXED DISPLAY condition.

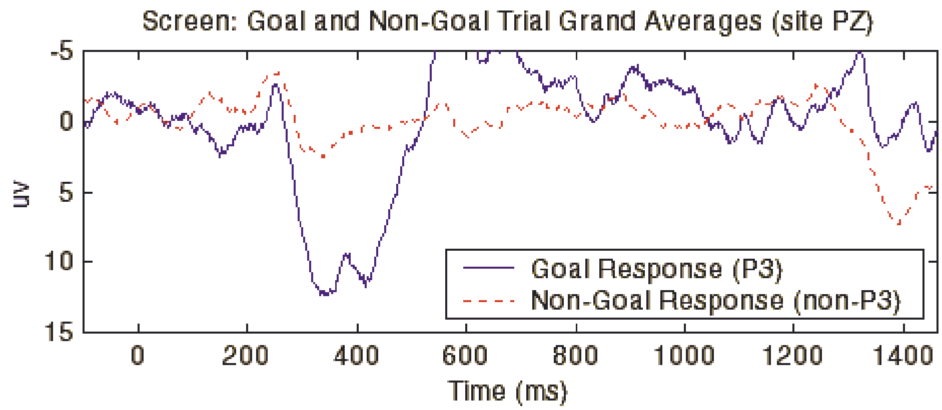


Figure C-11: Subject 1's goal and non-goal grand averages for the SCREEN condition.

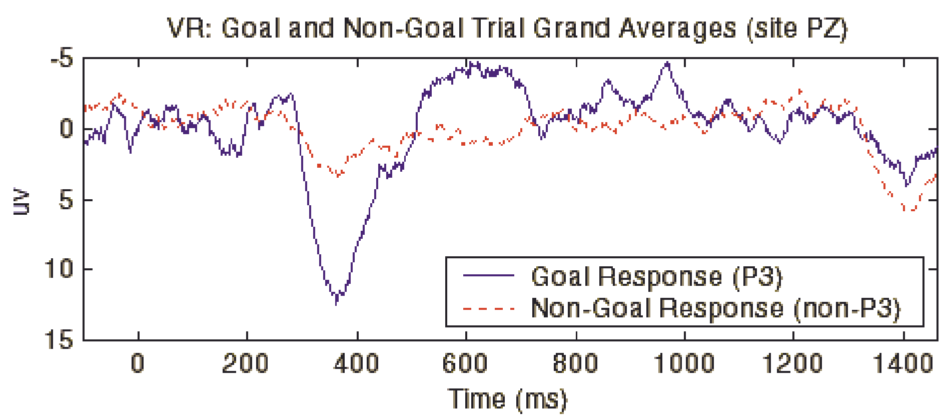


Figure C-12: Subject 1's goal and non-goal grand averages for the VR condition.

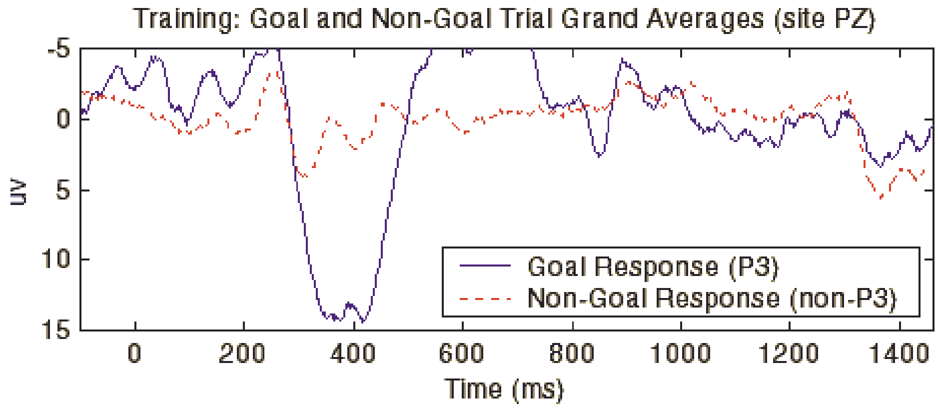


Figure C-13: Subject 1's grand averages during training.

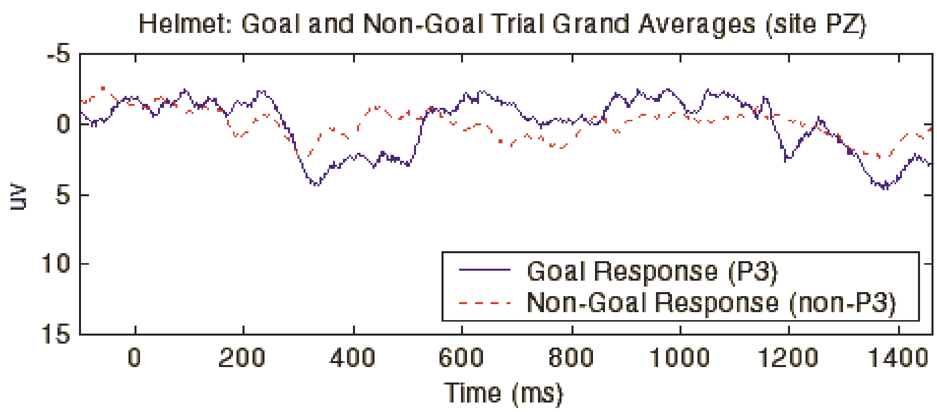


Figure C-14: Subject 2's goal and non-goal grand averages for the FIXED DISPLAY condition.

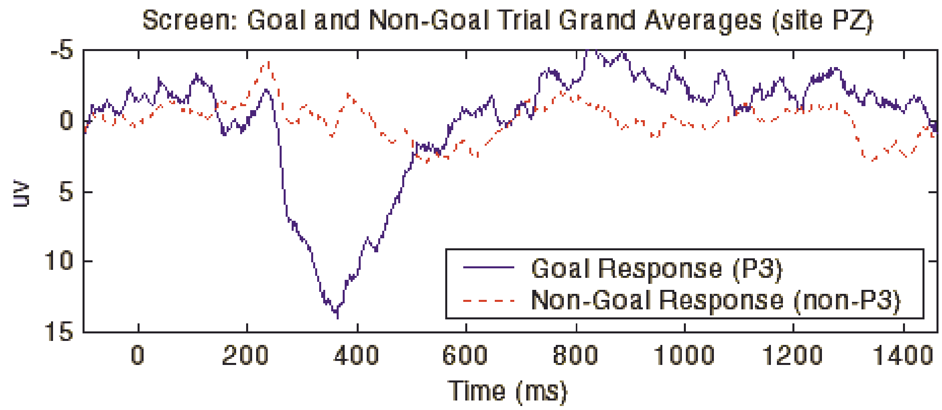


Figure C-15: Subject 2's goal and non-goal grand averages for the MONITOR condition.

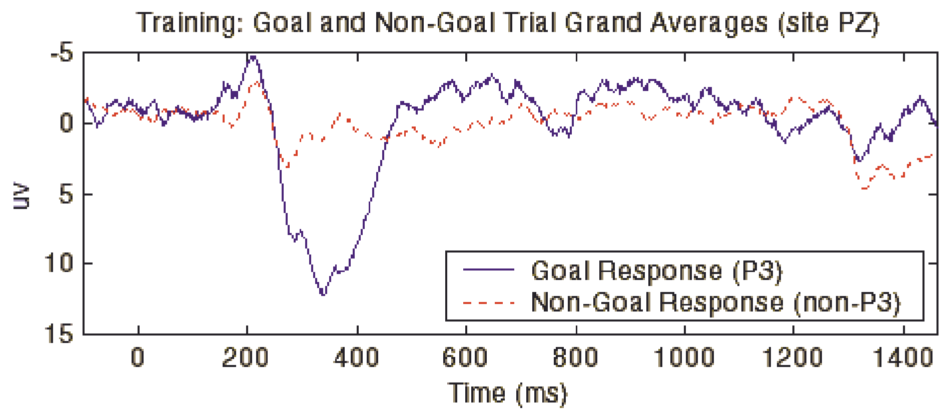


Figure C-16: Subject 2's goal and non-goal grand averages for the training condition.

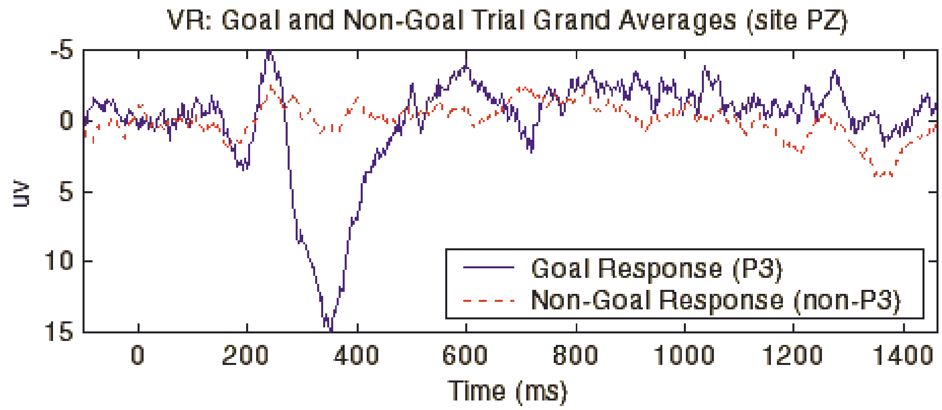


Figure C-17: Subject 2's goal and non-goal grand averages for the VR condition.

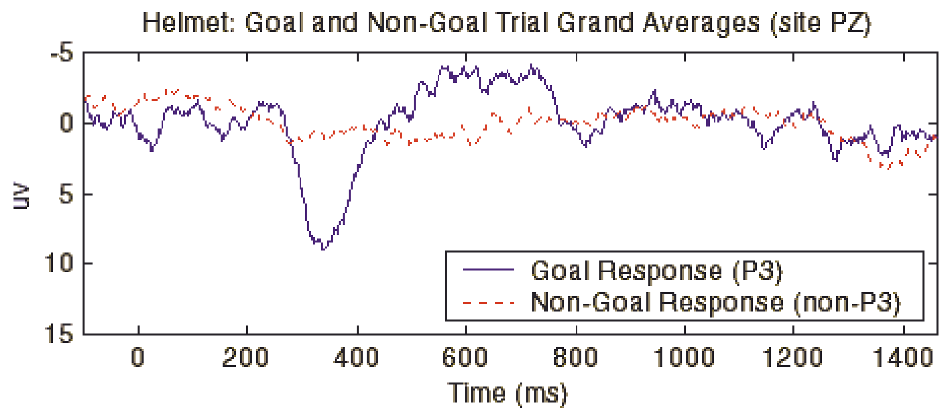


Figure C-18: Subject 3's goal and non-goal grand averages for the FIXED DISPLAY condition.

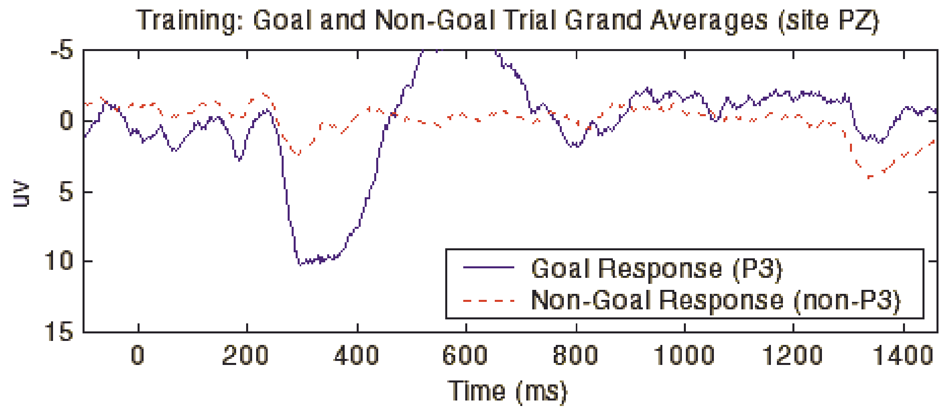


Figure C-19: Subject 3's goal and non-goal grand averages for the training condition.

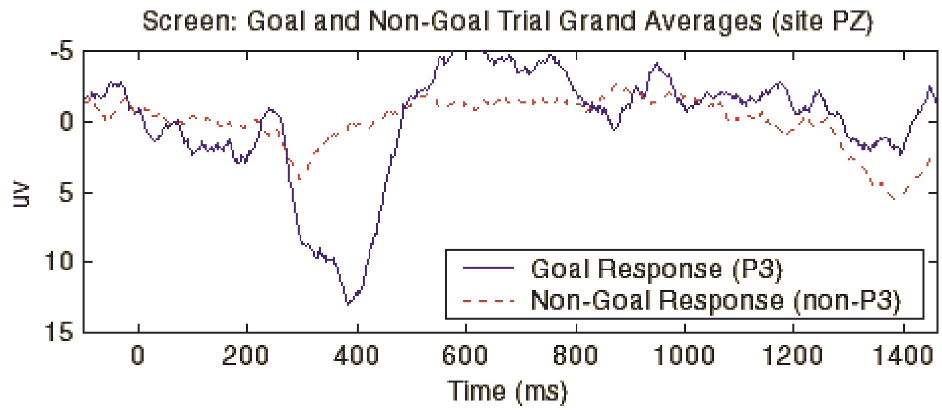


Figure C-20: Subject 3's goal and non-goal grand averages for the MONITOR condition.

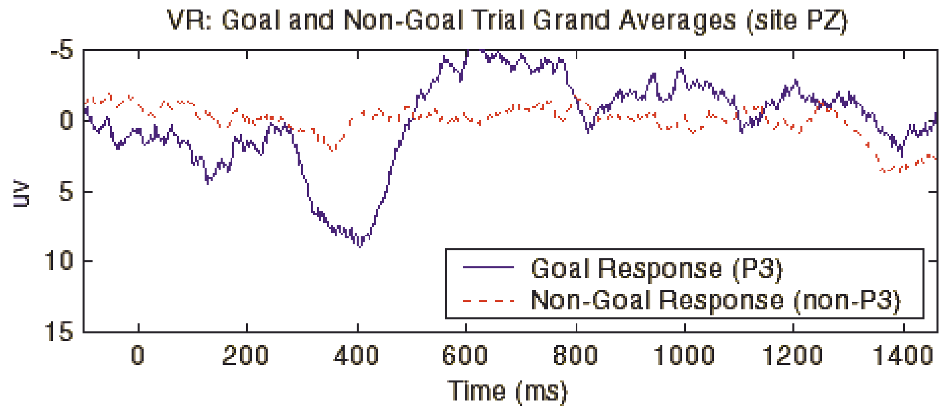


Figure C-21: Subject 3's goal and non-goal grand averages for the VR condition.

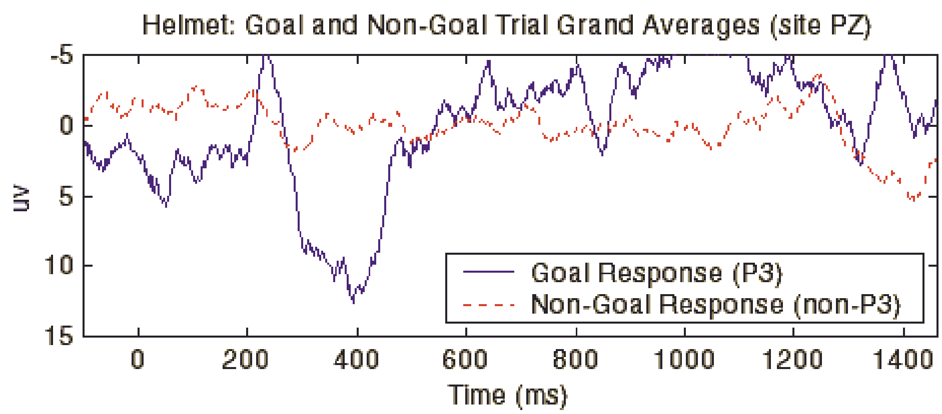


Figure C-22: Subject 4's goal and non-goal grand averages for the FIXED DISPLAY condition.

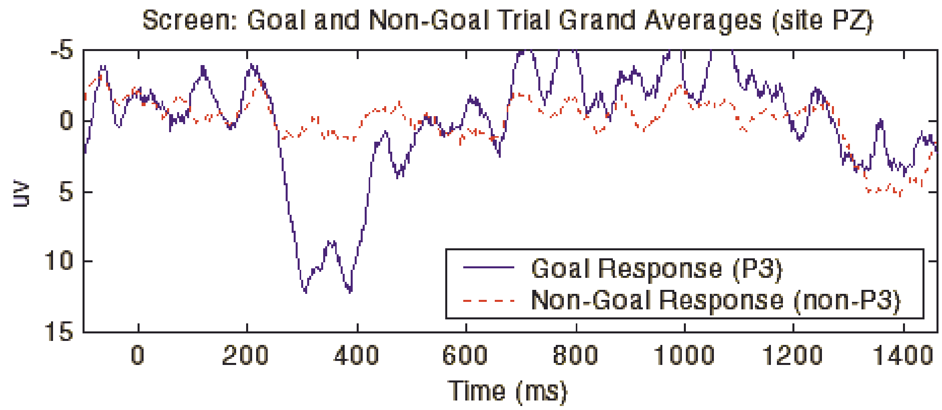


Figure C-23: Subject 4's goal and non-goal grand averages for the MONITOR condition.

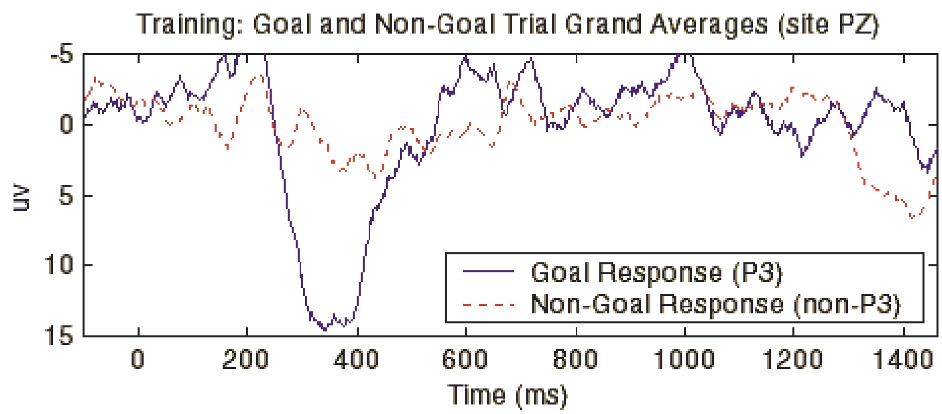


Figure C-24: Subject 4's goal and non-goal grand averages for the training condition.

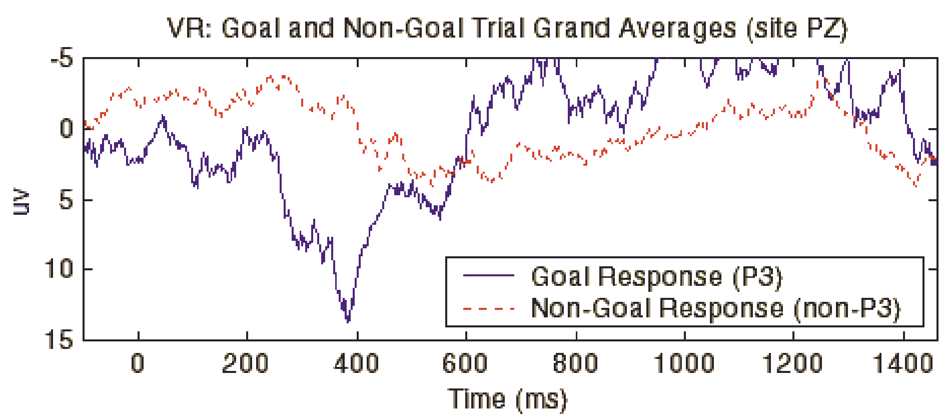


Figure C-25: Subject 4's goal and non-goal grand averages for the VR condition.

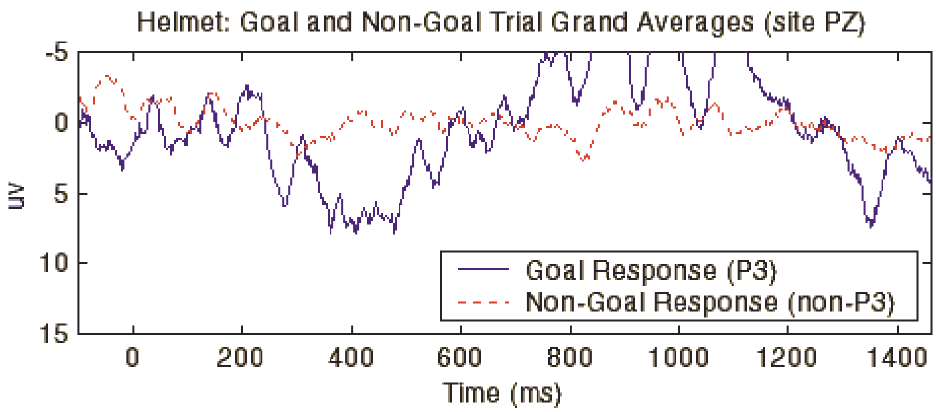


Figure C-26: Subject 5's goal and non-goal grand averages for the FIXED DISPLAY condition.

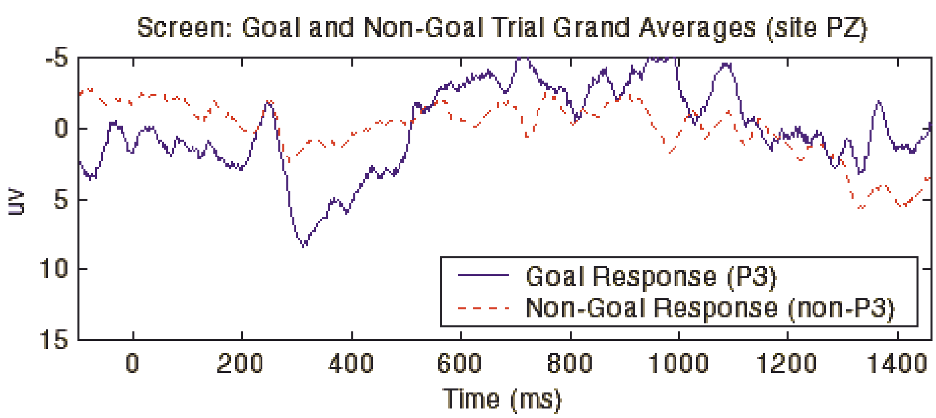


Figure C-27: Subject 5's goal and non-goal grand averages for the MONITOR condition.

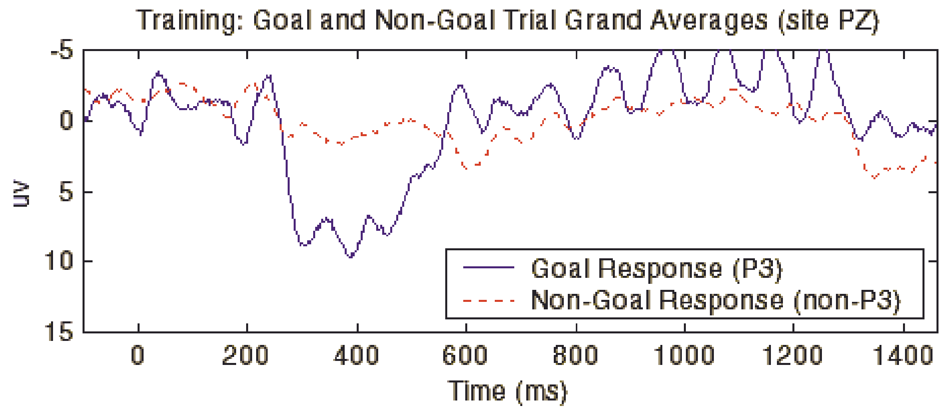


Figure C-28: Subject 5's goal and non-goal grand averages for the training condition.

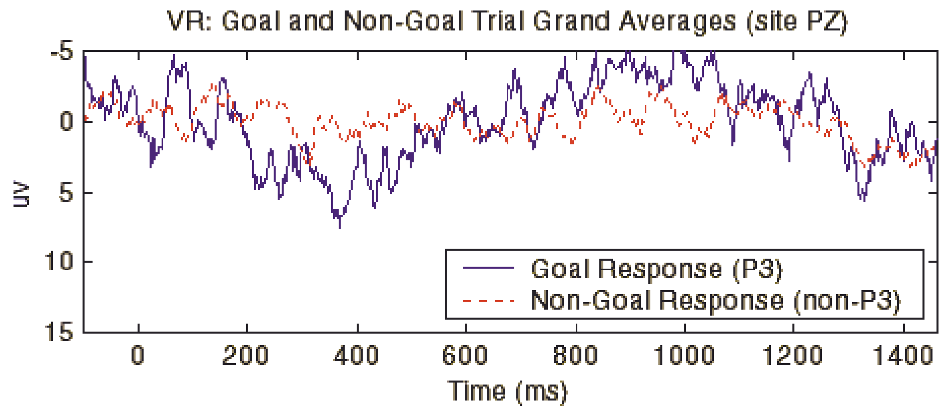


Figure C-29: Subject 5's goal and non-goal grand averages for the VR condition.

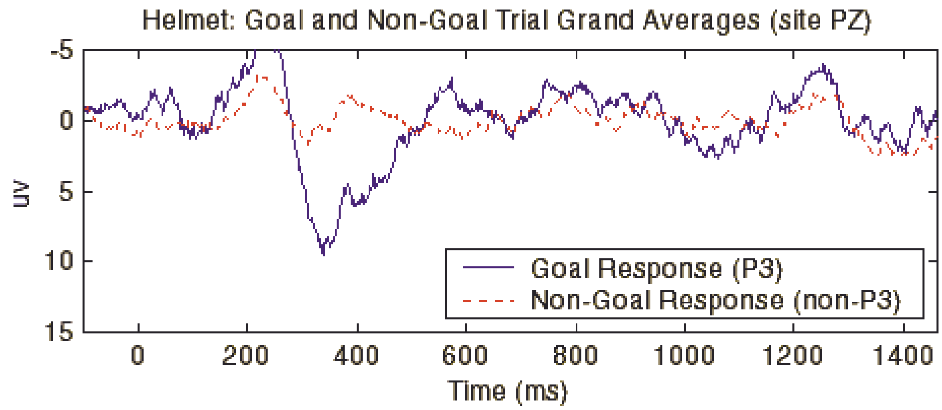


Figure C-30: Subject 6's goal and non-goal grand averages for the FIXED DISPLAY condition.

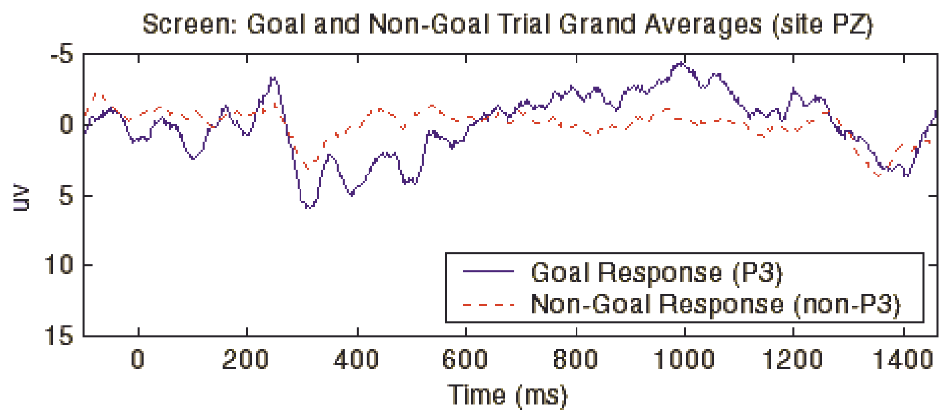


Figure C-31: Subject 6's goal and non-goal grand averages for the MONITOR condition.

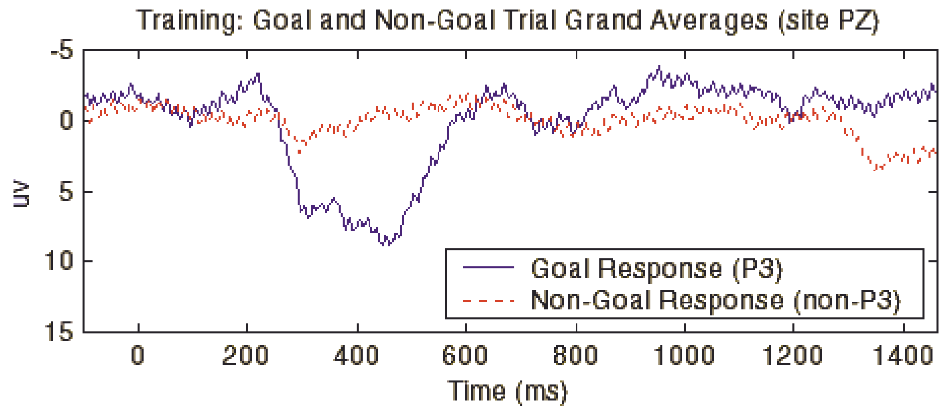


Figure C-32: Subject 6's goal and non-goal grand averages for the training condition.

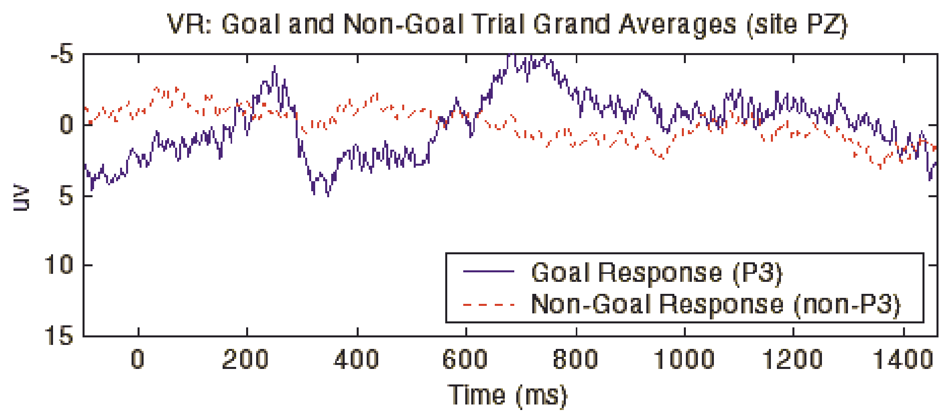


Figure C-33: Subject 6's goal and non-goal grand averages for the VR condition.

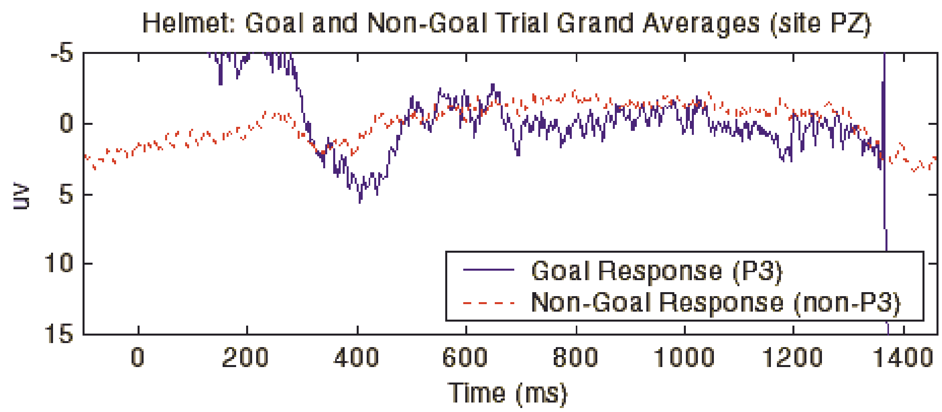


Figure C-34: Subject 7's goal and non-goal grand averages for the FIXED DISPLAY condition.

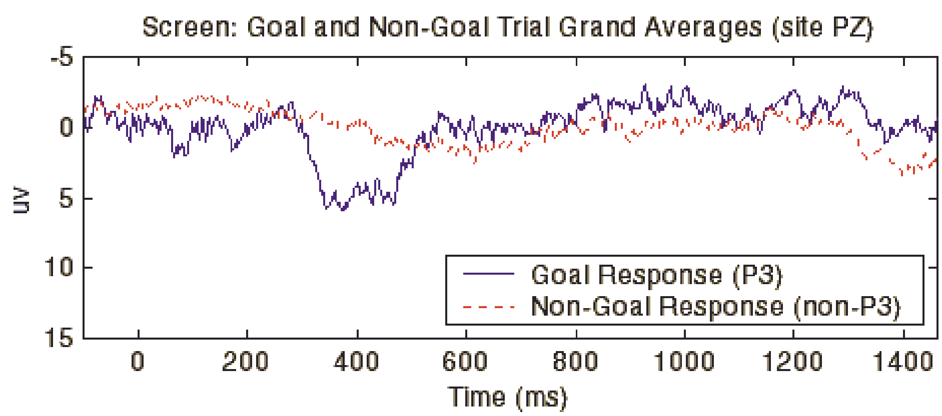


Figure C-35: Subject 7's goal and non-goal grand averages for the MONITOR condition.

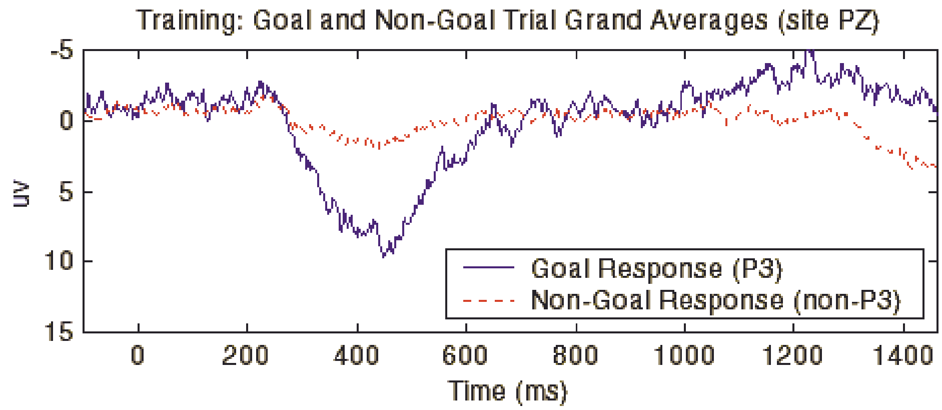


Figure C-36: Subject 7's goal and non-goal grand averages for the training condition.

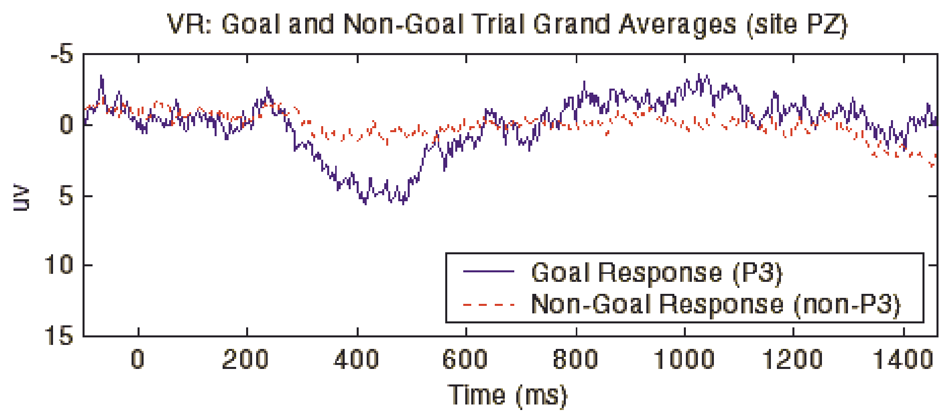


Figure C-37: Subject 7's goal and non-goal grand averages for the VR condition.

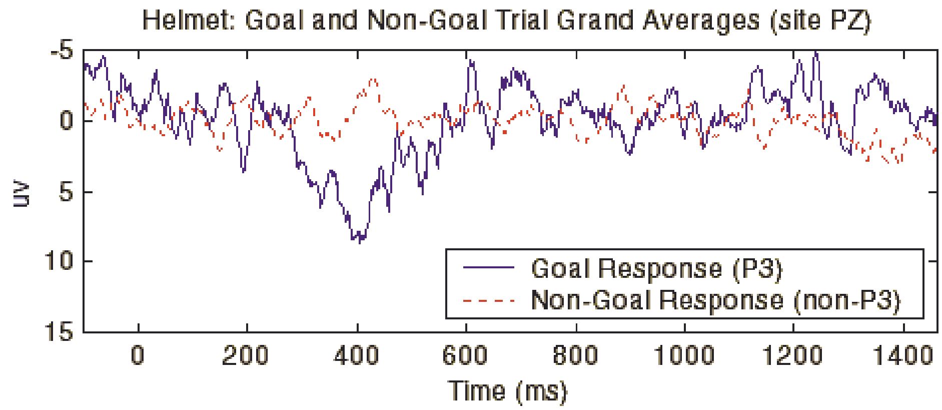


Figure C-38: Subject 8's goal and non-goal grand averages for the FIXED DISPLAY condition.

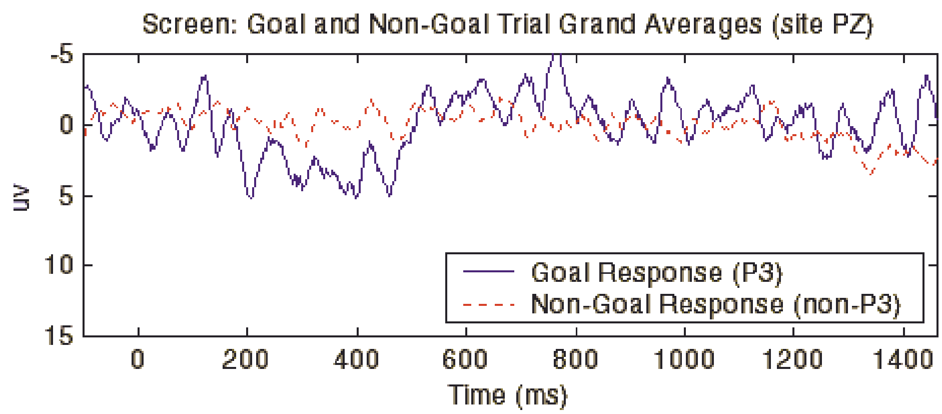


Figure C-39: Subject 8's goal and non-goal grand averages for the MONITOR condition.

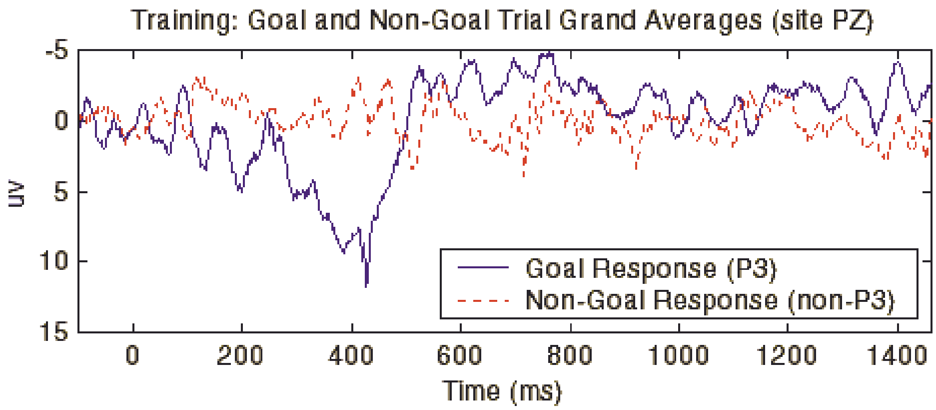


Figure C-40: Subject 8's goal and non-goal grand averages for the training condition.

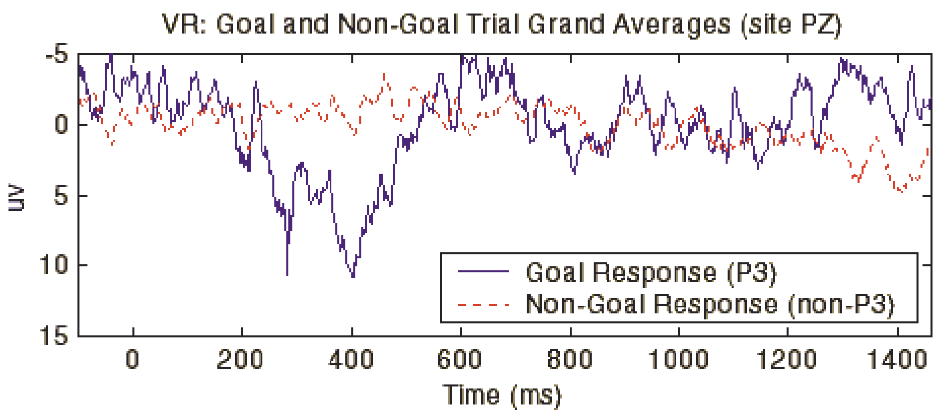


Figure C-41: Subject 8's goal and non-goal grand averages for the VR condition.

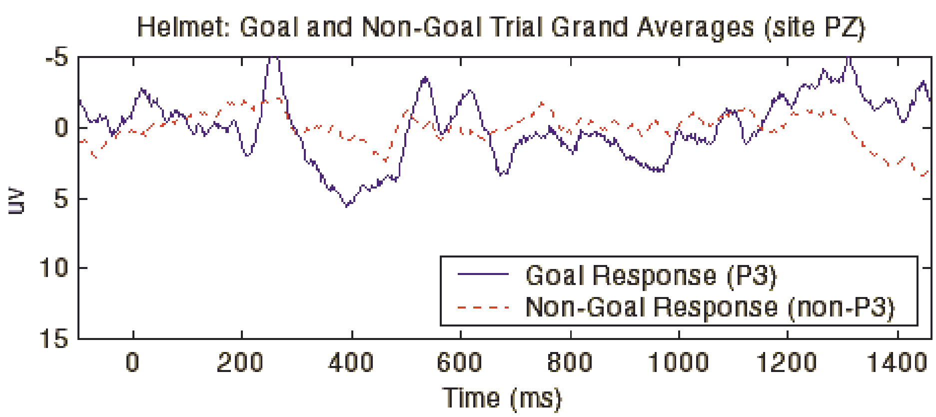


Figure C-42: Subject 9's goal and non-goal grand averages for the FIXED DISPLAY condition.

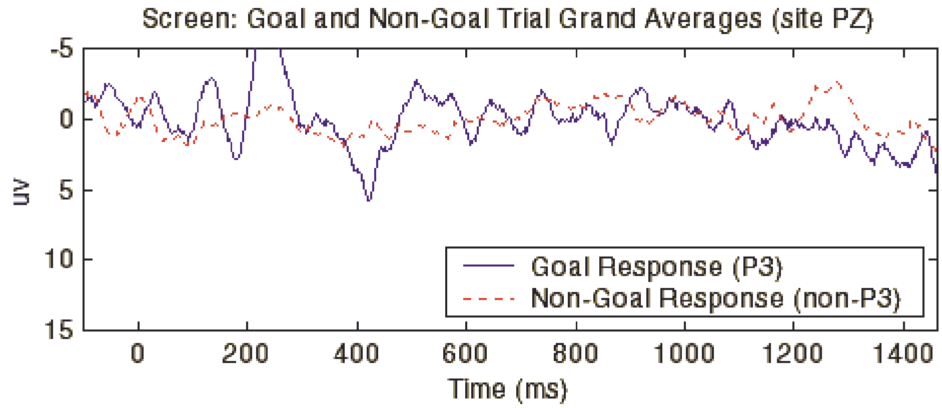


Figure C-43: Subject 9's goal and non-goal grand averages for the MONITOR condition.

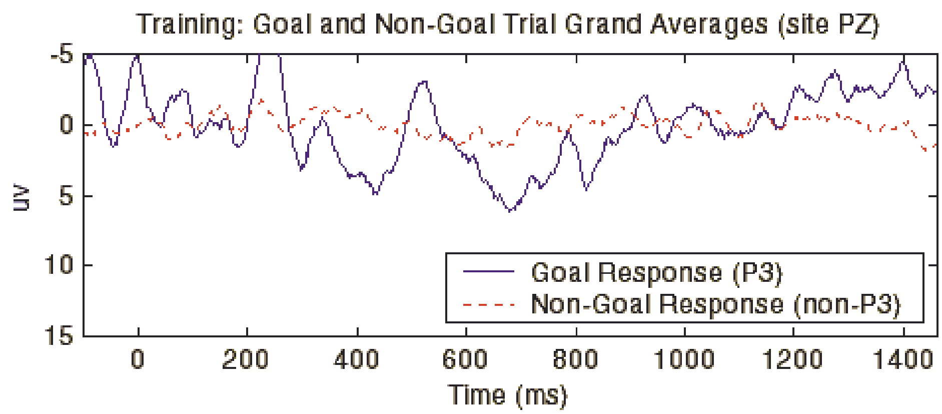


Figure C-44: Subject 9's goal and non-goal grand averages for the training condition.

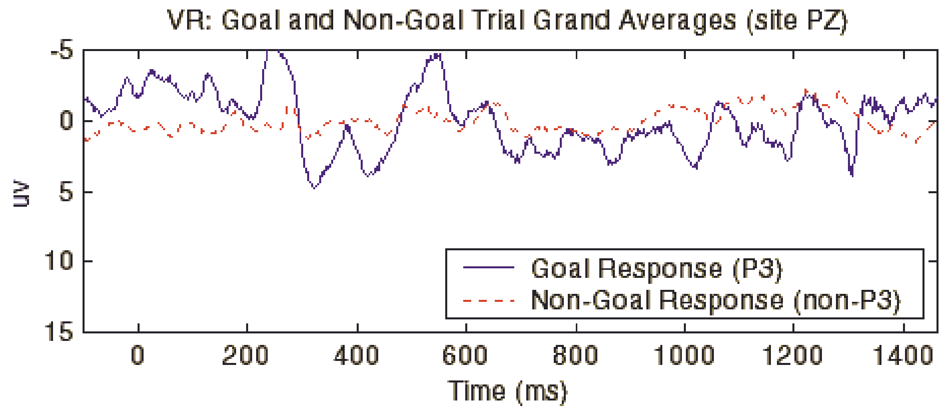


Figure C-45: Subject 9's goal and non-goal grand averages for the VR condition.

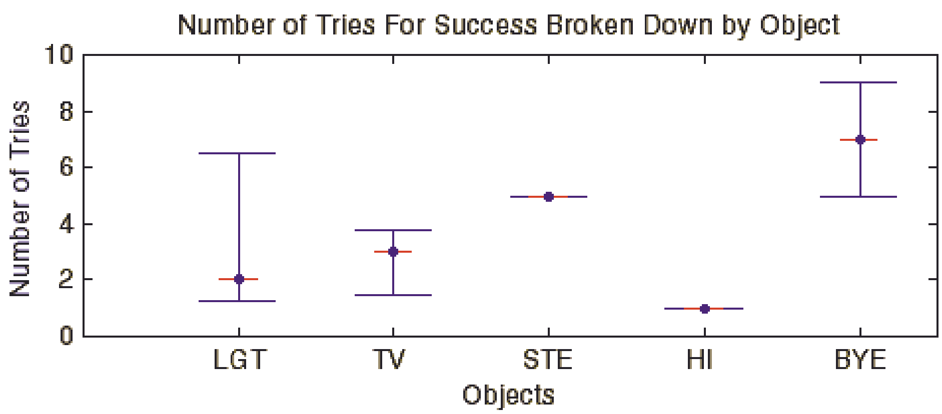


Figure C-46: Subject 1's number of tries per object in the FIXED DISPLAY condition.

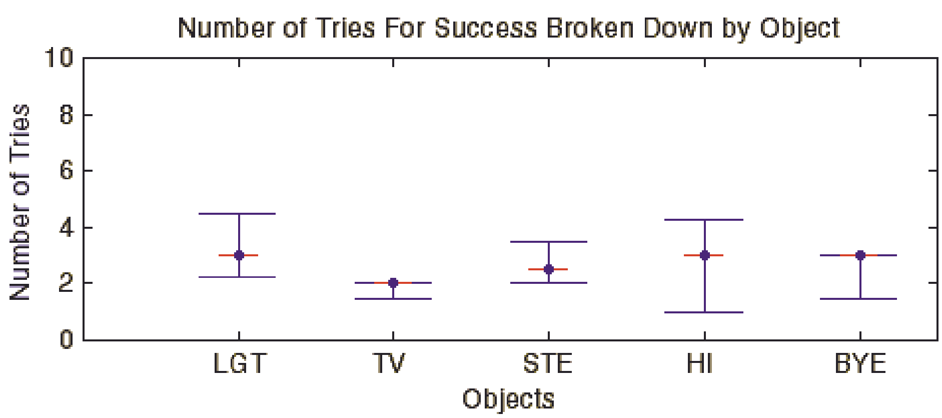


Figure C-47: Subject 1's number of tries per object in the MONITOR condition.

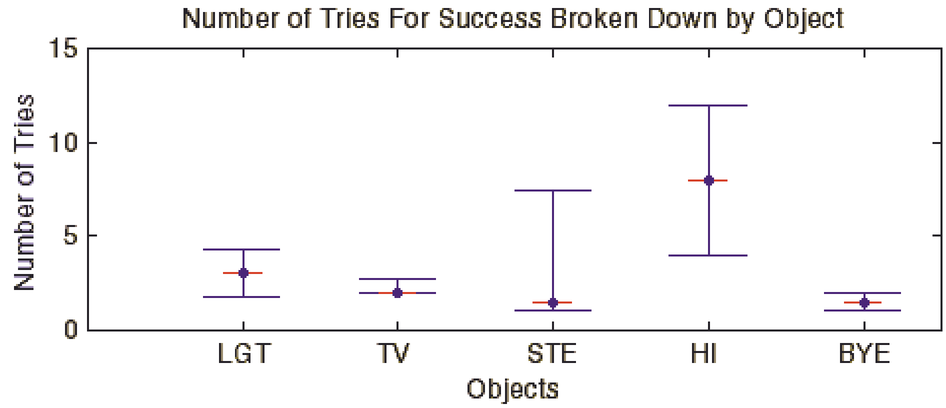


Figure C-48: Subject 1's number of tries per object in the VR condition.

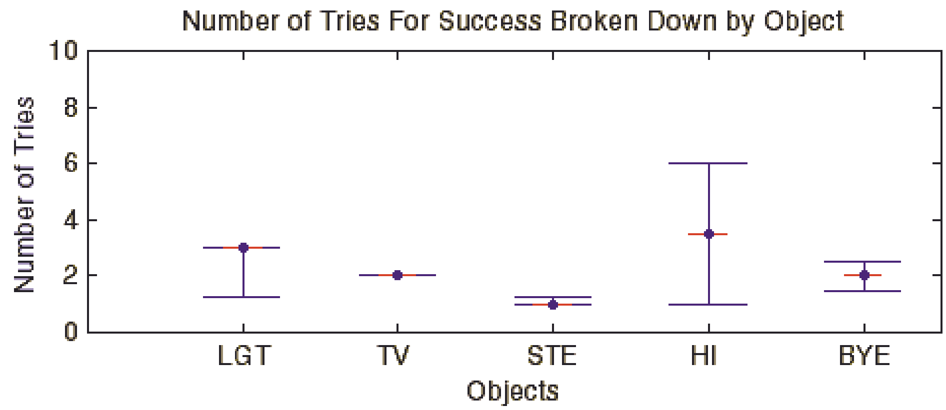


Figure C-49: Subject 2's number of tries per object in the FIXED DISPLAY condition.

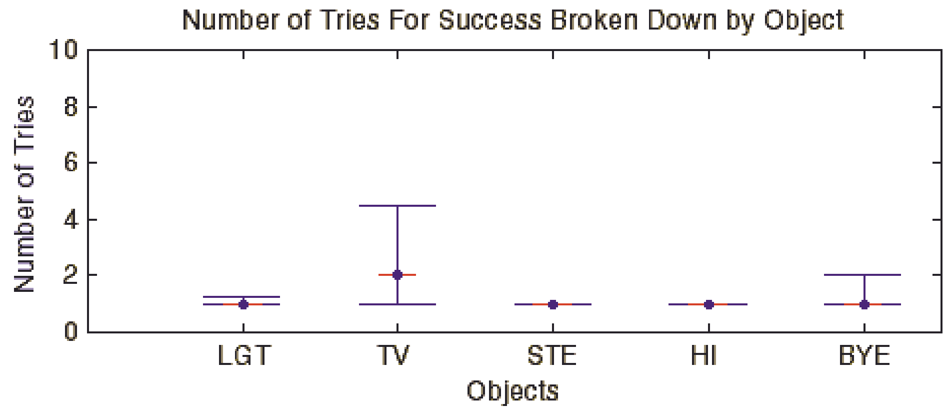


Figure C-50: Subject 2's number of tries per object in the MONITOR condition.

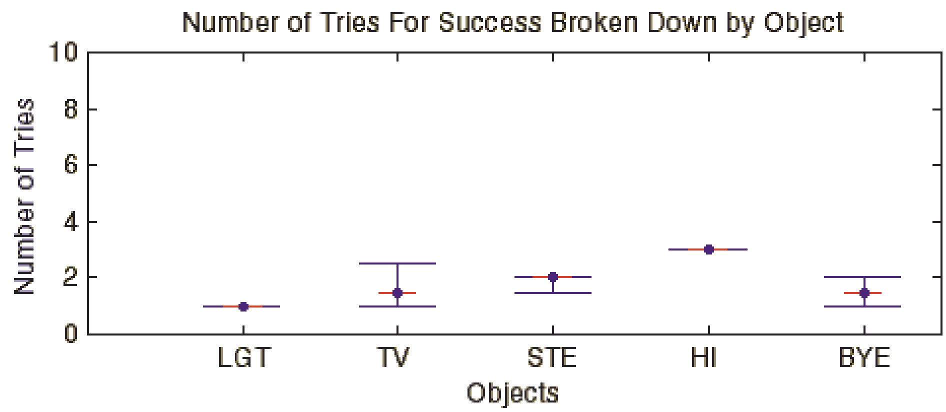


Figure C-51: Subject 2's number of tries per object in the VR condition.

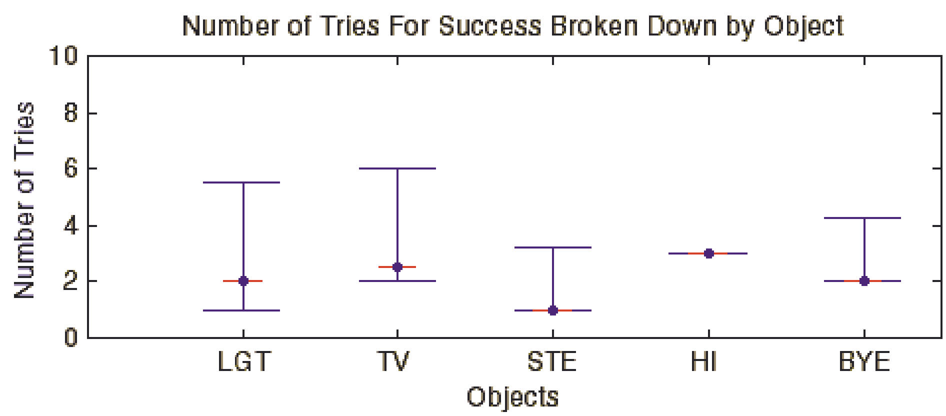


Figure C-52: Subject 3's number of tries per object in the FIXED DISPLAY condition.

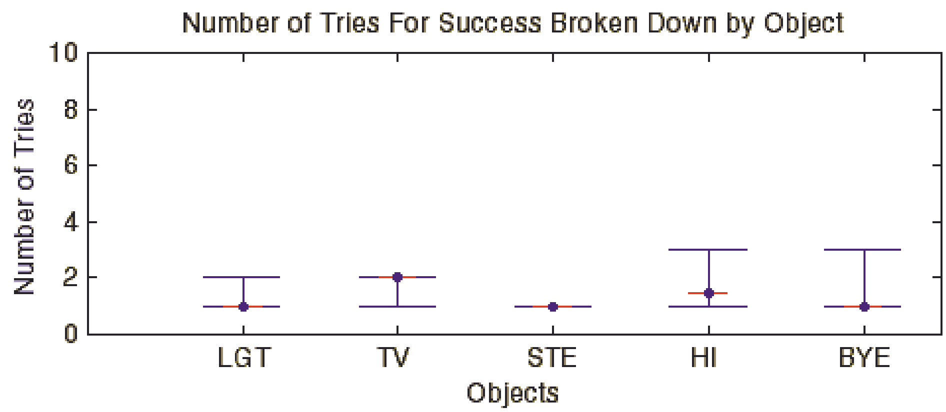


Figure C-53: Subject 3's number of tries per object in the MONITOR condition.

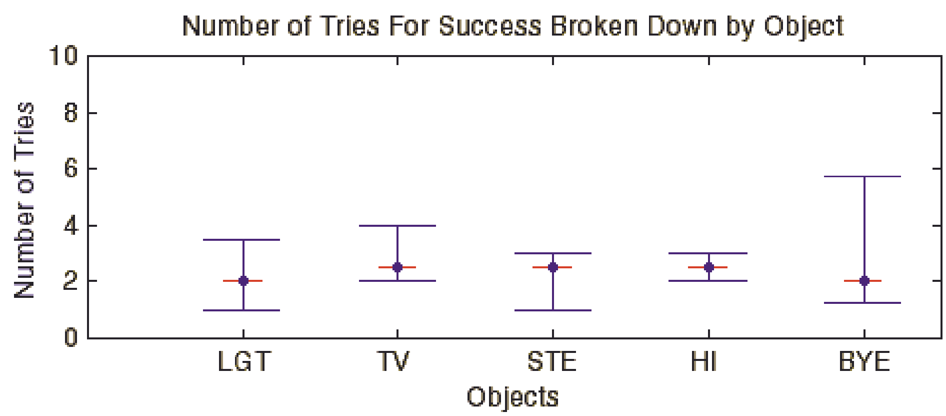


Figure C-54: Subject 3's number of tries per object in the VR condition.

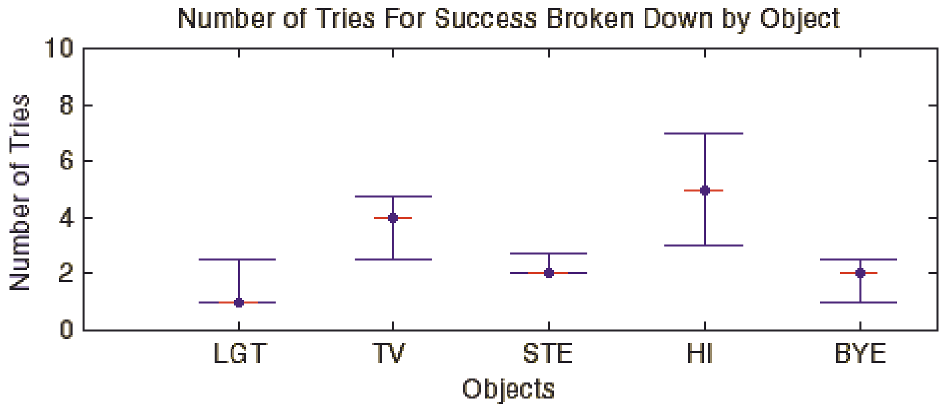


Figure C-55: Subject 4's number of tries per object in the FIXED DISPLAY condition.

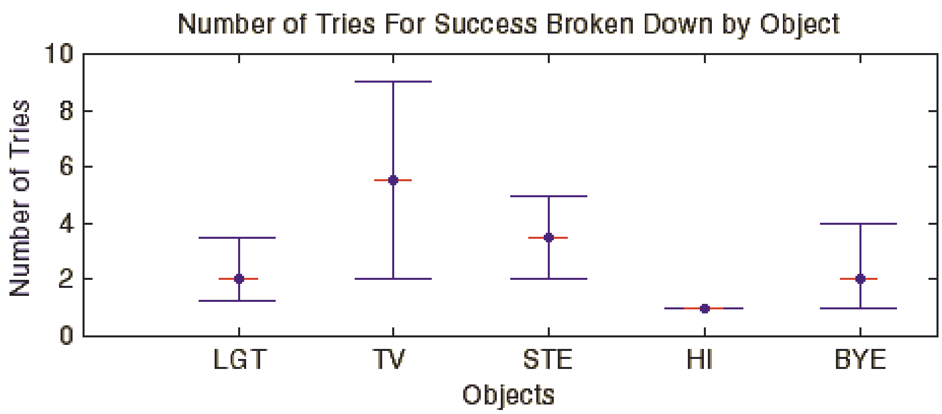


Figure C-56: Subject 4's number of tries per object in the MONITOR condition.

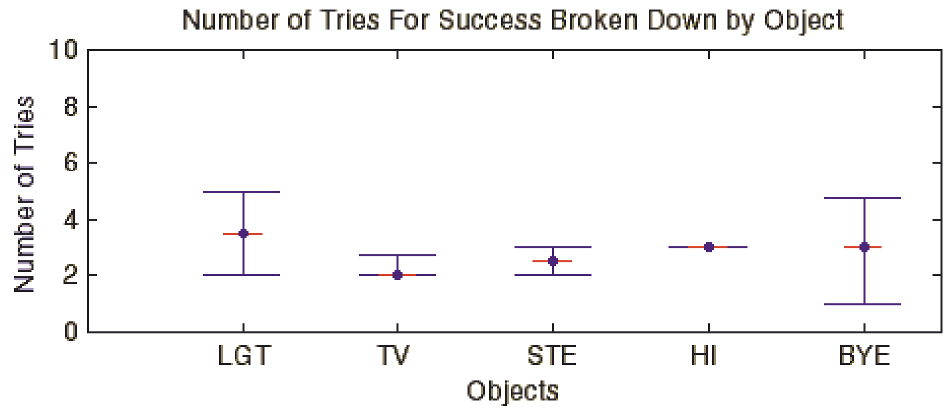


Figure C-57: Subject 4's number of tries per object in the VR condition.

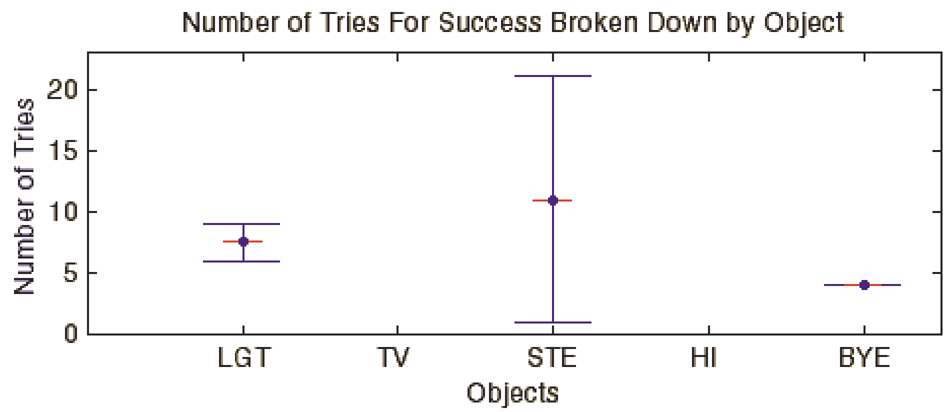


Figure C-58: Subject 5's number of tries per object in the FIXED DISPLAY condition.

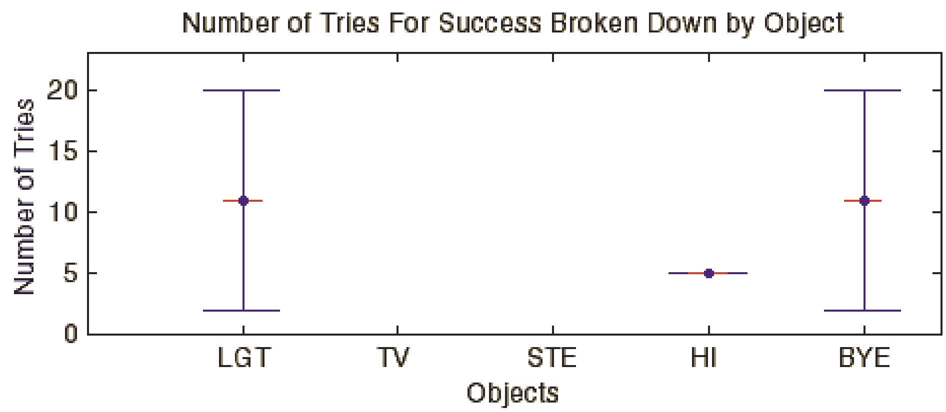


Figure C-59: Subject 5's number of tries per object in the MONITOR condition.

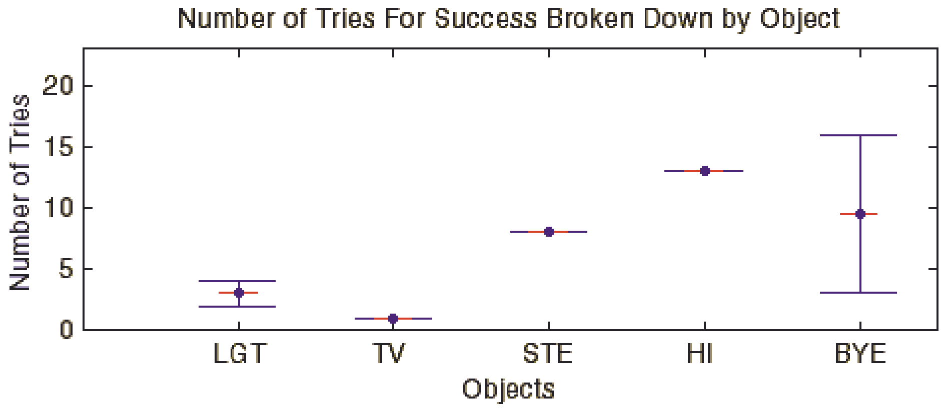


Figure C-60: Subject 5's number of tries per object in the VR condition.

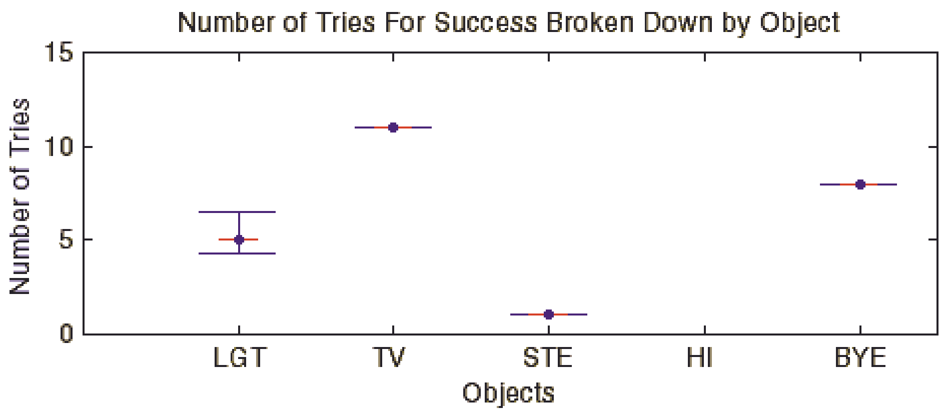


Figure C-61: Subject 6's number of tries per object in the FIXED DISPLAY condition.

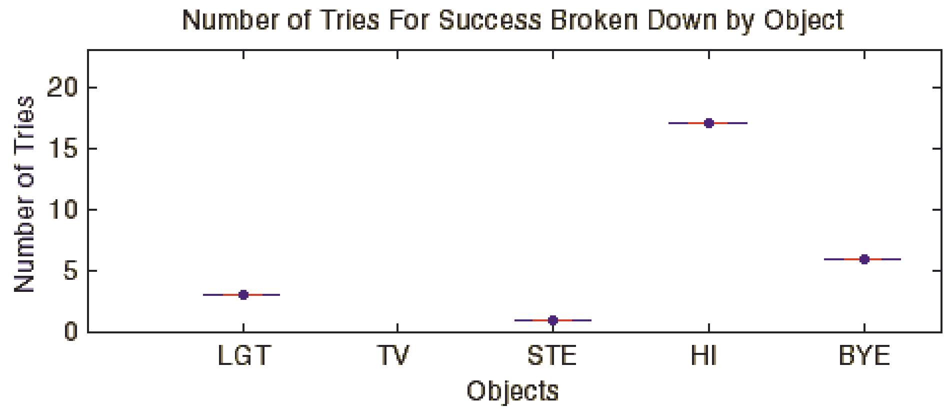


Figure C-62: Subject 6's number of tries per object in the MONITOR condition.

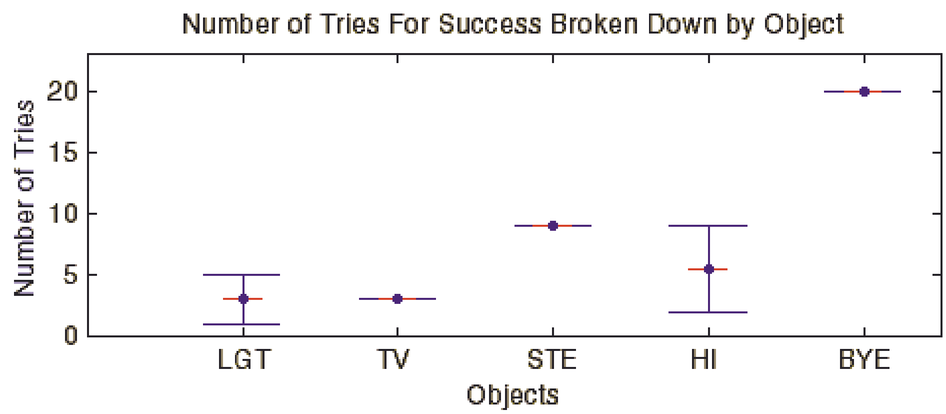


Figure C-63: Subject 6's number of tries per object in the VR condition.

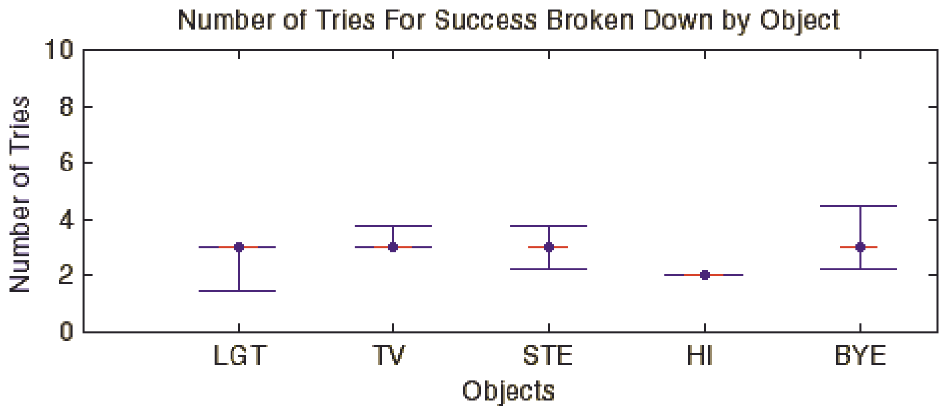


Figure C-64: Subject 7's number of tries per object in the FIXED DISPLAY condition.

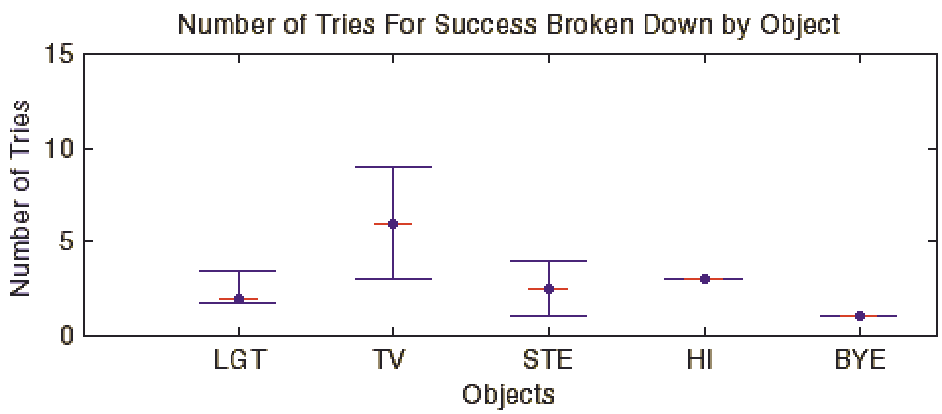


Figure C-65: Subject 7's number of tries per object in the MONITOR condition.

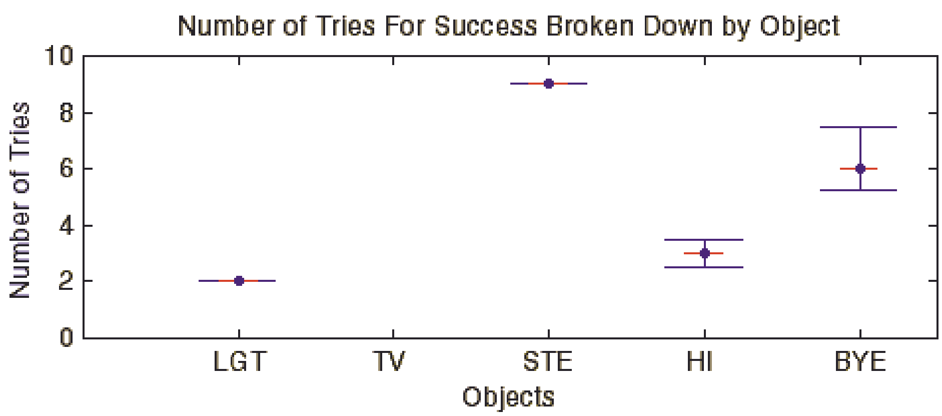


Figure C-66: Subject 7's number of tries per object in the VR condition.

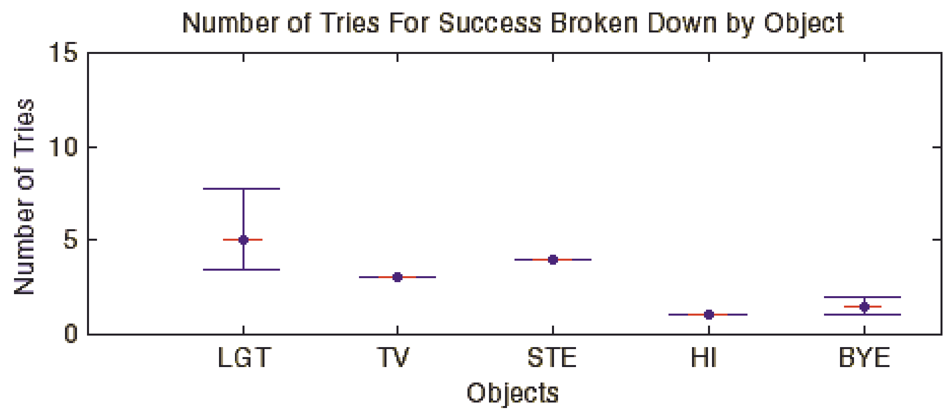


Figure C-67: Subject 8's number of tries per object in the FIXED DISPLAY condition.

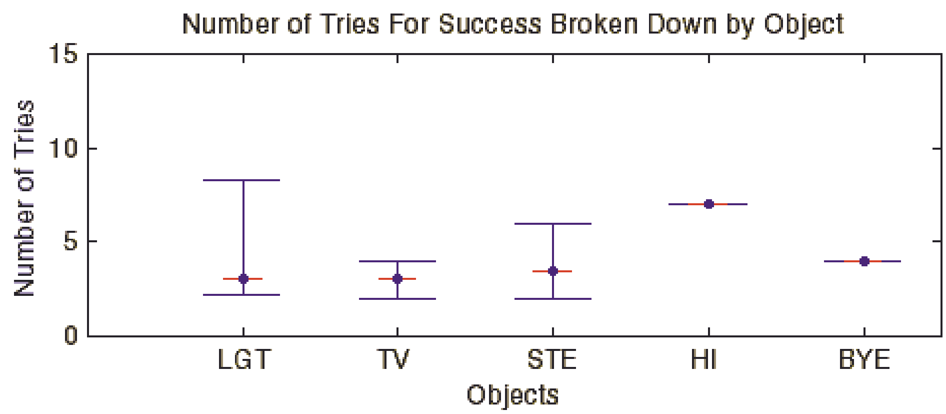


Figure C-68: Subject 8's number of tries per object in the MONITOR condition.

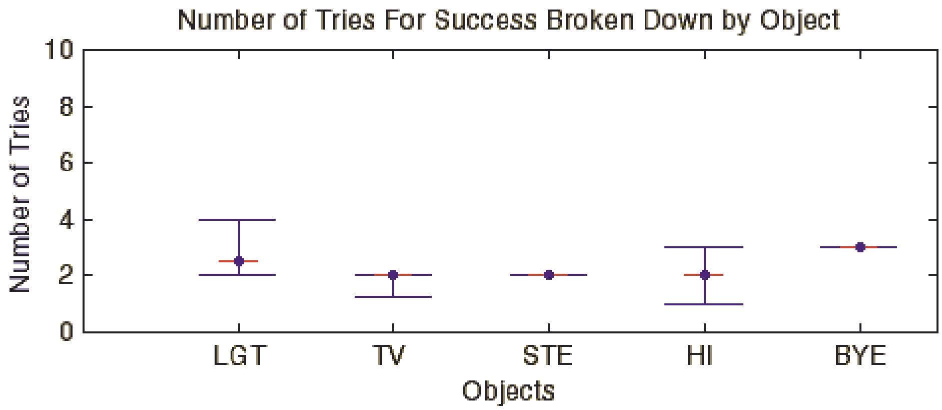


Figure C-69: Subject 8's number of tries per object in the VR condition.

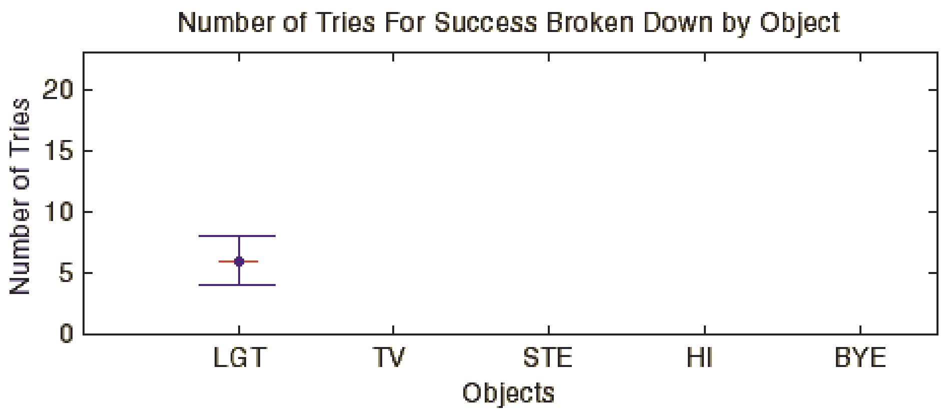


Figure C-70: Subject 9's number of tries per object in the FIXED DISPLAY condition.

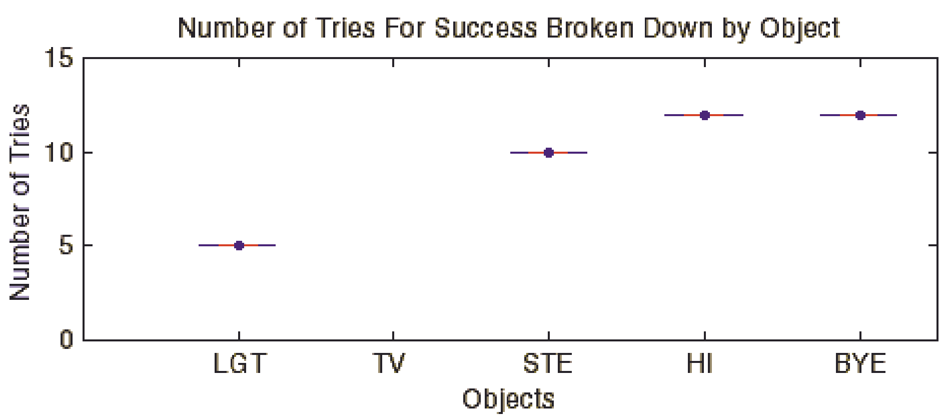


Figure C-71: Subject 9's number of tries per object in the MONITOR condition.

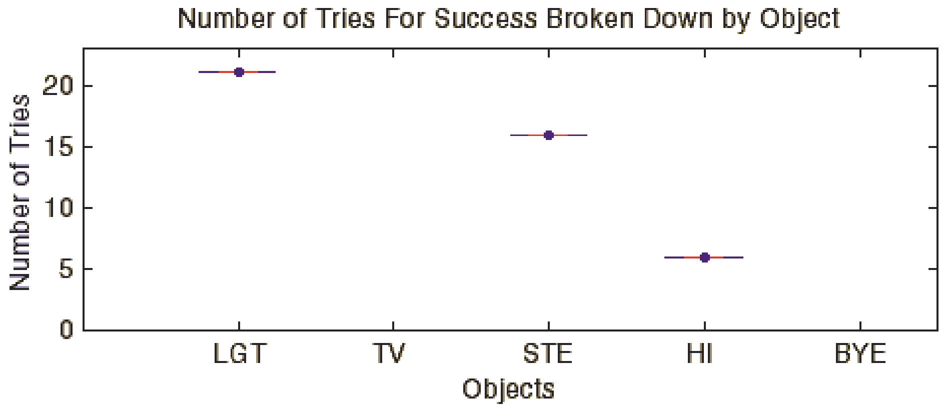


Figure C-72: Subject 9's number of tries per object in the VR condition.