
A Dissertation Presented
by

Hooman Nezamfar

to

The Department of Electrical and Computer Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical and Computer Engineering

Northeastern University
Boston, Massachusetts

March 2016
To my beloved parents, Mahmoud Nezamfar and Pariba Khatib Shahidi.

*May he live long and she rest in peace.*
## Contents

List of Figures iv  
List of Tables vii  
Acknowledgments viii  
Abstract of the Dissertation ix  

1 Introduction 1  
1.1 Visually Evoked Potentials .............................................. 3  
1.1.1 Frequency Based Stimulation ........................................... 3  
1.1.2 Code Based Stimulation ................................................ 5  

2 Real-Time System Design 8  
2.1 User Interface .............................................................. 8  
2.2 Stimuli ........................................................................... 9  
  2.2.1 Shape & Texture ......................................................... 10  
  2.2.2 Conformation .............................................................. 11  
  2.2.3 Color & Contrast .......................................................... 12  
  2.2.4 Presentation rate ......................................................... 13  
  2.2.5 Design & Optimization ................................................. 14  
2.3 Channel Selection ............................................................ 22  
2.4 Calibration ..................................................................... 24  
  2.4.1 Effective Strategies ....................................................... 24  
  2.4.2 Design ...................................................................... 27  
  2.4.3 Duration .................................................................... 28  
2.5 Equipment Setup ............................................................... 31  
2.6 Operation Modes ............................................................... 31  
  2.6.1 Immediate Selection ....................................................... 31  
  2.6.2 Cursor Based Selection .................................................. 31  
2.7 System Parameters & Default Values .................................... 34  
2.8 Offline Analysis & Simulation Tool ....................................... 34  
  2.8.1 Simulation Tool ............................................................. 34  
2.9 Alternative Input Modalities ................................................. 35
2.9.1 Eye Tracking ......................................................... 36

3 EEG Signal Modeling, Context Fusion & Decision Making ................................................. 39
3.1 Template Matching .................................................. 39
  3.1.1 Single Channel Template-Matching ......................... 40
  3.1.2 Multi-channel Template Matching ............................ 41
3.2 Kernel Density Estimation ........................................... 41
  3.2.1 One dimensional KDE .......................................... 42
  3.2.2 Multidimensional KDE ......................................... 42
  3.2.3 Naïve Bayesian Fusion ......................................... 42
3.3 Multivariate Gaussian Distribution ................................. 44
  3.3.1 Sparse Covariance Estimation ................................. 46
3.4 Introduction of Context Information ................................. 48
  3.4.1 Dynamic Confidence Threshold ............................... 49
  3.4.2 Mastery Task & Training ...................................... 50
3.5 Online Operation of FlashLife™ .................................... 51
  3.5.1 Fusion rule .................................................. 52
  3.5.2 Decision Rule ................................................ 54

4 Applications ............................................................ 55
4.1 FlashNav™ .......................................................... 56
4.2 FlashGrab™ .......................................................... 57
4.3 FlashType™ .......................................................... 58
  4.3.1 User Interface Design .......................................... 58
  4.3.2 Keyboard .................................................... 59
  4.3.3 Language Model Information Extraction ................... 60
  4.3.4 Operation Modes ............................................ 63
4.4 FlashPlay™ ........................................................... 64

5 Experimental Results .................................................. 66
5.1 Confidence Threshold & Visual Cue ................................ 66
5.2 Eye Tracking vs Code-VEP .......................................... 75
  5.2.1 User Experience ............................................... 76
  5.2.2 Results & Discussion ......................................... 77
5.3 Simulation based Performance Estimation ......................... 81
5.4 FlashType™ .......................................................... 84
  5.4.1 Results & Discussion ......................................... 84

6 Discussion & Future Work ............................................ 90
6.1 Future Work .......................................................... 92

Bibliography .............................................................. 93
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Frequency response of Oz channel in response to 15 Hz flickering LED stimulus</td>
<td>4</td>
</tr>
<tr>
<td>1.2</td>
<td>a) checkerboard stimulus pattern one, b) checkerboard stimulus pattern two, c) a pseudo-random binary control sequence of length 31 bit</td>
<td>5</td>
</tr>
<tr>
<td>1.3</td>
<td>Frequency response of the Oz channel to stimulation with four different m-sequence of length 31</td>
<td>7</td>
</tr>
<tr>
<td>2.1</td>
<td>General user interface theme, (a) Default stimuli, (b) Alternative stimuli for color blind. Purple and blue areas are just place holders and they are replaced by stimuli labels and application screen respectively.</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>Retina response density [1]</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>Perspective effect on the stimulus strength [2]</td>
<td>12</td>
</tr>
<tr>
<td>2.4</td>
<td>Electrode locations based on the 10-20 standard.</td>
<td>14</td>
</tr>
<tr>
<td>2.5</td>
<td>Probability of correct decision among 4 m-sequences as a spatial scalp distribution at 15 and 30 Hz flickering frequencies for subjects</td>
<td>16</td>
</tr>
<tr>
<td>2.6</td>
<td>(a) The checkerboard pattern corresponding to Bit &quot;0&quot;, (b) The checkerboard pattern corresponding to Bit &quot;1&quot;, and (c) The target arrangement of the stimuli.</td>
<td>17</td>
</tr>
<tr>
<td>2.7</td>
<td>Single channel classification accuracy of choosing among 4 simultaneously presented stimuli under different stimuli settings presented as a spatial scalp distribution. Every row corresponds to one participant. B/W, R/G and B/Y represent black and white, red and green and blue and yellow opponent color pairs.</td>
<td>19</td>
</tr>
<tr>
<td>2.8</td>
<td>Paired t-tests among different opponent color pairs under different bit presentation rates and different m-sequence lengths. Electrode locations marked with * showed significance with $p &lt; 0.05$.</td>
<td>20</td>
</tr>
<tr>
<td>2.9</td>
<td>Templates response of channel Oz for the stimulus 1 under different opponent color pairs for participant P02 on the left and P03 on the right. The black, red and blue lines are representing the black-white, red-green and blue-yellow opponent color pairs respectively.</td>
<td>22</td>
</tr>
<tr>
<td>2.10</td>
<td>Probability of correct decision for individual channels for each m-sequence using 15 Hz bit presentation rate – (a) male participant, (b) female participant.</td>
<td>25</td>
</tr>
<tr>
<td>2.11</td>
<td>Probability of correct decision for individual channels for each m-sequence using 30 Hz bit presentation rate – (a) male participant, (b) female participant.</td>
<td>26</td>
</tr>
<tr>
<td>2.12</td>
<td>Intending strategies known to produce stronger VEP responses using FlashLife™, (a) Blacks, (b) Joints. Every color is representing one intending strategy.</td>
<td>27</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>2.13</td>
<td>Probability of correct decision on test set using different number of training samples to determine templates for the second classifier for each m-sequence using 15 Hz bit presentation rate – (a) male participant, (b) female participant</td>
<td>29</td>
</tr>
<tr>
<td>2.14</td>
<td>Probability of correct decision on test set using different number of training samples to determine templates for the second classifier for each m-sequence using 30 Hz bit presentation rate – (a) male participant, (b) female participant</td>
<td>30</td>
</tr>
<tr>
<td>3.1</td>
<td>Classification accuracy of naïve Bayesian fusion of best-m channels for m from 1 to 16</td>
<td>45</td>
</tr>
<tr>
<td>3.2</td>
<td>Channel locations and the connectivity (red lines) defined using GLASSO method</td>
<td>47</td>
</tr>
<tr>
<td>3.3</td>
<td>Calibration accuracies over two days and two Calibration sessions each day, for single channel and multi-channel classification methods. (D stands for Day)</td>
<td>47</td>
</tr>
<tr>
<td>3.4</td>
<td>General System Graphical Model (Epoch $k$)</td>
<td>53</td>
</tr>
<tr>
<td>4.1</td>
<td>Three snapshots of the low level context information extracted from the environment [3].</td>
<td>57</td>
</tr>
<tr>
<td>4.2</td>
<td>An example of the screen in the initial state of FlashType™. The top center rectangle provides the user with a visual feedback of what has been typed so far.</td>
<td>59</td>
</tr>
<tr>
<td>4.3</td>
<td>The default keyboard provided by the FlashType™ in the initial state. The first part from the top, presents three predicted words. The second part, presents the top seven most probable characters based on the language model estimates. The third part, consisting of four rows presents the static part of the keyboard.</td>
<td>60</td>
</tr>
<tr>
<td>4.4</td>
<td>System graphical model in epoch $e$.</td>
<td>61</td>
</tr>
<tr>
<td>4.5</td>
<td>Sample screenshots of two maze games using red &amp; green and black &amp; white stimuli. In each maze the correct path is highlighted in yellow and the incorrect paths are highlighted in pink.</td>
<td>64</td>
</tr>
<tr>
<td>5.1</td>
<td>Mazes used for 4 different Mastery levels. The first row shows the mazes used in the first half of each session and the second row shows the mazes used in the second half.</td>
<td>67</td>
</tr>
<tr>
<td>5.2</td>
<td>Actual participant accuracy and number of sequences used for 8 task under difficulty levels during the first data collection session. SC stands for static confidence threshold, DC stands for dynamic confidence threshold and VC stands for visual cue consideration. (Part I)</td>
<td>69</td>
</tr>
<tr>
<td>5.3</td>
<td>Actual participant accuracy and number of sequences used for 8 task under difficulty levels during the second data collection session. SC stands for static confidence threshold, DC stands for dynamic confidence threshold and VC stands for visual cue consideration. (Part I)</td>
<td>72</td>
</tr>
</tbody>
</table>
5.3 Actual participant accuracy and number of sequences used for 8 task under difficulty levels during the second data collection session. SC stands for static confidence threshold, DC stands for dynamic confidence threshold and VC stands for visual cue consideration. (Part II) .................................................. 73
5.3 Actual participant accuracy and number of sequences used for 8 task under difficulty levels during the second data collection session. SC stands for static confidence threshold, DC stands for dynamic confidence threshold and VC stands for visual cue consideration. (Part III) .................................................. 74
5.4 (a) First task, (b) Second task, (c) Third task, and (d) Fourth task. .............................. 76
5.5 Illustration of the screen when task Maze 1 starts. .................................................. 77
5.6 Average setup and calibration duration for different input modalities. Two average setup durations are calculated for c-VEP method including and excluding the one time calibration duration. ................................. 78
5.7 Average completed task duration for different input modalities. ................................ 78
5.8 Average accuracy of the decisions made by the two input modalities. ......................... 79
5.9 Average number of infeasible decisions made by the two input modalities. .................. 80
5.10 Actual and estimated user performances using the Simulations over 4 different tasks. (a) Average accuracy of the decisions, (b) Average number of trials per decision and (c) Average number of mistakes. ........................................... 83
5.11 Parts (a), (b) and (c), show the average value of the prior probabilities for the correct choice, the average value of the the maximum prior probability of the incorrect choices and the percentage of the number of times when the prior assigned to the correct choice has been greater than the maximum of the priors assigned to the other choices for typing each word. Part (d) shows the number of decisions each participant made to complete the task. .................................................. 85
5.12 Prior probabilities estimated by the language model for every character on the screen at every stage of typing the word "seen". Each plot demonstrates how the prior probabilities evolve as the cursor moves towards the correct target. The solid thick line shows the prior probability of the correct choice and the dashed lines show the prior probability of the other choices. ........................................... 86
5.13 Prior probabilities estimated by the language model for every character on the screen at every stage of typing the word "with". Every row belongs to the moves towards choosing a letter, except the last row, which belongs to choosing the complete word from the predicted words row. ........................................................ 87
5.14 Prior probabilities estimated by the language model for every character on the screen at every stage of typing the word "buys". Every row belongs to the moves towards choosing a letter. ...................................................................... 88
5.15 The overall usage of the keyboard parts, Static Keyboard, Character Suggestions, and the Word Predictions by each participant. .................................................. 88
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Single channel classification performance, stimuli presented with bit presentation rate of 15 bps.</td>
</tr>
<tr>
<td>2.2</td>
<td>Single channel classification performance, stimuli presented with bit presentation rate of 30 bps.</td>
</tr>
<tr>
<td>2.3</td>
<td>Color pair, bit presentation rate and m-sequence length triplets studied. B/W represents black and white, R/G represents red and green and B/Y represents blue and yellow color pairs.</td>
</tr>
<tr>
<td>2.4</td>
<td>Stimulation trial duration.</td>
</tr>
<tr>
<td>2.5</td>
<td>Fundamental frequency components generated by each m-sequence under different bit presentation rates. The frequency sets are identical for all the m-sequences used no matter the length, except the underlined values which are only available for the m-sequences of length 127 bits.</td>
</tr>
<tr>
<td>2.6</td>
<td>Performance of channel Oz based on the calibration data under different colors, bit presentation rates and control bit sequence lengths. Best performance for each individual is marked in bold. The last two rows are column wise average and standard deviation respectively.</td>
</tr>
<tr>
<td>2.7</td>
<td>Available stimuli roles under different active dimensions and active directions. D, U, R and L represent Down, Up, Right and Left, respectively. $d_V$ and $d_H$ represent the active direction in the Vertical and Horizontal directions, respectively.</td>
</tr>
<tr>
<td>4.1</td>
<td>Experiment Task Durations.</td>
</tr>
<tr>
<td>4.2</td>
<td>Mastery task probability boundaries.</td>
</tr>
<tr>
<td>5.1</td>
<td>User response summary. B stands for Brain Interface (e-VEP), E stands for Eye Tracker, G stands for glasses and CL stands for contact lenses. Calibration accuracy is estimated based on the Calibration data.</td>
</tr>
</tbody>
</table>

Page vii
Acknowledgments

My experience at Northeastern would not be the same without the help of many people. First, I wish to thank my advisor Prof. Deniz Erdogmus, who kindly supported me during my Ph.D. studies and provided the means of research and knowledge in such a great field that can benefit individuals in need. Through his efforts, I have collaborated with many people at Northeastern University, Oregon Health & Science University, Boston University, and Worcester Polytechnic Institute.

I would like to thank my committee members Prof. Dana Brooks for his support inside and outside of my research, Prof. Gunar Schirner for great collaboration experience and Prof. Frank Guenther for invaluable input.

Being a member of the Cognitive Systems Lab. and B-SPIRAL, I would like to thank my friends, current and former members of the group, Sina (Mohammad) Moghadamfalahi, Jamshid Surati, Umut Orhan, Murat Akcakaya, Shalini Purwar, Asieh Ahani, Marzieh Haghighi, Golnaz Eftekhari, Sadegh Salehi, Fernando Quivira, Matineh Shaker, Nastaran Ghadar, Seyhmus Guler, Yeganeh Marghi, Jauma Col Font, Sheng You, Esra Cansizoglu, Matt Higger, Bruna Girvent, Paula Gonzalez and many others who created a warm, friendly environment during my research.

This research was supported by National Science Foundation (NSF) (ECCS-0929576, ECCS-0934506, IIS-0934509, IIS-0914808, BCS-1027724, CNS-1136027, IIS-1149570, CNS-1544895), National Institute on Disability and Rehabilitation Research (NIDRR) (90RE5017), and National Institute of Health (NIH) (R01DC009834).

Words cannot express my gratitude towards my beloved parents who supported me through every stage of my life. They have been my role models and taught me how to live, love and appreciate family and life. My dad for being the kindest man ever and my mom for being the most selfless angel. My sisters, Shiva, Bita, Gita and their families with whom I have shared my laughs and tears. I would like to thank them all for their love and support.
Abstract of the Dissertation


by

Hooman Nezamfar

Doctor of Philosophy in Electrical and Computer Engineering
Northeastern University, March 2016
Dr. Deniz Erdogmus, Adviser

Losing communication and control abilities imposes many restrictions on how, when and if different tasks can be done by the affected individuals. Unfortunately, accidents and many harmful diseases, including Amyotrophic Lateral Sclerosis (ALS) impose such disabilities on individuals on a daily basis around the globe. Advances in technology and medicine have made it possible to know more and do more in terms of assistive technology during the past few decades. However, still most of the assistive devices are designed for specific tasks, such as typing or control.

In this dissertation, we introduce FlashLife™, a context aware language independent brain interface, suitable for everyday needs of an individual with disabilities. FlashLife™ provides control and communication abilities all through the same stimulation method using a single EEG electrode or eye tracking. In addition, use of the context information along with a probabilistic classification and decision making mechanism adds more robustness and flexibility at the same time. The stimulation paradigm provides highly accurate and fast classifications making use of short Calibration sessions. FlashLife™ provides performance estimates for each individual for different tasks taking advantage of the Calibration data. The stimulation paradigm has been put into use by different applications to do different tasks. A short list of applications is, FlashType™ for typing, FlashNav™ for navigation, FlashGrab™ for object manipulation and FlashPlay™ for entertainment in a virtual environment.
Chapter 1

Introduction

Noninvasive brain computer interfaces (BCIs) based on electroencephalography (EEG) mostly use brain activities which can be categorized into main groups of, event related potentials (ERPs), visually evoked potentials (VEPs) and event related desynchronization synchronization (ERDS) or volitional cortical potentials (VCPs). Each category has its own advantages and disadvantages and mostly has been utilized towards specific tasks. ERP is the response of the brain to recognition of a target stimulus, visual, auditory or tactile. A primary component of these ERPs is the P300, which occurs about 300 millisecond after stimulus onset. VEPs are the response of the visual system to a flashing stimulus. VEPs are faster and their main component, P100, appears about 100 milliseconds after the flash onset. Once the stimulus becomes steady, such as a flashing light with a constant frequency, VEPs also become steady, hence called Steady State Visually Evoked Potentials (SSVEPs). SSVEPs are usually stronger than VEPs in response to isolated flashes. Unlike the P300 responses, SSVEP responses do not solely depend on the event recognition, they also depend heavily on gaze, eyesight and attention. ERDSs, are the responses of the brain to imagining a muscle movement like making a fist or moving a foot. These responses usually take a few seconds to appear and heavily depend on the user experience. Furthermore, with the low resolution of EEG signals, usually the number of recognisable muscle movements are less than five. Mostly limited to movement in the arms, feet and tongue. Among all the responses, motor imagery responses are the slowest but they do not need any eyesight. In addition to the response time, user training and system calibration are also factors to take into account while comparing different methods.

VEPs were first discovered in the 1930’s [4]. While the studies have continued since, these responses became an attractive source for BCIs in the 1980’s. Among all the mentioned brain response categories, VEP signals are usually faster and stronger. They are also less prone to artifacts.
CHAPTER 1. INTRODUCTION

like eyeblinks due to the fact that the occipital lobe is located at the back of the head and it has the farthest distance from the eyes, while ERPs are mostly generated in the central lobe. These properties make VEPs a very suitable choice for brain computer interfaces.

In this work, we design FlashLife™, a context aware code-VEP based BCI, with a probabilistic modeling approach to achieve high decision accuracy and fast response. Differently from the usual BCI systems, in FlashLife™, we try to use the same interface towards different applications. We utilize the interface towards applications providing tools for the control and communication needs of an individual in everyday life. Based on a survey collected from the actual BCI users in a virtual forum [5], their top priority is to have a reliable and robust system rather than a fast system prone to make many errors. Having this in mind, making use of a fast and reliable BCI system towards different applications can potentially improve the quality of life of BCI user. BCIs offer alternative means of communication, hence, they differ from the usual user interfaces, having an extra stimulation element. This makes it harder to get used to BCIs as opposed to other input modalities. Having the same stimulation method results in shorter training time, in addition, it would be easier to build the habit of using the interface. One of the main factors greatly affecting the performance of a BCI system, specifically a VEP based BCI is consistency in user behaviour.

While designing a BCI system factors such as stimuli performance and user experience should be considered. In addition, we try to estimate user performance for a specific task using the probabilistic model trained by the data gathered from the same individual during a Calibration session. In BCI systems, choosing an appropriate brain response plays an important role and factors such as stimuli design and performance evaluation should be considered. There are many factors that can affect the VEP responses, including but not limited to, age, eyesight, gaze control, attention, environment lightening and stimulus contrast. Studying the nature of the responses needs many controlled experiments and yet the responses can change from time to time and from person to person. The simulation capability provides the ability of fine tuning system parameters on individual bases. The immediate advantage would be eliminating the need to have several data collection sessions. The long term advantage might be finding common features that might help eliminate the calibration session.

Using VEPs as the main input modality in FlashLife™, we will proceed with a more in depth exploration of VEPs as the visual stimuli.
CHAPTER 1. INTRODUCTION

1.1 Visually Evoked Potentials

VEPs are the responses of the visual system to flashing stimuli [6, 7]. In the event that the flashing source follows a steady pattern, the responses in the visual cortex also become steady. In other words, when there is a flashing stimulus in the visual field, for example a blinking light with a constant frequency, the oscillations also show up in the amplitude of the electrical signals recorded from visual cortex with the same frequency and its harmonics resulting in SSVEPs.

Visual cortex response to a flash, VEP, starts to show up somewhere between 80 to 120 milliseconds after the flash onset [4, 8]. The response consists of several components, but mostly known as its main component P100 or P1, first discovered by Spehlmann in 1965 [9]. A series of P1 responses can lead to a steady state response. Comparing with ERPs where the primary response, P300 or P3, shows up about 300 milliseconds after the stimulus onset, VEP based systems have the potential to be faster. However, factors such as the usable frequency bandwidth, tiring effects, dependency on gaze and eyesight impose some limitations. There are two methods mainly used to take advantage of this phenomenon and create a BCI system. In the following sections we will describe them briefly.

1.1.1 Frequency Based Stimulation

Frequency based stimulation started with the discovery of the steady state visually evoked potentials. It uses constant or slowly varying frequencies for flashing and stimulating the user’s visual system. This method, is fast and it requires very short or almost no calibration. It has been used in visual system studies and also BCIs [10, 11]. Although, the principle sounds simple, but it has its own complications.

First, SSVEP and especially frequency based stimulation are very sensitive to environmental noise. EEG amplifiers have a very high gain by design, typically around 15 to 20 thousand. This high gain makes it possible to pick up changes in the order of micro-volts or less, but, it also amplifies the noise with the same gain. Let’s assume the system is working based on a specific set of frequencies, if another device in the vicinity starts to work on the same frequency band or even its harmonics, the radiation noise interferes with EEG signals severely. These devices could be as simple as a cooling fan, air conditioning, mixer or more complicated like ventilators. With a wise placement of the devices like fans, some of the interference might be eliminated, however, some devices like ventilators are essential and cannot be placed farther away from the users. In such cases, device specific studies should be done to characterize the noise. Based on the results, interfering frequency
bands should be avoided in the stimuli and special filters should be added to filter out the strong noise
components.

A more general solution would be changing the stimulation frequencies based on the information
from the target environment of the users. This solution also sounds promising at first, but, there
are more factors that are involved in the frequency set selection for the BCI system. The dominant
one, would be the frequency response of the visual cortex of the users. Unfortunately, in addition
to the limited bandwidth, the visual system frequency response is not constant over the frequency
bandwidth and it is not similar from one user to the other. The second limiting factor is the harmonics.
Visual cortex, creates the fundamental frequency response and its harmonics in response to a single
stimulation frequency. Figure 1.1 shows the fundamental frequency and its harmonics in response to
a 15 Hz flickering LED. These harmonics and their strength are different from one user to the other.

Figure 1.1: Frequency response of Oz channel in response to 15 Hz flickering LED stimulus

Most of the time, the first couple of harmonics are strong, but there are cases that they fill the whole
bandwidth, extending up to the 8th harmonic as shown in Figure 1.1.

Although VEP response is fast, most frequency based BCI systems need a few (~ 4) seconds
worth of EEG signals to extract the target frequency response reliably. In general, a good window
length is somewhere from two to four seconds. One should note that even though the two second
window might seem long, it contains several repetitions of the stimulus, comparing to a P300 or motor
imagery in which every repetition takes a few seconds. SSVEP responses are heavily dependent on
user attention as well. Looking at checkerboards flipping their patterns with a constant frequency could be very boring and tiring for the users, causing them to deviate their attention from the stimulus, leading to a sharp decrease in performance.

For EEG, studies show that visual cortex frequency response falls in the 5 to 100 or 120 Hz in most cases [12]. Considering a 100 Hz bandwidth along with second and third harmonics of each stimulus frequency, finding a few frequencies that do not overlap with other stimulus frequencies and their harmonics becomes almost impossible when the number of stimulus frequencies increase. Especially, when the flickering restrictions of the displays are added, the number of feasible frequencies becomes even less [13]. Jia et al. in [14] have tried using two frequencies for each stimulus in order to overcome the problems with the harmonics and noise. While the results might show a slightly better robustness comparing to systems that use single frequencies, still the problem with the limited number of feasible frequency sets exists.

1.1.2 Code Based Stimulation

Code based stimulation or code visually evoked potentials (code-VEP), are a very promising way of using VEP signals to build fast, reliable BCIs with more possible options. In this method of stimulation, the flashing or flipping pattern of the stimulus is controlled by a binary sequence, control bit sequence, where 1s are assigned to one pattern and 0s to the other. Different stimuli shapes can be used. Figure 1.2 shows the two different patterns of a checkerboard stimulus and an m-sequence of length 31 bit as the control bit sequence. The control bit sequences act very similar to code-words in

![Figure 1.2: a) checkerboard stimulus pattern one, b) checkerboard stimulus pattern two, c) a pseudo-random binary control sequence of length 31 bit](image-url)
CHAPTER 1. INTRODUCTION

a communication system. Similarly, taking the code-word considerations while choosing the control bit sequences has a direct impact on the performance of the stimuli.

Among the possible bit sequence patterns, some sequences such as Maximum length sequences (m-sequences) [15] and gold sequences [16] show a superior performance. These sequences are also widely used in the communication systems [17]. Gold sequences are generated using two different m-sequences and have more variations available with the same length as opposed to their counterpart m-sequences. Gold sequences share almost similar properties as the m-sequences. Gold sequences can be used when the m-sequences of the same length do not have the required number of variations. The improve in the performance comes from the neat properties of these sequences. Some of the properties of m-sequences are as follows. These sequences are almost balanced in the number of zeros and ones. By definition, the length of an m-sequence is always an odd number. No matter what m-sequence is chosen, the difference in the number of zeros and ones is always only one. More accurately, in an m-sequence of length \(2^n - 1\) bit, where \(n\) is the number of binary bits used to generate the m-sequence, there exists \(2^{(n-1)}\) ones and \(2^{(n-1)} - 1\) zeros. They also have a very low auto-correlation at any lag other than zero. The cross-correlation between different m-sequences of the same length is also small. Due to the pseudorandom nature of these sequences, they can be considered broadband. This property results in the utilization of the whole visual system bandwidth available to the system. Figure 1.3 shows the frequency response of the Oz channel to four different checkerboard stimuli controlled by four different m-sequences of length 31 bit and a sample of the m-sequences used. Frequency based stimulation can be achieved by assigning consecutive zero and ones as the control bit sequence and adjusting the bit presentation rate to the desired value.

Sutter, a pioneer in code-VEP BCI, used one m-sequence and its time lagged versions to build a brain computer typing system [18, 19], the results were exceptional, although the system was tested on only one subject with implanted electrodes and some have reported the results unreproducible [20]. Having the electrodes implanted on or in the cortex increases the signal to noise ratio, prevents electrode displacements and provides higher resolution, but, being invasive it carries high procedure and infection risks. Being located on the skull or the cortex these complications can become fatal very quickly. Hence, non-invasive EEG based systems have become a source of attention for the past few decades [20, 21].

In FlashLife™ while there are no technical limits on the length of the m-sequences, m-sequences of short lengths e.g. 31 or 63 bits are often used. The combination of these m-sequence lengths and bit presentation rates, mostly translates to trial lengths of 0.5 to 1 seconds. Use of different m-sequences in addition to providing more shifted lags if needed has the advantage of also being less
CHAPTER 1. INTRODUCTION

Figure 1.3: Frequency response of the Oz channel to stimulation with four different m-sequence of length 31

tiring for the user. Code-VEP systems are highly desirable due to their fast response time and high accuracy, but they also need a calibration session. The duration of the calibration session depends on the bit presentation rate. More details about the Calibration and system parameters are discussed in Section 2.4.
Chapter 2

Real-Time System Design

In this chapter the considerations and choices made while designing different building blocks of FlashLife™ are described. In addition, the real-time operation, and modes of operation of FlashLife™ are explained. The complete system consists of the following parts, stimuli presentation, data acquisition, signal processing and action execution.

2.1 User Interface

A general user interface is designed. User interface contains two parts, stimuli and application. Typically, a computer display is utilized to present the user interface. A general sketch with some placeholders is presented in this section and more detail about the application part of the user interface is provided in Chapter 4. Figure 2.1 illustrates the general theme of the user interface. In this figure, the central blue area is reserved for application visualization and feedback. The purple rectangles are the placeholders for the stimulus labels. Labels are assigned based on the stimulus role in every application. Based on the application needs labels can be static or dynamic.

Four checkerboards at the four corners of the display perform the role of stimuli. This spatial configuration is chosen to utilize the display more efficiently. Simple but important aspect of this stimuli placement are maximizing the inter stimuli distance and keeping the symmetric placement intact. Each stimulus is assigned a square are where the size of the sides is equal to one third of the screen height. This special size maximizes the individual stimulus sizes and inter stimulus distances. A yellow colored frame around a stimulus, is the indication of the that stimulus being the target stimulus during the Calibration task or being detected as the intended stimulus in the other operation modes. This is the immediate feedback provided to the user as what the system has
CHAPTER 2. REAL-TIME SYSTEM DESIGN

detected. Depending on the target application more feedback is provided to the user as what the
system will do in response to the detected stimulus. After the actions to an intended stimulus are
performed the frame color becomes darker to indicate the past decision.

![General user interface theme](image)

Figure 2.1: General user interface theme, (a) Default stimuli, (b) Alternative stimuli for color blind.
Purple and blue areas are just place holders and they are replaced by stimuli labels and application
screen respectively.

2.2 Stimuli

Stimuli as the input modality of a BCI systems play a crucial role in the overall performance. In
other words, the stronger the stimuli the stronger the response would be. Factors that can affect the
effectiveness of the stimuli depend on the type of stimulation. For example, implementation of the
oddball paradigm in an ERP based BCI has a direct effect on the initiation of the brain activity in
response to the stimuli [22]. However, in VEP based stimuli oddball effect doesn’t apply and other
factors such as stimuli shape, texture, size and color are important.

FlashLife™ uses code-VEP as the main input modality, so for the rest of this chapter we focus
on the design and optimization of the VEP stimuli. Although VEPs are the responses of the visual
cortex to flashing stimuli, but, not every flashing stimulus generates the same strength response. The
strength of the response depends on a few factors which can be divided to environmental related
and stimulus specific factors. Environment illumination is one of the environmental factors that can
possibly affect stimuli effectiveness. In an environment too bright, it would be harder to distinguish
a flashing source, hence the effect of a flashing stimuli might decrease. On the other hand, darker
environments increase the afterimage [23] effect caused by the flashing sources. Stimulus specific
factors include the shape of each stimulus, the contrast between the bright and the dark pattern, the
CHAPTER 2. REAL-TIME SYSTEM DESIGN

distance between flashing parts of the stimuli, the distance of the flashing parts from the center of the visual field and finally the distance of the stimuli from the user’s eye.

2.2.1 Shape & Texture

Historically, different shapes have been used to induce SSVEP response. Zhu used flashing arrows [24], Edlinger used flashing icons and letters [25] and Sutter [18] have used flashing checkerboards. Among all the mentioned methods, flashing checkerboards have shown to induce a stronger response [26]. This can partly come from the fact that a checkerboard pattern stimulates cells in response to the lighter and darker colors at the same time. And partly from the fact that checkerboards, having parameters such as size and number of blocks, provide more degrees of freedom and can be fine tuned.

Checkerboards or in general VEP stimulus can be presented in two ways. First, switching between a checkerboard or stimuli pattern and a blank screen as the alternating pattern. This method translates to the On, Off pattern, during which a checkerboard or symbol is shown during the On time and a blank screen is presented during the Off time. Second, switching between a checkerboard and its reversed pattern with opposing colors as the second pattern. The latter generates a stronger response since it stimulates the cells in the retina twice as much. As a result, in frequency based stimulation, while a constant frequency is used to control the stimulus, the second harmonic appears stronger than the others. However, in a checkerboard pattern stimuli, more sensitivity to eye movements is observed. Simply, if the users shift their point of attention (gaze) by one block it will reverse the effect on the response, causing a disturbance in the brain responses to the stimuli. This sensitivity is even higher in the code-VEP systems since the gaze shift by one block is equivalent to inverting the control bit sequence.

The next factor is the size. One might think that the larger the size of the flashing stimulus the larger the effect would be. However, it won’t be true for all the stimulus types. There is a need for a systematic study on the optimum stimuli size. However, through pilot experiments with checkerboard stimuli with three different block sizes, large, medium and small, we observed a sharp decrease in the performance when the blocks were too big or too small. In this study the stimuli area were kept constant. For all three block sizes, square checkerboards with the same total area were used and the block sizes and accordingly the number of blocks in each checkerboard were varied. Keeping the stimuli area constant results in constant average illumination. Overall results showed the medium checkerboard block size creates a stronger response. The medium size block was 1.75 centimeters or...
CHAPTER 2. REAL-TIME SYSTEM DESIGN

approximately, the width of a thumbnail. Participants sat at 60cm distance from the screen. In this setting, the block sizes translate to approximately 1.6 degrees in the visual field. Anecdotal outcome of the experiments we have collected is, the larger the blocks in the checkerboards the more irritating their flashes get. Based on [27][28], width of the thumbnail held at an arm length is about 1.5 degrees. This could be correlated with the distance of the retina cells. In [29] Yoshor et al. show that the smallest size of the Receptive Field measured in the early visual cortex is one degree.

2.2.2 Conformation

Visual field is considered to be symmetric with respect to the vertical axes. With good approximation, the visual field can be considered to have circular or elliptical symmetry. Having said that, it seems like a good idea to put the same level stimulus on the same distance from the center. On the contrary, Sutter, [19] has shown that most of the visual effect comes from the center two degrees of the visual field. Figure 2.2 shows this effect, which can be approximated by a \(\frac{1}{r}\) factor. Where \(r\)

![Figure 2.2: Retina response density [1].](image)

is the radial distance from the gaze point. So for the stimulus to have the maximum effect, it has to fall in that area. Being bounded to the display size, we have used an 18 to 22 inch display at the 60 centimeters distance, it would fall in the 22 to 30 degrees of the visual field. When it comes to the stimulus orientation, one the most important factors would be the user capabilities and the fact that they have overt orientation or not. For those who have overt orientation, the response will be much stronger compared to the group that only have covert attention. This comes from the gaze dependency of the SSVEP response. Stimuli for covert attention should fit in the ten degrees of the visual field to create maximum effect [30]. Overt attention or orientation is the selective movement of the eyes to focus on a target while covert attention is the involuntary movements or no movements in the eye to focus on the target [31]. The size of the stimulus also matters. Studies had been done to find the effects of the size of the stimulus and the perspective effects [2]. The results showed a stronger response to an object of the same angular size farther in the perspective, comparing with an
CHAPTER 2. REAL-TIME SYSTEM DESIGN

object of the same angular size but closer in the perspective. Figure 2.3 shows the same angular size stimulus ball in the front and back of the perspective, the response of the visual cortex to the ball in the back covers a larger area of V1 cortex compared to the response to the front stimulus. This finding has been challenged by Don McCready [32].

![Figure 2.3: Perspective effect on the stimulus strength](image)

2.2.3 Color & Contrast

Stimulus color and contrast, can affect the strength of the VEP response generated in response to the stimuli. One of the factors making a flash distinguishable is the contrast between the two colors. Intuitively, the higher the contrast the more distinguishable the colors get. However, the long term effect also should be considered. A key point while adjusting the contrast is to avoid creation of the afterimage effect. Afterimage, is the remaining effect of a flash on the retina cells [23]. The intensity of the afterimage effect is proportional to the amount of light dispensed from the stimulus, for example a camera flash while taking a close shot from a person’s eye will create a strong after image effect, which might even take several seconds to clear. Considering the afterimage effect, the brightness and the contrast of the display should be adjusted appropriately. In the studies done in this work an LCD display or a laptop of size 18 to 22 inch has been used with a minimum distance of 60cm between the display and participants’ eyes. Using a standard display keeps the brightness and contrast values in a safe range for the human visual system. Earliest guidance on the color pair in the BCI feild comes from Sutter in [19] who has mentioned observing stronger responses using opposite color pairs such as black & white or red & green where the later response might be stronger.
A more detailed study of the effect of opposite color pairs of black & white, red & green and blue & yellow, showed that the responses generated using the red & green stimuli are stronger and more comfortable for the users [33].

2.2.4 Presentation rate

Presentation rate is defined as the number of stimuli patterns (controlled by bits of the control bit sequence) presented per second. Bit presentation rate directly affects the trial length. Trial length in code-VEP based system is equal to the number of bits in the control bit sequence divided by the bit presentation rate. The bandwidth of the human visual system is limited to about 100 Hz [12], and is one of the limiting factors for BCIs with visual stimuli. Studies on the human visual system show the frequency response is stronger around 15 Hz. On the other hand, low frequency flicker being more visible, is more irritating to the users. Furthermore, the risk of inducing photoepileptic seizure increases for frequencies between 15 and 25 Hz, compared to higher stimulation frequencies [34].

Use of the computer displays as the stimulation media provides more flexibility but, also imposes more restrictions on the available frequency sets. Being limited by the refresh rate of the computer display, the highest stimulation frequency possible is equal to half of the refresh rate. For the frequency based systems, this means that only frequencies that are divisors of 60 can be used. In the code-VEP system, it is the maximum rate of the bit presentation. One method to overcome the refresh rate limitation is using LEDs or LED arrays as the stimuli. Using LEDs there is virtually no limitation on the stimulus frequency targeted for the human visual system. However, it imposes some restrictions on the stimulus shape and less flexibility is available.

In addition to the stimuli hardware limitations, environmental noise and more specifically the AC power line noise should be considered. Typically, the interference from the AC power line shows up as a strong pick around the carrier frequency, 50 or 60 Hz depending on the country and less strong pick at the harmonics. A usual way of eliminating this noise is applying a notch filter on the carrier frequency. This interference source should be considered while deciding on the stimuli presentation rate and also stimuli frequency set in the case of frequency based stimuli.

VEPs being rapid responses makes the stimuli onset timing an important factor while choosing the target response window of the EEG signals. Sensitivity to the timing for the methods in time domain, such as template matching, is more comparing to methods applied in frequency domain like power spectral density. The sensitivity becomes even more obvious when the bit presentation rate increases. As the bit presentation rate increases the duration of time between the events, hence, the
number of associated EEG samples decreases.

2.2.5 Design & Optimization

To choose the optimum combination of the stimuli parameters two set of experiments have been performed. The initial set is to compare effects of the bit presentation rates of 15, 30 bits per second (bps) using different control bit sequences of length 31 bit [35]. The second set of experiments compare the effects of the bit presentation rates of 30, 60 and 110 bps along with the effects of the opposite color pairs of Black & White, Red & Green and Blue & Yellow.

2.2.5.1 Part 1

Starting from the initial set of experiments, the choices of bit presentation rates of 15, 30 bps and control bit sequence length of 31 bit, results in trial lengths of 2.06 and 1.03 seconds respectively. Data collected from four healthy participants from 22 to 28 years old, with normal or corrected to normal vision, based on an approved Northeastern University IRB was processed. Data was collected from 16 scalp locations based on the 10 − 20 standard, O2, Oz, O1, PO4, POz, P4, P2, Pz, P1, P3, Cp2, Cp1, C4, Cz, and C3. Having a VEP stimuli, electrode locations were selected to have a higher density over the occipital lobe. Figure 2.4 shows the corresponding scalp electrode location on the 10-20 standard.

![Figure 2.4: Electrode locations based on the 10-20 standard.](image)

Each individual, participated in two data collection sessions. Bit presentation rate was the only variation from one session to the other. Participants completed two Calibration sessions. In this
CHAPTER 2. REAL-TIME SYSTEM DESIGN

experiment *Calibration* was performed in a serial manner. In other words, stimuli were presented to the user one at a time, in the center of the screen using $8 \times 8$ checkerboards of size $14 \times 14$ cm with the block side of 1.75 cm. Stimuli were presented using the Psychophysics Toolbox (Brainard, 1997; Pelli, 1997; Kleiner et al, 2007). Participants were sat at a 60 cm distance from the center of the screen. The stimuli were covering approximately 20 degrees of the visual field. There were no restrictions on the head or eye movements.

Participant performances over each *Calibration* session is evaluated using the classification method described in Section [3.1.1](#). A closer look at the single channel classification performance shows a slight increase for the bit presentation rate of 30 bps. The overall single channel classification performance based on the data collected from four participants is summarized in Tables 2.1 and 2.2. This increase in performance can be due to the fact that with a faster presentation rate trials are shorter and it is easier for the participants to keep their attention on the stimulus. Single channel classification results for each participant are demonstrated over the scalp locations in Figure 2.5.

### Table 2.1: Single channel classification performance, stimuli presented with bit presentation rate of 15 bps.

<table>
<thead>
<tr>
<th>Channel Placement</th>
<th>Min</th>
<th>Max</th>
<th>Average</th>
<th>Std.</th>
</tr>
</thead>
<tbody>
<tr>
<td>C3</td>
<td>36.5</td>
<td>84.5</td>
<td>55.5</td>
<td>7.39</td>
</tr>
<tr>
<td>CZ</td>
<td>44.75</td>
<td>63.75</td>
<td>53</td>
<td>7.13</td>
</tr>
<tr>
<td>C4</td>
<td>38.5</td>
<td>83.75</td>
<td>55.75</td>
<td>6.20</td>
</tr>
<tr>
<td>CP1</td>
<td>39.5</td>
<td>56.25</td>
<td>47.25</td>
<td>11.67</td>
</tr>
<tr>
<td>CP2</td>
<td>23.5</td>
<td>57.25</td>
<td>43</td>
<td>6.27</td>
</tr>
<tr>
<td>P3</td>
<td>19</td>
<td>84.25</td>
<td>51.25</td>
<td>10.16</td>
</tr>
<tr>
<td>P1</td>
<td>22.75</td>
<td>67</td>
<td>45</td>
<td>10.15</td>
</tr>
<tr>
<td>PZ</td>
<td>47.75</td>
<td>79.75</td>
<td>64.25</td>
<td>10.21</td>
</tr>
<tr>
<td>P2</td>
<td>40</td>
<td>99</td>
<td>63.75</td>
<td>6.30</td>
</tr>
<tr>
<td>P4</td>
<td>36.75</td>
<td>68</td>
<td>53.25</td>
<td>5.64</td>
</tr>
<tr>
<td>PO3</td>
<td>39.75</td>
<td>73.25</td>
<td>52.5</td>
<td>12.28</td>
</tr>
<tr>
<td>POZ</td>
<td>48</td>
<td>87.75</td>
<td>66</td>
<td>10.15</td>
</tr>
<tr>
<td>PO4</td>
<td>44.75</td>
<td>82.5</td>
<td>58.5</td>
<td>7.34</td>
</tr>
<tr>
<td>O1</td>
<td>50.75</td>
<td>99.25</td>
<td>77</td>
<td>10.48</td>
</tr>
<tr>
<td>OZ</td>
<td>48</td>
<td>95</td>
<td>73.25</td>
<td>3.96</td>
</tr>
<tr>
<td>O2</td>
<td>25</td>
<td>89</td>
<td>65</td>
<td>6.09</td>
</tr>
</tbody>
</table>

### Table 2.2: Single channel classification performance, stimuli presented with bit presentation rate of 30 bps.

<table>
<thead>
<tr>
<th>Channel Placement</th>
<th>Min</th>
<th>Max</th>
<th>Average</th>
<th>Std.</th>
</tr>
</thead>
<tbody>
<tr>
<td>C3</td>
<td>33</td>
<td>82</td>
<td>55.75</td>
<td>9.31</td>
</tr>
<tr>
<td>CZ</td>
<td>36.75</td>
<td>69.25</td>
<td>55.5</td>
<td>10.73</td>
</tr>
<tr>
<td>C4</td>
<td>38</td>
<td>87.75</td>
<td>61.25</td>
<td>10.09</td>
</tr>
<tr>
<td>CP1</td>
<td>34</td>
<td>61.25</td>
<td>50.5</td>
<td>8.73</td>
</tr>
<tr>
<td>CP2</td>
<td>32.75</td>
<td>65.75</td>
<td>48.75</td>
<td>8.65</td>
</tr>
<tr>
<td>P3</td>
<td>31</td>
<td>83.5</td>
<td>56</td>
<td>10.56</td>
</tr>
<tr>
<td>P1</td>
<td>37</td>
<td>71</td>
<td>54.25</td>
<td>8.00</td>
</tr>
<tr>
<td>PZ</td>
<td>42</td>
<td>86.25</td>
<td>65.25</td>
<td>7.50</td>
</tr>
<tr>
<td>P2</td>
<td>41</td>
<td>96.75</td>
<td>66.5</td>
<td>6.47</td>
</tr>
<tr>
<td>P4</td>
<td>40</td>
<td>72.25</td>
<td>60.5</td>
<td>7.67</td>
</tr>
<tr>
<td>PO3</td>
<td>41.75</td>
<td>88.25</td>
<td>59</td>
<td>9.29</td>
</tr>
<tr>
<td>POZ</td>
<td>51</td>
<td>94</td>
<td>67.25</td>
<td>8.13</td>
</tr>
<tr>
<td>PO4</td>
<td>41</td>
<td>87.25</td>
<td>61</td>
<td>6.92</td>
</tr>
<tr>
<td>O1</td>
<td>52.25</td>
<td>96.75</td>
<td>77.25</td>
<td>6.32</td>
</tr>
<tr>
<td>OZ</td>
<td>55.75</td>
<td>99</td>
<td>74.75</td>
<td>1.71</td>
</tr>
<tr>
<td>O2</td>
<td>37.25</td>
<td>92.75</td>
<td>67.75</td>
<td>7.40</td>
</tr>
</tbody>
</table>

**2.2.5.2 Part 2**

Next, we proceeded with a more extensive study with the aim of optimizing the stimuli parameters of color, bit presentation rate and control bit sequence length to achieve well performing and
comfortable stimuli. In human eye, retina, i.e. the light-sensitive layer, is responsible for light perception. It contains two different types of photoreceptors: rods and cones. Rods are more in number than cones, have poor acuity and are color blind. They are suitable for human perception in low luminance conditions. In contrast, cones are color sensitive and are responsible for vision in daylight. According to the opponent-process theory of color vision [36], cones feed three neural channels for color processing. Each channel is composed of opponent color pairs: yellow/blue, red/green and black/white. Responses to one color of an opponent channel are antagonistic to those of the other color. In other words, opponent colors are never perceived together. Black-white visual stimulation, mostly stimulates rods. In this study, we wanted to examine the hypothesis that using opponent colorful pairs, cones and rods are both stimulated, more information is transferred to the visual cortex, leading to a higher classification performance.

Based on anecdotal feedback from the individuals exposed to both frequency based and code based stimulation, code based stimulation is less tiring as opposed to f-VEP based stimulation, hence, we proceeded using the performance of our c-VEP based system as the performance measure. The goal was to find a set of stimuli parameters which can increase or maintain a high performance while being less tiring for the user.

Two reversed pattern $5 \times 5$ checkerboards have been used as the visual stimulus. Each checkerboard pattern consists of 25 units of opponent colors covering a $10\text{cm} \times 10\text{cm}$ area. Four visual stimuli were placed at the four corners of a $51 \times 29 \text{ cm}$ computer display. Figure 2.6 shows two revered pattern checkerboards and a sample stimuli screen. Participants were seated with their
CHAPTER 2. REAL-TIME SYSTEM DESIGN

<table>
<thead>
<tr>
<th>m-sequence length (bit)</th>
<th>Bit presentation rate (b/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>30</td>
</tr>
<tr>
<td>31</td>
<td>B/W</td>
</tr>
<tr>
<td>63</td>
<td>B/W, R/G, B/Y</td>
</tr>
<tr>
<td>127</td>
<td>B/W, R/G, B/Y</td>
</tr>
</tbody>
</table>

Table 2.3: Color pair, bit presentation rate and m-sequence length triplets studied. B/W represents black and white, R/G represents red and green and B/Y represents blue and yellow color pairs.

head fixed, such that the distance from their eye to the stimulation screen was 80 cm. A different m-sequence was assigned to each stimulus, as the control bit sequence. To ensure visual stimulus transitions occurred precisely at the intended times, the monitor refresh rate was set to 60 Hz for the 30 and 60 bps bit presentation rates, and to 110 Hz for the 110 bps bit presentation rate. Four stimuli were presented simultaneously. A complete presentation of the checkerboard patterns according to their control bit sequence is considered a trial. The onset of each trial is marked using a hardware trigger signal sampled simultaneously with EEG signals. Nonstop presentation of 12 trials is considered an epoch during the Calibration session. Each Calibration session consisted of 20 epochs, where each stimulus was randomly selected as the target of the epoch 5 times. By eliminating the first and last trial of every epoch, every Calibration session results in 50 trials per stimulus. The number of trials per stimulus was determined based on the results of a previous study [35, 37] and proven to be adequate. The study consists of 12 different Calibration sessions.

![Figure 2.6](image)

Figure 2.6: (a) The checkerboard pattern corresponding to Bit ”0”, (b) The checkerboard pattern corresponding to Bit ”1”, and (c) The target arrangement of the stimuli.

Table 2.3 shows different color pair, bit presentation rate and control bit sequence length triplets used. Sessions with control bit sequences of length 31 bit were only collected for the black and white color pair to limit the duration of the data collection session. Table 2.4 shows the single stimulation trail duration under different settings. The Calibration sessions corresponding to the black and white color pair were collected first, followed by the Calibration sessions for the red and green color pair,
CHAPTER 2. REAL-TIME SYSTEM DESIGN

<table>
<thead>
<tr>
<th>m-sequence length (bit)</th>
<th>Bit rate (b/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>30</td>
</tr>
<tr>
<td>31</td>
<td>1.3 s</td>
</tr>
<tr>
<td>63</td>
<td></td>
</tr>
<tr>
<td>127</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.4: Stimulation trial duration.

<table>
<thead>
<tr>
<th>Bit rate (b/s)</th>
<th>Fundamental frequencies (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>15, 10, 7.5, 6, 5, 4.3, 3.75</td>
</tr>
<tr>
<td>60</td>
<td>30, 20, 15, 12, 10, 8.6, 7.5</td>
</tr>
<tr>
<td>110</td>
<td>55, 36.6, 27.5, 22, 18.3, 15.7, 13.75</td>
</tr>
</tbody>
</table>

Table 2.5: Fundamental frequency components generated by each m-sequence under different bit presentation rates. The frequency sets are identical for all the m-sequences used no matter the length, except the underlined values which are only available for the m-sequences of length 127 bits.

and ending with the sessions for the blue and yellow color pair. For every color pair, the calibration sessions were started with the shorter bit sequence length and slower bit presentation rate and proceed by increasing the bit presentation rate. For every m-sequence length, 4 different m-sequences were chosen such that their cross correlation is minimized. To further minimize the cross-correlation between the chosen m-sequences of each length, the pairwise Hamming distance [38] between all the pairs and lags of the 4 different m-sequences were studied and the appropriate lags were applied. Table 2.5 illustrates the fundamental frequencies induces by each m-sequence under different bit presentation rates. EEG signals along with the trigger signal indicating the onset of the events were collected using active g.Butterfly electrodes, a g.Gammabox, and a g.USBamp by G.tec. A notch filter at 60 Hz was used to eliminate the effect of AC power line noise and a bandpass filter from 0.1 to 100 Hz was applied to eliminate DC drifts in the signals. A sampling rate of 256 Hz has been used. With the focus of the study on the responses to the visual stimuli, EEG sites were selected with higher density around the visual cortex at Oz, O1, O2, Po3, Poz, Po4, P1, P2, Cpz, Po7, C4, C3, Fz, Cz, Po8, and Pz based on the 10-20 standard.

Six healthy subjects, 3 male and 3 female, with normal or corrected to normal vision ranging in age from 23 to 27 participated in this study after having passed the Ishihara color blindness test [39]. Each individual consented, and participated in a two hour data collection session where the data was collected following an approved Northeastern University IRB.
Figure 2.7: Single channel classification accuracy of choosing among 4 simultaneously presented stimuli under different stimuli settings presented as a spacial scalp distribution. Every row corresponds to one participant. B/W, R/G and B/Y represent black and white, read and green and blue and yellow opponent color pairs.

To separate the effects of the parameters under study, EEG channels were processed individually using the method described in Section 3.3. In addition, multi-channel classifications with a structured and regularized covariance matrix [40] were performed and did not show any performance increase as opposed to the single channel accuracy of the channel Oz. This can be due to information content of the poor performing channels being very low. Figure 2.7 presents the single channel classification accuracy for all the participants and conditions.

The closer the EEG sites are to the visual cortex the higher their classification accuracy will be. However, the drop in performance is slower for sites on the center. Table 2.6 shows the classification accuracy of the channel Oz for all the participants under different conditions. Located on the center of the visual cortex, channel Oz was previously reported as the best performing channel for black and white color pair [37], our results confirm that for all the participants and under all the conditions, Oz is the best performing channel.

Based on the results presented in table 2.6 red and green color pair, 60 bps bit presentation rate and 63 bit long m-sequences result in the highest performance on average. This setting, has the smallest standard deviation among all the other settings, showing the best consistency overall. A few paired t-tests were performed to find the significance of different opponent color pairs among
CHAPTER 2. REAL-TIME SYSTEM DESIGN

<table>
<thead>
<tr>
<th>Participant ID</th>
<th>B/W 31 bit</th>
<th>B/W 63 bit</th>
<th>B/W 63 bit</th>
<th>B/W 127 bit</th>
<th>B/W 30 b/s</th>
<th>B/W 60 b/s</th>
<th>B/W 110 b/s</th>
<th>B/W 60 b/s</th>
<th>B/W 110 b/s</th>
<th>B/W 110 b/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>P01</td>
<td>80</td>
<td>82</td>
<td>71</td>
<td>79</td>
<td>97.5</td>
<td>91</td>
<td>89.5</td>
<td>85</td>
<td>89.5</td>
<td>87</td>
</tr>
<tr>
<td>P02</td>
<td>94.5</td>
<td>89</td>
<td>72</td>
<td>97.5</td>
<td>100</td>
<td>97.5</td>
<td>88</td>
<td>97.5</td>
<td>97.5</td>
<td>99.5</td>
</tr>
<tr>
<td>P03</td>
<td>54</td>
<td>85</td>
<td>40.5</td>
<td>65</td>
<td>97.5</td>
<td>86.5</td>
<td>40.5</td>
<td>79</td>
<td>65</td>
<td>61.5</td>
</tr>
<tr>
<td>P04</td>
<td>96</td>
<td>91.5</td>
<td>82</td>
<td>98.5</td>
<td>99.5</td>
<td>98</td>
<td>90</td>
<td>85.5</td>
<td>89</td>
<td>97</td>
</tr>
<tr>
<td>P05</td>
<td>100</td>
<td>98</td>
<td>89.5</td>
<td>99</td>
<td>96.5</td>
<td>97</td>
<td>98.5</td>
<td>92.5</td>
<td>98.5</td>
<td>100</td>
</tr>
<tr>
<td>P06</td>
<td>94</td>
<td>83</td>
<td>75</td>
<td>96</td>
<td>100</td>
<td>97</td>
<td>94</td>
<td>97.5</td>
<td>91.5</td>
<td>94.5</td>
</tr>
<tr>
<td>AVG</td>
<td>86.41</td>
<td>88.08</td>
<td>71.66</td>
<td>89.16</td>
<td>98.5</td>
<td>94.5</td>
<td>83.41</td>
<td>89.5</td>
<td>88.5</td>
<td>89.91</td>
</tr>
<tr>
<td>STD</td>
<td>17.26</td>
<td>6.05</td>
<td>16.77</td>
<td>14.05</td>
<td>1.51</td>
<td>4.69</td>
<td>21.36</td>
<td>7.52</td>
<td>12.19</td>
<td>14.7</td>
</tr>
</tbody>
</table>

Table 2.6: Performance of channel Oz based on the calibration data under different colors, bit presentation rates and control bit sequence lengths. Best performance for each individual is marked in bold. The last two rows are column wise average and standard deviation respectively.

Figure 2.8: Paired t-tests among different opponent color pairs under different bit presentation rates and different m-sequence lengths. Electrode locations marked with * showed significance with \( p < 0.05 \).
CHAPTER 2. REAL-TIME SYSTEM DESIGN

the same channel pairs for all the participants. Figure 2.8 summarizes the results of these paired
\( t \)-tests. Channels P1, Pz, and P2 for 60 bps bit presentation rate for both B/W-R/G and B/W-B/Y
were significantly different while they did not show a significant difference for R/G-B/Y.

For the higher bit presentation rate of 110 bps, channels P1 and Pz for B/W-R/G were significantly
different and channels Pz and Fpz were significantly different for B/W-B/Y. For the bit sequences of
length 127 bits and bit presentation rate of 110 bps, only one channel, Fz, was significantly different
between B/W-B/Y. The last row of figure 2.8 shows paired \( t \)-test among all the participants and all
the sessions with different color pairs.

The overall results show a significant difference in using opponent colorful pairs as opposed
to black and white at channels O1, O2, Oz, P1, P2, Pz, FPz, Cz and Fz. No significantly different
channel was found between the opponent colorful pairs. Based on the paired \( t \)-test results, 60 bps
bit presentation rate and 63 bit m-sequence length has the most significant difference between the
opponent colorful pairs and the black and white color pair.

The significant difference in using opponent colorful pairs, can come from the fact that they
stimulate more photo-receptors (both Rods and Cones) resulting in a stronger SSVEP response.
Figure 2.9 shows the template response of channel Oz for the stimulus 1 for participants with best
and worst classification performance (participant P02 and P03 respectively). Template responses
to the colorful opponent pairs had a stronger response. While the slight decrease using 110 bps bit
presentation rate might be coming from the fact that cones are less responsive to high frequency
changes. Surveys collected from the participants after the experiment, were in agreement that the red
and green and the 110 bps bit presentation rate were the most comfortable. This choice of color is in
line with the previously reported findings [19]. Blue and yellow was the runner up and the black and
white was the last. Red and green being equiluminant is one of the main factors making them more
comfortable, especially against the black and white pair which makes flickers more visible.

In summary, one second of EEG evidence and red and green opponent color pair results in the
highest accuracy. Red and green opponent color pair also shows the least standard deviation for the
average accuracies comparing with the other color pairs under the same presentation parameters.
When the decision rate is the most important factor, with a small decrease in performance, 110 bps
bit presentation rate with the 63 bit long m-sequences can be used to make decisions in almost half a
second. On the other hand, to increase the comfort of the users, 110 bps bit presentation rate and 127
bit long m-sequences would be the best choice. Finally, a good compromise comes from using 63 bit
long m-sequences, 60 bps bit presentation rate, and using the red and green opponent color pair. This
setting, achieves the highest performance, while having the second best choice of the bit presentation
Figure 2.9: Templates response of channel Oz for the stimulus 1 under different opponent color pairs for participant P02 on the left and P03 on the right. The black, red and blue lines are representing the black-white, red-green and blue-yellow opponent color pairs respectively.

rate from the comfort point of view. A systematic study is required to find the best color pair for individuals who are color blind. However, at least the same pair of the control bit sequence length and bit presentation rate with the color pair of Black & White can be used to achieve reasonable performances for individuals who are color blind. Bit presentation rate of 120 bps was intentionally eliminated due to the overlap with the AC power line noise at 60 Hz.

2.3 Channel Selection

EEG data acquisition being a non-invasive method, makes EEG based systems favorable. However, still putting the electrodes on the scalp, applying the conductive gel uniformly, occasional signal monitoring and renewing the conductive gel and washing the hair afterwards is not easy. Many efforts are in place to make the process easier. Stronger signal processing algorithms and the advances in the technology are playing their role also. As an example, the early EEG acquisition systems were using passive electrodes which required abrasing the skin and also applying a conductive past. The next generation, active electrodes, only require a conductive gel. The conductive gel is much easier to wash off as opposed to the conductive past. The efforts are continuing by moving towards dry electrodes. However, in this trajectory, the impedance between the electrode and skin is increasing, requiring higher amplification gain and in return more susceptibility to noise and artifacts. With
today’s technology, active electrodes provide an acceptable compromise between ease of use and signal quality.

In addition, the whole setup and maintenance process becomes extremely easier and the price becomes cheaper when the number of electrodes is small. In general, the less number of electrodes the better. Thinking about the appropriate channel selection two questions come to mind; first, how many channels should be used and second, which channels should be used. EEG electrodes placed on different parts of scalp capture signals from different parts of the brain. Therefore, the physiology of the brain while deciding on the electrode locations should be considered. However, having passed through several tissue layers such as skull, the signal to noise ratio is low. In addition layer such as skull combine the signals resulting in correlation between neighbouring sites and decreasing the resolution. Hence, careful considerations should be taken into account while deciding on the electrodes and their placement.

To find minimum number of electrodes required for FlashLife™ to operate reliably, we examined the calibration data collected from four participants aging from 22 to 28 years to find out the performance of signal collected from different locations of the scalp. Detailed single channel performance results are presented for a 22 year old male and a 27 year old female. Data was collected bit presentation rates of 15 and 30 bps. Figures 2.10 & 2.11 show the performance results for two users performing two calibration session with two different bit presentation rates. The overall single channel performance results for the all the participants are presented on the scalp locations in figure 2.5. These results have been generated using templates of order 70 to eliminate the effect of the template order on the results. The relevance of the template order and the achieved accuracy is discussed in more detail in Section 2.4.3. Based on the results from all the participants and both bit presentation rates, channels Oz, O1 and O2 stand out in performance, while Oz emerges as the best individual channel, being positioned right on top of the occipital lobe where the visual cortex is located. Furthermore, the same analysis has been done on the experiment data explained in Section 2.2.5.2. The same findings about single channel performance holds for different combination of bit presentation rate, control bit sequence length and stimuli color pair.

Next would using the information from multiple channels to make decisions. One of the main factors in this step is method that the information from different channels are fused. Simple methods, such as simple multichannel template matching in Section 3.1.2 cannot produce better performing classifiers since different channels have different information content. A naïve Bayesian fusion based method described in Section 3.2.3 is used to evaluate the effect of using multiple channels here. Figure 3.1 presents the results for four participants and under two bit presentation rates of
15 and 30 bps. The results presented here are the classification performance of the best set among all sets of \( m \) channels. At least for the used classifier, adding more channels not even does not increase the performance but also leads to a decrease in performance as more and more channels are included. One of the factors playing a role in this decrease comes from the classifier being naïve and considering a independence between the channels. One method to consider inter channel dependencies is to use the classifier described in Section 3.3 in a multi channel setting. This classifier is capable of considering the channels independent or dependent by considering various covariance matrices. For large number of channels, the dependent features become a lot, requiring many samples for estimation. Gathering many samples translates to longer Calibration sessions. While there might be room to increase the Calibration duration slightly, eventually, it becomes infeasible. A workaround, is to introduce some structure on the covariance matrix to decrease the number of variables to estimate and accordingly decrease the number of required training samples. An example would be regularization and shrinkage methods such as Regularized Discriminant Analysis (RDA)\[41\]. In addition methods such as Graphical lasso \[40\], in which a connectivity graph can be defined between different dimensions of the covariance matrix has also shown to be able to make improvements in the classification performance using more then one channel.

Although, more advance and electrode location aware multi-channel classification might boost the performance even more, but, considering the superb performance of a single channel, Oz, and its ease of use, it can be confidently used to achieve high classification performances.

\section*{2.4 Calibration}

Calibration is the mode in which the system gathers the necessary EEG evidence from the user. During this mode, the user is presented with targeted stimulation where in each epoch, one stimulus is selected as the target. The user has to focus on the target stimulus for the duration of the epoch.

\subsection*{2.4.1 Effective Strategies}

A battery of Calibration strategies should be performed to find the optimum strategy for each potential user. Considering the physiological differences from one individual to the other, it comes as no surprise that the optimum calibration strategy is also individual dependent. Here, we would like to summarize calibration strategies we found to be more effective based on the Calibration performance of many individuals who used FlashLife\textsuperscript{TM}. These strategies are provided as possible starting
Figure 2.10: Probability of correct decision for individual channels for each m-sequence using 15 Hz bit presentation rate – (a) male participant, (b) female participant.
Figure 2.11: Probability of correct decision for individual channels for each m-sequence using 30 Hz bit presentation rate – (a) male participant, (b) female participant.
CHAPTER 2. REAL-TIME SYSTEM DESIGN

Figure 2.12: Intending strategies known to produce stronger VEP responses using FlashLife™, (a) Blacks, (b) Joints. Every color is representing one intending strategy.

points and they might not necessarily provide the optimum strategy. Figure 2.12 illustrates different intending strategies, known to produce stronger responses using FlashLife™. Some individuals generated stronger responses by intending on the blocks in the checkerboards while others generated stronger responses intending to the joints. In each method, intending to blocks or joints, every color represents one strategy.

2.4.2 Design

Calibration is designed to gather the visual response to the stimuli. In order to reduce the effect of artifacts such as eye blinks, the Calibration is split to several epochs each lasting a few seconds. Each epoch consist of several trials. Every epoch has a single target stimulus. Each epoch start with presenting a fixation symbol, a plus sign, for one second in the center area assigned to the target stimulus. A yellow frame remains visible around the target stimulus for the duration of each epoch. This frame reminds the participant of the target of each epoch. In every epoch, the first trial is considered contaminated by the visual cue. Hence, in addition to the trials used for feature extraction, each epoch has an additional trial in the beginning to give the user enough time to focus on the target stimulus and also populate the visual cue. The target stimulus of each epoch is chosen randomly from a uniform distribution. While the order of targets are random, they all appear equal number of times during a Calibration session. The total number of epochs per stimulus and trials per epoch are determined based on the total number of required trials per stimulus. A typical setting to achieve 50 good quality trials per stimulus assigns 5 epochs containing 12 trials each to each stimulus. From the user point of view, since the user only follows the instructions provided by the system, we can
define the Calibration as a passive mode as opposed to the other modes of operation where the user is actively choosing the target stimulus.

2.4.3 Duration

Epoch duration is determined based on the length of the control bit sequences, the bit pretension rate and the number of repetitions needed per stimulus. Table 2.4 summarizes the trial duration under different bit sequence lengths and bit presentation rates. Gathering more data would be beneficial in making better estimates. However, shorter Calibration is always preferred. To find an estimate on the number of trials per stimulus that would lead to a good Calibration and classification performance by our classification method described in Section 3.1.1, a study was performed using different number of trials per stimulus. Calibration accuracy is defined as the percentage of correct classifications estimated using a leave one out approach on every trial of stimuli in the Calibration data. Two healthy individuals, a 22 year old male and a 27 year old female, with normal or corrected to normal vision consented and participated in two Calibration sessions. Data collection was performed following an approved Northeastern University IRB.

A four option code-VEP stimuli, controlled by four different m-sequences of length 31 bit was used. Each Calibration contained 50 epochs per stimulus. Each epoch consisted of 12 trials, resulting in 600 trials for each m-sequence. To study the effect of the template order, templates of order 10 to 120 were built for the best channel, Oz, and the performance was evaluated with cross validation. Template order is defined as the number of trials per stimulus used to build a template response to that stimuli. As expected, an incremental trend is seen in the classification performance as the template order is increased. The results vary from one individual to the other, but, a general trend of performance improvement with the increase in the template order is common. The trend has slight decreases occasionally due to noisy outlier trials included in the templates. Figures 2.13 & 2.14 show the trend for the two users. Data collected from the female user seems to have a low quality probably due to bad connection between the EEG electrodes and the scalp. The performance of each sequence has been shown separately to emphasize the effect of template length on the individual templates. The results show over 90 percent performance with templates of order 30 in the best case. Overall, templates of order 50, a value close to the stabilization point, have experimentally shown a very good performance with different individuals.
Figure 2.13: Probability of correct decision on test set using different number of training samples to determine templates for the second classifier for each m-sequence using 15 Hz bit presentation rate – (a) male participant, (b) female participant.
Figure 2.14: Probability of correct decision on test set using different number of training samples to determine templates for the second classifier for each m-sequence using 30 Hz bit presentation rate – (a) male participant, (b) female participant.
CHAPTER 2. REAL-TIME SYSTEM DESIGN

2.5 Equipment Setup

In all the studies, gUSBAnp from g.Tec has been used with the following setting unless said otherwise. The sampling rate has been set to 256 Hz. A hardware notch filter at 60 Hz has been used to decrease the AC power line noise, and a hardware bandpass filter from 0.01 Hz to 100 Hz has been used to eliminate the DC drift. This amplifier is able to acquire data from up to 16 analog channels along with an eight bit digital channel. Digital channel has been utilized to mark the onset of the events in the stimuli. Trigger signals indicating the onset of events on the display are transferred to the amplifier using the computer parallel port. Active butterfly electrodes along with a DC g.GammaBox have been used. All the data collections have been done following an approved Northeastern University IRB.

2.6 Operation Modes

Depending on the number of target options and number of stimuli, FlashLife™ can operate in two modes, immediate or cursor based selection.

2.6.1 Immediate Selection

When the number of target options is less than or equal to the number of stimuli, there can be a one to one mapping between the target options and the stimuli. So, in this selection mode, every stimulus represents its own target option. Hence, every target option can be selected in one decision making epoch.

2.6.2 Cursor Based Selection

When the number of target options is greater than the number of stimuli, several methods can be utilized to make a selection, such as grouping and grid based selection. Grouping method splits the target options in to several subgroups where the number of subgroups matches the number of stimuli. At every step, subgroups get associated with their corresponding stimulus. When a subgroup is selected, members of the selected subgroup will split in to smaller subgroups and a new selection procedure starts. The same procedure will continue until a one to one mapping between the remaining target options and stimuli becomes possible. The final selection will then take place. One of the drawbacks of such a method for the user is keeping track of their targeted option in the subgroups.
CHAPTER 2. REAL-TIME SYSTEM DESIGN

A second method, grid based selection, presents all the target options on a grid and a selection can be made using a cursor. The size of the grid is determined based on the number of target options. The grid of target options is displayed on the center of the screen in the area reserved for the target applications, the blue area is Figure 2.1.

In FlashLife™, the cursor based method is used to make a selection when the number of target options is greater than the number of stimuli. Between the grouping and the cursor based selection, cursor based selection eliminates the need of searching for the target option among several groups and makes it easier for the user to keep track of the target option. Cursor based selection on a grid also provides an opportunity to make a selection using only a single stimulus. Here, multi-stimuli and single stimulus selection methods are described.

2.6.2.1 Multi-stimuli

The four stimuli, control the movement of a cursor and the selection. The cursor moves in two dimensions, Vertical and Horizontal. At every point in time only one dimension is active. The selection mechanism is a two step process, selecting the row and selecting the column. For every selection, the system starts with a vertical sweep, where the active row is presented with a yellow frame. The active column is presented with a purple frame inside the yellow frame. The cursor starts from a default location, usually the first row from the top. The user should proceed with selecting the row in which the target option exists. After selecting the row, the next step is moving towards the target option. Once the cursor reaches the target option, the user has to make a selection. The four stimuli located at the four corners, are associated with, Select, Horizontal, Vertical and Reverse capabilities starting from the top left and moving clockwise.

To move the cursor and make a selection using four stimulus, at each point in time, an active dimension and an active direction are considered. Active dimension can be Vertical or Horizontal and active direction can be Down or Up when the active dimension is Vertical and Right or Left when the active dimension is Horizontal. Stimuli roles depend on the dimension in which the cursor is active.

- **Select**, selects the active row when the vertical dimension is active and selects the active row and column when the horizontal dimension is active.

- **Horizontal**, moves the cursor one column towards the active horizontal direction. Also will select the active row and switch the dimension to horizontal if selected during the vertical sweep.
CHAPTER 2. REAL-TIME SYSTEM DESIGN

<table>
<thead>
<tr>
<th>Active Direction</th>
<th>Active Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Vertical</td>
</tr>
<tr>
<td>(d_V = D, d_H = R)</td>
<td>Switch, Right, Up, Down</td>
</tr>
<tr>
<td>(d_V = U, d_H = R)</td>
<td>Switch, Right, Down, Up</td>
</tr>
<tr>
<td>(d_V = D, d_H = L)</td>
<td>Switch, Left, Up, Down</td>
</tr>
<tr>
<td>(d_V = U, d_H = L)</td>
<td>Switch, Left, Down, Up</td>
</tr>
</tbody>
</table>

Table 2.7: Available stimuli roles under different combinations of the active dimension and the active direction. D, U, R and L represent Down, Up, Right and Left, respectively. \(d_V\) and \(d_H\) represent the active direction in the Vertical and Horizontal directions, respectively.

- **Vertical**, moves the cursor one row towards the active vertical direction. If selected while the active dimension is horizontal, will also switch the active dimension to vertical.

- **Reverse**, will change the direction in the active dimension and move the cursor by one. For example, during the vertical sweep, the cursor moves from the top to the bottom by default. Selecting the Reverse, will cause the cursor to start moving from the bottom to the top and move by one row. Similarly, during the horizontal sweep, selecting reverse will change the direction of the sweep and move the cursor by one column. After selecting the reverse once, the user should continue moving the cursor using the active dimension stimulus, Vertical for the vertical movements and Horizontal for horizontal movements.

Table 2.7 shows the stimuli roles under different combinations of the active dimension and the active direction. The cursor moves circularly once at the borders, making it easier to reach the targets that are closer to the opposing border. It might take some time for a potential novice user to build the habit of using the stimuli and their switching roles, however, once the habit is built it becomes very easy.

### 2.6.2.2 Single stimulus (Auto Scroll)

In this mode, only one stimulus is required, Select. The system starts from the default position on the top row and sweeps vertically downwards. When the cursor comes to the desired vertical location, the user has to pay attention to the Select stimulus. After a successful selection, the system starts sweeping horizontally in the selected row from the left to the right. At this time, the user has to pay attention to the Select stimulus once the cursor overlaps with the intended target. Once the system detects the active attention, it will select the target and the procedure starts over. In this mode,
the system will use a predefined number of trials per step for stimulation usually around two. The choice for the number of trials is user dependent and it can be adjusted. This mode of operation is specifically useful where there is minimal to no gaze control.

2.7 System Parameters & Default Values

In FlashLife™, bit presentation rate is set to a default value of 110 Hz, in case this rate is not supported by the attached display, 60 Hz will be used as the second best choice. The default control bit sequence of 63 bits is used, which in combination with the default bit presentation rate results in trial length of 0.57 seconds. By default, four stimuli, placed at the four corners of the display are presented to the user, which can be mapped to different commands based on the target application. If desired more stimuli might be added to the system. Each stimulus is presented by a checkerboard at one of the corners of a rectangular LCD display placed horizontally. Each checkerboard is automatically scaled to fill up an area equal to one ninth of the width of the display squared and it consists of 5 blocks on each side. The center of the screen is reserved for the target application, e.g. the emulated environment or live feedback from the camera mounted on a remotely operable robot. Communications between FlashLife™ and target application or devices is done using a custom designed communication package based on TCP/IP or UDP protocols. Depending on the target application, four stimuli can be used to perform immediate actions or used to scroll through a menu or on a grid to make a selection among several options.

2.8 Offline Analysis & Simulation Tool

The analysis starts with a simple performance estimation, based on a Calibration data collected from an individual. This performance estimation is based on the trials in the Calibration data. The Calibration performance is estimated by applying a leave one out method on the trials in the Calibration data, splitting the trials to two folds, train and test. At each iteration, the twofold classifier described in Section 3.3 is trained using the train set and evaluated on the test samples.

2.8.1 Simulation Tool

To estimate users’ performance for a specific task, one can have users perform the tasks. To make the average performance estimates more accurate, users need to perform the task several times. Although, this method would result in the most accurate performance estimates, but, it is very time
CHAPTER 2. REAL-TIME SYSTEM DESIGN

consuming. In addition, even an actual user might perform differently from one session to the other. So it is not realistic to expect completely accurate estimates. Having an estimate of users’ performance for a specific task is helpful in adjusting or fine tuning the effective system parameters for example the cap on the number of trials to be used while making each decision.

A Simulation tool, is basically a tool to estimate users’ performance performing a task with predefined correct moves/answers. It is not possible to completely simulate a user response, however, making a few assumptions would provide enough simplification to make estimation possible. The first assumption is that users would show the same behaviour as they have shown during their Calibration session. As an example, they would look at the stimuli following the same strategy and at the same points. Second, user’s would not make any mistakes by choosing the wrong option/stimulus. In other words, users would always focus on the stimulus corresponding to their intended target. This assumption can be relaxed if user behaviour in choosing the correct targets could be modeled, for example if a user gets confused between two of the options and there exists consistent statistics able to explain the behaviour. The probabilistic model estimated for physiological evidence (Section 3.3), e.g. EEG, provides the capability of getting new samples from each stimulus class.

Estimation of the user performance is done through a series of Monte Carlo simulations. Performance estimation for a predefined task is done based on an actual Calibration data collected from the user. To consider randomness, while simulating the user response, samples are drawn from the conditional probability distribution for the target stimulus. Using this method, classification is done based on the estimated cross correlation scores skipping the template matching step. The alternative method is to choose randomly from the actual EEG trails of the target class and pass the selected EEG samples through the complete classification process. However, since this would feed the exact EEG signals used to build the templates, it would lead to over estimation of the performance.

Another use of the Simulation tool is to estimate how complete the considered models are. In other words, the closer the estimated performance gets to the actual user performance, it indicates that models considered for physiological responses e.g. EEG, and user behaviour are more accurate. Results of an experiment comparing actual and estimated performances using the Simulation tool are provided in Section 5.3.

2.9 Alternative Input Modalities

Code-VEP signals are considered as the main input modality of FlashLife™. However, the system being visual, alternative visual input modalities such as eye tracking are also viable choices.
CHAPTER 2. REAL-TIME SYSTEM DESIGN

There has been a debate, that if the individual is capable of controlling the eyes, would using an eye tracker be cheaper and easier compared to an EEG based brain interface. It is clear that for individuals without gaze control eye trackers are of no use. It has been shown that it is possible to shift the visual attention without shifting the gaze [7, 42–44]. This phenomena is called covert attention. SSVEP responses under covert condition have been studied and the separability has been detected [45–47]. Although, in such conditions a performance decrease is expected specially for novice users. Furthermore, separation of overlapping SSVEP stimuli is also shown to be possible [48].

A study by Golla on the individuals with cerebellar disorders [44], unveiled that the contributions of the cerebellum to attention are confined to overt manifestations based on goal-directed eye movements. While, the individual amount of saccadic dysmetria did not correlate with the individual performance in the covert attentional paradigm. These findings are supporting that using brain interfaces based on shifts in visual attention would be feasible [49, 50].

On the other hand, eye trackers are also not very responsive for individuals with corrected vision, especially individuals wearing eye glasses. Working based on image analysis and dealing with rapid eye movements, eye trackers usually require expensive video capture equipment and they have limited visual angle ±45° [51, 52].

2.9.1 Eye Tracking

An Arrington Research USB220 binocular head fixed eye tracker with ViewPoint EyeTracker software has been used [53]. The sampling rate of the eye tracker is 220 Hz on average. Eye tracking is done on both eyes using two independent cameras pointing at the user’s eye from approximately 7 cm. Working based on image processing principles, the sampling rate is affected by a few factors such as the processing speed and the processing load on the system, hence, it is not constant.

Using eye tracking as input modality is a two step process. The first step is to find the gaze point accurately, and the second step is to determine if the user has an intent over that point. Finding the gaze point can be done by several methods [54], some of which are proprietary to the producer company. In this study, the ViewPoint EyeTracker software from Arrington Research with a USB220 Binocular eye tracking system provides the gaze point estimates.

2.9.1.1 Calibration

Calibrating the eye tracker starts with adjusting participant’s head in the head fixed setup using the chin holder, the nose bracket and the side holders. Next, is setting up the cameras and their
corresponding infrared LEDs to point at the participant’s eyes. The process continues with adjusting the camera lens opening and focus for each eye. There are also some settings to be adjusted in the software, such as the distance between the participant’s eyes to the screen and the eye frame placed around the eye image coming from the cameras. Once all the hardware and software settings are adjusted, it is the participant’s turn to follow a set of calibration points on the screen. The number of calibration points is adjustable. For this study, we have used 9 calibration points distributed in a $3 \times 3$ matrix covering all the corners and the center points on the screen. While more calibration points usually provide more accurate calibration, it also requires more calibration time. Nine calibration points usually provide a good balance between accuracy and calibration time. After the calibration, a collection of estimated gaze points is provided to the operator, which can describe how well the points are estimated. For example, after performing the calibration on a rectangular screen, the estimated gaze points should also present a rectangle with the same side ratios. If estimated calibration gaze points are not representing the screen shape accurately, the calibration process should be repeated for all the points or for selected points for whom the calibration is off. Next, the participant is asked by the operator to look at the four corners of the screen to make sure the estimated gaze points presented by ViewPoint software are matching closely to the location that the participant is looking at.

### 2.9.1.2 Classification

Several methods can be used for intent detection such as blinking and fixating. Targeting a group of disabled individuals, intentional blinking would not be a feasible option, leaving the fixation as a better candidate. In this study, two simultaneous methods have been used, fixation detection by ViewPoint software, and the effective position of the gaze during the decision period. A natural way of using an eye tracker is very similar to using a mouse. Gaze point will be used as the location of the mouse indicator on the screen and the intent detection will act similar to making a left click with the mouse.

The same four square areas at the four corners of the screen are used as eye tracking stimuli. All the user has to do, is to keep their gaze inside the area assigned to the target stimulus.

Two simultaneous methods has been used to identify the user intent at all times, detecting a fixation and the effective position of the gaze during the decision period. During the designated time window, when ever one of the classifiers detects an acceptable intent a decision will be made.
CHAPTER 2.  REAL-TIME SYSTEM DESIGN

Fixation Based  Fixation is defined as maintaining the gaze on a specific location. A common method to detect fixation, is to monitor the gaze point using a sliding time window. If the gaze point is kept in the predefined area without fast movements, then, a fixation point can be estimated. However, parameters such as the area of the specific location, variations of the gaze point and the duration of time that the gaze has to be kept constant have a high impact on the fixation estimation quality. ViewPoint calculates fixation as the length of time that the velocity was below the saccade velocity criterion. The fixation events are detected using the ViewPoint software with a dwell time of one seconds to match the system parameters and also decrease the number of false positives.

Gaze Based  Position of the gaze changes both rapidly and frequently, specially when an individual is searching over an area in the visual field. On the other hand, fixating over a point is also very tiring and sometimes not very feasible when eye movements are not all intentional. Here we propose an intent detection method solely based on the gaze points during the decision time window. In this method, wide areas are assigned to the stimuli and the gaze movement speeds are not considered, relaxing the fixating parameters to make things easier for the user. With the decision rate of 1 Hz, the location of the gaze is monitored over the 1 second period of time. To estimate a single point as the intended point for each decision period, there is a need to combine the information gathered from all the gaze point estimates during that period. Here, the number of samples is about 220 samples per decision. Considering the fast and sometimes unintentional movements of the eyes, a method, robust to the outliers, would make a better estimate. Hence, we use median as a dimensionality reduction tool which has a better tolerance to outliers. Every gaze point is defined by two values representing the horizontal and the vertical location. We estimate the intended point during each epoch by taking the median over the horizontal and vertical dimensions independently.

When a single point is estimated as the gaze point, it will be matched against the areas assigned to the stimuli. In case of an overlap between one of the stimuli and the estimated gaze point, that stimulus will be selected as the user intent, otherwise, a new trial will begin.
Chapter 3

EEG Signal Modeling, Context Fusion & Decision Making

This chapter focuses on EEG processing, dimensionality reduction, feature extraction and classification. In addition to EEG processing, context information fusion, and decision making rules are described for a general application. Feature extraction heavily depends on the type of stimulation. When frequency based stimulation is used, most of the time, feature extraction is based on the power spectral density (PSD) of the signals around the target stimulation frequencies and their harmonics [55]. Alternative methods such as canonical correlation analysis (CCA) [56] also has been widely used. However, using a code based stimulation mainly, we proceed by the time domain feature extraction method used in FlashLife™.

3.1 Template Matching

Template matching refers to a method in which a template response is estimated given a specific input. Template matching is a capable method applicable to time invariant systems. The similarity between the estimated template response and the measured response is the main distinguishing factor. Since the templates can’t be updated rapidly, this method is more suitable for time invariant systems. However, depending on the rate of the changes, more complex methods can be incorporated to consider slight time varying changes. Template responses are estimated using labeled data, collected during a Calibration session. Given the low signal to noise ratio of EEG measurements, template responses are estimated using averaging methods in order to increase the signal to noise ratio and estimate better templates. The simplest method is taking the sample mean, however, being prone
to artifacts, sample median performers better being more robust to the outliers [57]. Defining the 
Breakdown Point (BP), as the highest fraction of outliers that the estimator can handle without 
breaking down, [57] we compare the sensitivity of these estimation methods to outliers. Considering 
\(N_t\) as the number of trials per stimulus, the finite sample BP for the sample mean is \(\frac{1}{N_t}\) resulting 
in an asymptotic breakdown point of zero. However, for the sample median, the finite sample 
BP is \(\frac{N_t - 1}{2N_t}\) resulting in an asymptotic breakdown point of 0.5. In other words, even where up to 
half of the data points are outliers, the sample median will stay close to the actual templates. The 
quality of the templates also depend on the type of signal used as the system input. Inputs known 
to have less similarities would generate more distinguishable responses. One of the best choices 
are the m-sequences or the similar gold sequence family. Some of the properties of these sequences 
are discussed in the previous chapters such as being almost balance and a high auto-correlation 
only at lag zero. The longer sequences of these kinds get, their auto-correlation would become a 
closer estimation of the delta function. This property specifically can be used to extract the system 
impulse response. Although, human visual system gets its input from a large visual field, still 
using these sequences as the control bit sequence for the stimuli helps to build a stable and robust 
template response. The same property makes these sequences wideband. This property, even for 
short sequences, is illustrated in Figure 1.3 frequency response of VEP responses to the code-based 
stimulation with m-sequence based control bit sequences. Using enough repetitions in template 
estimation and the fact that processing the correlation scores between the templates and the measured 
responses preserves the frequency content, template matching has been shown to provide accurate 
and robust results [37]. As a default setting, 50 repetitions of each sequence collected during a 
Calibration session are used to build the templates. More details about the Calibration and parameter 
selections are discussed in sections 2.2 & 2.4. Template matching classification can be applied in 
two ways, single channel or multichannel.

### 3.1.1 Single Channel Template-Matching

This classifier simply correlates the template built for each sequence with the EEG signal. The 
m-sequence template that yields the highest correlation coefficient is selected. Specifically, the 
decision of channel \(c\) is \(d_c = \arg \max_i \rho_c^i\), where \(\rho_c^i\) is the correlation coefficient between the \(t_c^i\) 
template for the \(i\)th m-sequence for channel \(c\) and \(s^c\) the windowed EEG signal from that channel 
given by

\[
\rho_c^i = s^{c^T} t_c^i
\]  

(3.1)
CHAPTER 3. EEG SIGNAL MODELING, CONTEXT FUSION & DECISION MAKING

The performance of each channel is estimated on the training data with k-fold cross-validation and the best performing channel is selected as the only one that makes the final decisions.

3.1.2 Multi-channel Template Matching

Multi-channel classification can be done in several ways. First, we start with an extension of the single channel classifier. It performs the same operation as above, but concatenates the signals from all channels into a large template and signal vector. Specifically, the decision is given by 

\[ d = \arg \max_i \rho_i \] 

where \( \rho_i = (s^T t_i) \). Here, \( s = [s_1, s_2, \ldots, s_C]^T \) and \( t_i = [t_{i1}^1, t_{i2}^2, \ldots, t_{iC}^C]^T \), where \( C \) is the total number of channels.

A series of single channel template matching classifiers also could be used to make single channel decisions and use a majority voting classifier to come to the overall decision. However, the above mentioned method makes use of the correlation difference information while, majority voting simply neglects the difference.

While a simple template matching classifier shows high performance, there are some goals that cannot be achieved using only this approach. To add the capabilities such as confidence threshold, variable number of sequences, simulating the user performance and incorporating the context information, we moved towards a probabilistic approach. A probabilistic approach also takes into account the variations of the correlation scores. The probabilistic approach is implemented by adding another layer on top of the scores from the template matching. If desired, any other dimension reduction method can replace the template matching. We used density estimation methods to fit a probability density to features extracted from the calibration data set for each class. Different methods have been evaluated which will be described in the following sections.

3.2 Kernel Density Estimation

Without loss of generality, we chose Gaussian kernel due to simplicity and reasonable results. Here we start with a general form of the Kernel Density Estimation (KDE). Then the variations in the bandwidth selection method and the feature vector dimensions will be discussed. A general Gaussian kernel can be represented as,

\[ p(\rho_{i,j} | i) = \frac{1}{N_i} \sum_{s=1}^{N_i} C \sigma_i^2 (\rho_{i,s}^C - \rho_{i,s}^C) \] 

(3.2)
where, $N^i$ is the template order, $\sigma^2_i$ is the bandwidth parameter, $\rho^c_i,s$ is the correlation score between the template for $l^{th}$ m-sequence and windowed EEG signal $s$ from channel $c$ and $\rho^c_i,i$ is the correlation score with template for $l^{th}$ m-sequence and the $i^{th}$ sequence in the template.

### 3.2.1 One dimensional KDE

The features used to build the kernels are the template matching scores between the EEG sample and the template of the target class. In this case scores $\rho$ are scalars. So for each class we extract $N_i$ scores and build a kernel density estimate. In this model, the bandwidth parameter $\sigma^2_i$ is also a scalar and calculated using Silverman’s rule of thumb \[58\]

$$\sigma^2_i = \frac{1}{n} \text{tr}(\text{Cov}(\rho^c_i,s)) \left( \frac{4}{(2n + 1)N^i} \right)^{\frac{2}{n+4}}$$  \hspace{1cm} (3.3)

where $n$ is the dimension of the data, set to unity in this case. The covariance also reduces to standard deviation due to the unit dimension of correlation score data. The kernels in this case only consider the variations of the correlation score between the test data and the target class template.

### 3.2.2 Multidimensional KDE

While the variation of the correlation scores between the target class and the windowed EEG makes the main difference between the classes, considering the variation of scores between the windowed EEG and the non-target classes creates a better model for the data. Therefore, the kernels are changed to multi dimensional kernels to incorporate the score changes across the target and non-target classes. In this case, the correlation score, $\rho$, is a vector containing the correlation scores between the test sample and all the templates. So the dimension of the densities is the same as the number of classes.

We used different bandwidth estimation methods such as Silverman rule of thumb in the multidimensional case \[58\] and K nearest neighbor (KNN) method. The Silverman rule of thumb shows more sensitivity in the kernels during the online testing. On the other hand, KNN method offers some adaptation in the amount of smoothing to the local densities. So KNN performs more robust and the estimated performance using the simulation tool explained in section 2.8 are closer to the reality.

### 3.2.3 Naïve Bayesian Fusion

We further advanced the multichannel classifier to a naïve Bayesian fusion classifier. The motivation to use a Bayesian fusion classifier is to complement the best channel by leveraging
useful information from other EEG channels, in order to increase the accuracy of the BCI classifier. Independence of the channels is the key assumption behind this method, hence the descriptor naïve. The naïve Bayesian fusion classifier uses the same scores from the template matching classifier described above; this allows for a simple linear dimension reduction in the overall feature vector. For the calibration data correlation scores for each channel and m-sequence pair, a Gaussian Kernel Density Estimate (GKDE) is obtained. The bandwidth parameter for the Gaussian kernel is calculated using the Silverman rule of thumb specified below [58]. During the test session, after receiving all scores for channel and m-sequence pairs for the new EEG trace under consideration, using the estimated GKDEs, a new probabilistic score for each correlation score is obtained. Using the channel-score conditional independence assumption (given the m-sequence) and taking the logarithm of the likelihood to obtain log-likelihood, the overall decision is obtained based on conditional a posteriori likelihood calculations; these are the summation of the logarithm of the individual channel/m-sequence probabilities. The sequence which has the highest a posteriori likelihood (assuming uniform priors for m-sequences) will be the winner. The decision criterion is

$$d_c = \arg\max_l \log p(l|\rho^1_c, ..., \rho^C_c)$$  \hspace{1cm} (3.4)$$

where $C$ is the number of channels, $\rho^c_l$ is the correlation score for channel $c$ and template for sequence $l$ defined as given above in the template matching.

The GKDE for the probabilistic distribution of these correlation scores, obtained using calibration set data is given by

$$p(\rho^c_l|i) = \frac{1}{N^i} \sum_{s=1}^{N^i} G_{\sigma^2_i}(\rho^c_l - \rho^c_{i,s})$$  \hspace{1cm} (3.5)$$

where $\rho^c_{i,s} = y^c_s T^c_i$. Under the assumption of conditional independence of the channels given the sequence, the decision $d_c$ will be simplified as

$$d_c = \arg\max_l \log p(y^c_l|i)$$  \hspace{1cm} (3.6)$$

$$= \arg\max_l \log \left( \frac{1}{N^i} \sum_{s=1}^{N^i} G_{\sigma^2_i}(\rho^c_l - \rho^c_{i,s}) \right)$$

where $\rho^c_l$ is the new correlation score from the test data and $\rho^c_{i,s}$ is the correlation score from the training data sample $s$ for channel $c$ and m-sequence $i$. In this GKDE model, the bandwidth parameter $\sigma^2_i$ is calculated using eq. 3.3, where $n$, the dimension of the data, is set to unity.
This classifier could use one or multiple channels up to the total EEG channels. By adding more channels which contain some information to the Bayesian fusion classifier, the overall results would improve as long as the assumption of having individual conditionally independent channels holds. In the cases where the channels have some correlation, adding more channels with the independence assumption may decrease overall classification accuracy, especially if these correlated channels are poor performers themselves.

Here we present the results of a study conducted on four healthy users with normal or corrected vision with ages 22 to 28 years. To investigate the effectiveness of the fusion we evaluated the performance starting with only one channel (no fusion) and scaled up to the all the 16 channels. We should mention that the presented results are for the best \( m \) channels. In other words, we evaluated the performance of all the \( m \) channel pairs and presented the results from the best set of size \( m \). Figure 3.1a shows the overall classification accuracy across four \( m \)-sequences using the naïve Bayesian fusion of best-\( m \) channels (best-\( m \) for each \( m \) taking values 1 to 16 are obtained using brute force combinatorial search to provide the best possible results). Figure 3.1b shows the same performance results for the 30 Hz flickering frequency. Although, the assumption of conditionally independence between the channels can hold due to the fact that different locations on the scalp are presenting the reactions to different characteristics of the stimulus, i.e. the response to the vertical edges as opposed to the horizontal edges, but due to the superposition effect of the skull on the electrical activity, they can have a good amount of correlation. These results clearly demonstrate that the naïve Bayesian fusion approach is not effectively combining information from different channels; this can be attributed to the likely fact that EEG signals; therefore correlation scores extracted from them via template projections are correlated with each other especially between neighboring and nearby sites. As a result the accuracy of this classifier begins with the same accuracy of the single channel template matching classifier and goes down as the number of channels included in the fusion increases.

### 3.3 Multivariate Gaussian Distribution

A multivariate Gaussian density with \( d \) dimensions can be written as

\[
p(\rho) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}e^{x}p\left[ -\frac{1}{2}(\rho - \mu)^t \Sigma^{-1}(\rho - \mu) \right]
\]

(3.7)

where \( \rho \) is a \( d \)-component vector containing template matching scores, \( \mu \) is a \( d \)-component mean vector and \( \Sigma \) is the \( d \times d \) covariance matrix with \( |\Sigma| \) and \( \Sigma^{-1} \) being its determinant and inverse [59]. It can be easily shown that these values for mean and covariance are the maximum likelihood
CHAPTER 3. EEG SIGNAL MODELING, CONTEXT FUSION & DECISION MAKING

Figure 3.1: Classification accuracy of naïve Bayesian fusion of best-m channels for m from 1 to 16 estimates. Although, the covariance estimate is biased, it is still close enough when the number of samples is not too small.

Here, we used the template matching scores for each template as the EEG features extracted. Using the labeled calibration data, each trial in the calibration set will have one score from each template. Although, we can build a density only using the score from the target template, but adding the scores from the non-target templates will consider their variations. A multivariate density will perform better, having extra information from the competing templates. During the testing mode, for each new trial, we get a likelihood score considering each template as the target. The likelihood scores get fused with the context information based on the fusion rule explained in section 3.5.1 resulting in the decision with a confidence value. In our system we generate 50 samples for each class, and the usual number of dimensions is four. We examined the shrinkage and the regularization effect on the densities, but, since there wasn’t any significant change in the performance they haven’t been used later on.

After careful examination of user performances during real-time navigation and control task and estimating their performance using their calibration data set, we came to the conclusion that Kernel Density Estimation methods create very sharp densities such that small amount of noise might make a big difference in the likelihood values. It appears that a multivariate density created using the sample mean and covariance creates a better fit. Different factors are used to compare the overall performance of different methods. The criteria are the combination of the error rate while evaluating the calibration data and the real-time performance of the user compared with the estimated performance by simulations. The real-time performance is measured in the sense of user ability to complete the task and the amount of mistakes made.
This method can easily extend to multi-channel scenario. Unlike the naïve Bayesian fusion method, inter channel dependencies can be modeled and considered in the estimation of the covariance matrix for each class. The drawback of adding more channels is of course the need to have more Calibration data to be able to make good estimates. A good rule of thumb is having the number of Calibration trials to be at least about 10 times the number of features. However, the number of features can grow rapidly as the number of channels increase. Hence, soon there comes a point where the length of the Calibration becomes infeasible and the sample covariance becomes a poor estimate of the actual covariance. In such situations, the number of required calibration samples can be reduced by introducing some structure on the covariance matrix.

### 3.3.1 Sparse Covariance Estimation

One of the methods suitable to introduce some structure (sparsity) on the covariance for Gaussian densities is the Graphical Lasso method [40]. Considering the physiological structure of the brain, it is possible to define a connectivity matrix between the channels. In other words, the connectivity matrix can put some constraint on the channels that should be considered independent and the ones that should be considered dependent. Typically, EEG channels located in a close neighbourhood should be considered dependent, where the amount of dependency decreases with an increase in the pairwise distance. By means of the connectivity matrix, the channels that are far from each other are considered conditionally independent.

Using the connectivity matrix, in situations where some channels can be defined independent, can decrease the number of elements requiring estimation in the covariance matrix. As a result, the number of Calibration trials per stimulus can also be decreased. However, using multi-channels increases the setup time and decreases user comfort. Furthermore, in cases where the single channel (Oz) performance is high, 95% for example, performance estimates of sample trials of multi-channel Calibration data shows slight increase of about 2%. The same comparison on a data set where the single channel (Oz) performance is poor e.g. 68% has been able to increase the performance by 10%. Hence, multi-channel use is recommended when single channel performance is poor.

To further investigate the effect of multi-channel classification taking advantage of the connectivity matrix defined above, a study of multi-channel Calibration with 10 participants over two days with two Calibration sessions in each day is considered. Figure 3.3 shows that use of the connectivity defined by the GLASSO method the multi-channel classification has been able to increase the Calibration performance with higher increase for the poor performing participants.
CHAPTER 3. EEG SIGNAL MODELING, CONTEXT FUSION & DECISION MAKING

Figure 3.2: Channel locations and the connectivity (red lines) defined using GLASSO method.

Figure 3.3: Calibration accuracies over two days and two Calibration sessions each day, for single channel and multi-channel classification methods. (D stands for Day)
3.4 Introduction of Context Information

In this section, we introduce the use of probabilistic operation behavior models on context information. Context information, is the information extracted from the non-physiological sources. These sources depend on the target application. For example, in a typing application, such as FlashType™, the context information would be in the form probability of letters, words and sentences based on a language model. In a control application, such as FlashNav™, the context information sources start with the geographic information, floor map, surrounding obstacles and can spread to include scheduled tasks and user habits. In another control application, FlashGrab™, an application to manipulate objects, the context information can even include the sequence in which several objects might be used or have been used by the user. The context information, if utilized correctly, is priceless. It can increase the performance of the system remarkably, decreasing the cognitive load on the user and increasing the reliability and sustainability of the decisions.

As a simple example, we can consider controlling a wheelchair or rover using immediate control commands, *Turn Right*, *Turn Left*, *Go Forward* and *Stop* in an indoor environment. The most preliminary context information source in such an application, is the floor map. The context information at this level provides a mean to almost eliminate the infeasible options from the selections. Meaning that, if there are four simultaneous options all the time, but at a location, there are only two feasible options, the other two would be highly unlikely to be chosen. Therefore, the same consideration will be made by the classifier, giving lower prior probabilities to the infeasible options and making the feasible options more likely. As an example, we can consider being in the middle of a corridor, in which only feasible options would be going forward or backward. Even at this stage, the context information can get more advanced by considering physical limitations of the target device. For example, a wheelchair might not be able to perform some of its operations in a narrow corridor. This determination depends on the relative size of the wheelchair and the corridor. Furthermore, context information can also be used on the application side. In this example, using the context information on the application side would translate to feasibility assessment before executing a received intent from the classifier. The example provided here is very simplistic, more advanced and detailed, application based context information is discussed in applications in Section 4.

In FlashLife™, context information gets utilized as the a priori information about the options provided to the user and gets fused with the extracted feature scores from the biological information sources, i.e. EEG evidence, details are provided in Section 3.5.1. In the absence of context information, simply all the options will be considered equally likely. Consequently, context information can
reduce the possibility of infeasible options to extremely low values. We should note the probabilistic nature of the context information. In other words, since the context information is extracted using the sensory information like Sonar and Lidar\textsuperscript{1}, there exists some degree of uncertainty in the information. In an uncertain world, all the options should have a positive possibility. The user should be given the opportunity of selecting all the options even though they appear highly unlikely given the context information. Hence, the probability of infeasible options gets decreased, but it won’t turn to zero any time, giving the user the ability to overcome the context information only with a very strong EEG response. Although, the system provides the ability of overriding the context information, but, decisions that are found dangerous by the system will be ignored to provide the safety measures. Instances of such conditions are application dependent, in the wheelchair control application, going over the satires or running into the fireplace, even if chosen by the user, would be overruled by the application.

3.4.1 Dynamic Confidence Threshold

One of the advantages of having a probabilistic model is the ability to provide a confidence measure, explaining how likely decisions are given the context information and the biological evidence. On the other hand, a probabilistic model, requires a confidence threshold to categorize the decisions to final or in process. A decision would be final when the confidence threshold is met. If the estimated confidence doesn’t meet the threshold, new evidence should be collected. The classifier whether discards the old evidence and starts fresh or adds the new information and evaluates the newly calculated confidence. A higher confidence in decisions would usually translate to more reliability. However, a trade off exists between how fast the decisions are made and how confident the decision are. In other words, achieving a higher confidence usually requires gathering more data, in this case EEG data, which would take longer and decrease the decision rate.

There are not trivial ways to find the absolute balance. And finding the optimum threshold selection is not an easy task. A low threshold would make the decision making process faster, but at the same time it might increase the probability of making a wrong decision. On the other hand, a very high threshold will decrease the error probability, but will also decrease the rate of the decisions. To avoid the two extreme cases, the threshold should be somewhere in between.

To find a lower bound for the confidence threshold, we can utilize the context information. In cases where context information heavily supports one of the options, a high prior probability would

\textsuperscript{1}LIght Detection And Ranging
be assigned the aforementioned option. Hence, even with weak physiological evidence, the classifier might make confidence decisions towards that option. This is approximately equivalent to making decisions based on the context information. To avoid decision making heavily influenced by or solely based on the context information, the lower bound for the confidence threshold should be higher than the most probable option based on the context information. On the other hand, we don’t want to make the decision making very hard, since it might require several trial of EEG evidence. Hence, we define a confidence gap parameter. Confidence gap is the difference between the maximum a priori context information and the confidence threshold. We can define this gap such that the confidence threshold resides in the middle of the maximum context information and one (the maximum possible value for the confidence). Although, every point in that period is a valid choice, but we think choosing the center will create a good balance. Additional constraints such as the difference between the confidence values of the most confident decision and the runner up decision can be considered. Creating a dependency between the context information and EEG evidence automatically makes the decision making more sensitive to the physiological evidence.

3.4.2 Mastery Task & Training

Using the context information concept, we have designed specific tasks with different difficulty levels, called Mastery tasks. Mastery task are basically a sample run of the target BCI application, such as typing, e.g. FlashType™ and control e.g. FlashNav™ or FlashPlay™. However, Mastery tasks have different difficulty levels and the correct decisions are known beforehand. Having several difficulty levels plays an important role in training the potential users. Lower level Mastery tasks help the users to make the correct decisions at every step. Amount of help provided by the system, decreases as the Mastery task levels go higher. Eventually, the difficulty of the tasks reaches a point that the system actively makes the incorrect decisions more likely. In hard Mastery tasks the user has to put more effort and focus in order to complete the tasks. The benefit becomes more clear when the user starts to use the system on a daily basis and in uncontrolled environments.

Considering the dependency of the Mastery tasks on the applications, their implementation differs from one application to the other but they all follow the same principles. Here, we describe how in general the difficulty of the task are adjusted. The special cases and specific considerations are discussed in more details in the applications in Section 4.

The common rule applied to adjust the difficulty level in Mastery tasks is as follows. The decision difficulty level is defined as the ratio of the probability of making the correct decision, $P_{Correct}$, and
the maximum probability among the incorrect choices, \( \text{MaxP}_{\text{Incorrect}} \).

\[
\text{Decision Difficulty} = \frac{P_{\text{Correct}}}{\text{MaxP}_{\text{Incorrect}}}
\]  

(3.8)

using the above definition, we define the \textit{Mastery Task} difficulty level by defining lower and higher bounds for the \textit{Decision Difficulty} values. So for each level the decision difficulty is bounded by the limits. While executing the task, at each step, a random value is drawn between the valid upper and lower limit. To find a good fit for the boundary values, one should consider a few factors such as the total number of choices, the average confidence of EEG (or biophysical only) classifier and may be the type of the biophysical response.

3.5 Online Operation of FlashLife™

To start using FlashLife™, a potential user should start with a \textit{Calibration} session described in Section 2.4. Once a well performing \textit{Calibration} is reached, user can proceed by using the system in one of the provided/supported applications. After every \textit{Calibration}, performance is estimated using a leave one out method on the trials collected during the \textit{Calibration} session. Novice users might need to perform the \textit{Calibration} a few times and try different strategies until a suitable strategy with acceptable performance is achieved. For example, in a four stimuli setting where the chance level is 25%, usually a \textit{Calibration} performance of 85% or higher is considered a good performance. The \textit{Calibration} duration depends on system parameters discussed in Section 2.4.3. With the default settings a \textit{Calibration} only takes about 3 minutes. A well performing \textit{Calibration} can be used again as long as electrode placement and more importantly user strategy do not change.

The number of stimuli is easily expandable and supported by the signal processing and classification methods discussed in Chapter 3. However, considering the fact that the stimuli are flickering, with an increase in the number of stimulus the irritation and tiring effects also increase. In addition, with an increase in number of stimulus, the inter-stimulus distances have to decrease resulting in an increase in the inter-stimuli interference. Hence, the default number of stimuli is set to four placed at the four corners of the display. This specific arrangement of the stimuli provides a good balance among the number of stimuli, inter-stimuli distance, area reserved for the applications and visualizing feedback to the user. Stimuli roles might change from one application to the other, discussed in more detail in Chapter 4.

After \textit{Calibration} is done, the twofold classifier described in Section 3.3 gets trained using the \textit{Calibration} data. When the application of choice starts, first the application is presented to the
user to give the user time to evaluate the state of the application and decide on what action to take next, and then the stimuli start. Using the classification method described in Section 3.5.1 and more specifically eq. 3.13, trials can be processed one by one as soon as they become available. This method decreases the stimulation and classification dependency. The stimulation can run without an interrupt presenting trials of stimuli one after the other. At the same time, trials when completed, get processed by the classifier in the background. Based on the confidence of the classified intent, the system decides to gather more trial or make the decision final. The number of trials used for every decision is limited cap value. If the number of trials reaches the upper bound, then depending on the system setting, a less confident decision might be made or the classifier might discard the past trials and start gathering trails for the same decision again. When a decision is made based on the general decision rule discussed in Section 3.5.2, stimulation stops, the decision gets presented to the user and the appropriate action is taken by the target application. Once the corresponding action execution is over, the latest state of the system is presented to the user and stimulation starts to gather data and identify user intent for the next action. While this pause period is not necessary for system operation, it helps the users to understand the state of the system and gives them some time to think what they want to do next.

The system is capable of making decisions after each trial, which leads to a decision time of one or even half a second when the higher presentation rate is used, the probabilistic confidence based approach provides a smooth way of increasing the confidence of the decisions, thus, increasing the reliability of the decisions. On the other hand, when decisions are made too quickly, every one or half a second, the cognitive load on the user is more and the overall usage of the system will get harder requiring precise attention all the time. This design choice is in line with the potential user preferences [5]. BCI users in the virtual forum provided a feedback that for them the robustness and stability are the first priority and rate and seed comes next.

3.5.1 Fusion rule

This section describes how the features extracted from the physiological sources such as EEG and non-physiological sources, such as context, are fused together to identify a common intent. Here the general method is described. However, different applications might have slightly different fusion or decision rules discussed in Chapter 4. Having a Bayesian framework makes it possible to make a joint inference based on several sources. Here, mainly two sources are considered, physiological, EEG, and non-physiological, context. Context information is considered independent from EEG
evidence. In addition, EEG evidence from different trials are also considered independent given the class label. Furthermore, every epoch is assumed to have only a single target.

Figure 3.4 shows the general system graphical model. Consider \( x_i \) as EEG response of a trial, \( X = [x_1, \ldots, x_{N_t}] \) will represent all the EEG evidence in an epoch. The correlation score for the templates will be \( \rho_i = x^T t_i \) so \( \rho \) will be a \( d \) dimensional vector containing the correlation scores between the windowed EEG \( x_i \) and all the templates. Next step, is to estimate the conditional probability density of the scores given the class labels. With the above assumption, this can be calculated using one of the kernel density estimation methods explained in section 3.2 or from a multivariate Gaussian described in section 3.3.

Using the class conditional probabilities and by applying the Bayes’ theorem, we can write the fusion with the context (non-physiological) information, \( \omega \), as follows

\[
P(c_s = c|X, \omega) = \frac{P(X|c_s = c)P(c_s = c)}{P(X|\omega)P(\omega)}
\]  

(3.9)

having the independence assumption between EEG evidence and the context information

\[
P(c_s = c|X, \omega) = \frac{P(X|c_s = c)P(\omega|c_s = c)P(c_s = c)}{P(X|\omega)P(\omega)}
\]  

(3.10)

which then can be simplified as

\[
P(c_s = c|X, \omega) = \frac{P(X|c_s = c)P(c_s = c|\omega)}{P(X|\omega)}
\]  

(3.11)

The conditional probability of the EEG evidence for each trial, \( P(x_i|c_s = c) \), can be calculated using eq. 3.7 or eq. 3.2.
Considering EEG evidence between trials independent, along side with the assumption that the target stays the same in each epoch, we can write the conditional probability as

\[ P(X|c_s = c) = \prod_{i=1}^{N_t} P(x_i|c_s = c) \]  (3.12)

which will simplify eq. 3.11 as

\[ P(c_s = c|X, \omega) = \frac{\left(\prod_{i=1}^{N_t} P(x_i|c_s = c)\right) P(c_s = c|\omega)}{P(X|\omega)} \]  (3.13)

where \(N_t\) is the number of trials used to calculate the posterior probability and \(c_s\) is the class of the stimulus. The recursive update helps to collect only as much data as needed. In addition, the update being recursive, posterior probabilities calculated at each trial act as the priors for the next trial and keep the number of computations the same. Using the minimum amount of data required and having small number computations are important factors for real-time system operation. In eq. 3.13 the only term that is unaccounted for is the denominator, \(P(X|\omega)\). Although it is not possible to calculate this term, but since it is common between all the classes and they are probability mass functions (PMF), without loss of generality we can normalize the PMFs without this term.

### 3.5.2 Decision Rule

The decision making mechanism is, in fact, a maximum a posteriori classifier. However, in the absence of the context information the classifier will act as a maximum likelihood classifier. The final decision will be made based on eq. 3.14 once there exists a class whose posterior probability, confidence, is higher than the confidence threshold. Confidence threshold adjustment is explained in section 3.4.1.

\[ D = \arg \max_c P(c_s = c|X, \omega) \]  (3.14)

The above fusion and decision rule, provide a probabilistic approach to use variable number of trials to make a decision. During each epoch, stimulation continues until the classifier is confident enough to make a decision.
Chapter 4

Applications

Brain Computer Interfaces have been under research and development for decades. Different brain responses are utilized such as Event Related Potentials, P300, Visually Evoked Potentials and Event Related Synchronization / Desynchronizations. Every response has its own strengths and weaknesses. Mostly, BCIs have been utilized for specific tasks such as typing or control. Hence, there is a need for customization based on the user needs and the brain response characteristics while deciding on what assistive technology might be suitable for an individual with disabilities.

Every system needs some training, and it takes time to build the habit to use it comfortably. Therefore, using a single input modality for different applications might be preferred. In addition, the results of a virtual forum discussion among the current users of assistive technology has highlighted that for these users the highest priority is having a reliable, robust, accurate and easy to setup system. High decision rate and ITR have a lower priority [5]. With an attempt to provide a brain interface capable of supporting basic needs for everyday life we have used FlashLife™ and developed several applications. FlashLife™ provides the stimulation and physiological feature extraction and intent classification incorporating the context information provided by different applications. The main applications supported by FlashLife™, are FlashType™, FlashNav™, FlashGrab™, and FlashPlay™. FlashLife™, provides fast, reliable, robust classification while using only a single EEG electrode placed on the center of the visual cortex, at location Oz based on the 10 – 20 electrode placement standard. Taking advantage of code Visually Evoked Potentials (c-VEPs) and stimuli optimization [33], decision rate of approximately 1 Hz and more than 90 percent accuracy, a reliable channel for the participants is provided to interact with their target applications. In addition, calibration only takes less than three minutes and it is not required frequently. Individuals with gaze control can use FlashLife™ with eye tracking as an alternative input modality described in
CHAPTER 4. APPLICATIONS

Section 2.9.1 A battery of Calibration sessions can be used for every user to optimize system parameters such as presentation rate and stimuli size and color and boost the performance even further. Stimuli roles play a key role in adaptability of FlashLife™, however, stimuli itself stays the same throughout.

4.1 FlashNav™

Navigation, is one of the most important abilities for every individual. As an example, navigating a wheelchair through out a house is one of the basic examples. The second example would be navigating a remotely operated robot in order to reach to certain places. In FlashNav™, two control modes are supported, immediate control and destination based navigation. Using the immediate control mode, users have the capability of controlling the target device step by step. In this mode, stimuli roles starting from the top left corner and going clockwise are defined as Turn Left, Turn Right, Stop and Go Forward. Destination based navigation, takes advantage of autonomous navigation and the user only selects a target location. Based on the number of target locations, immediate or menu based selection scenarios are used. If the number of possibly desired locations are equal or less than the number of stimuli, a one to one mapping would be used to choose the target location. In case the number of possibly desired locations is greater than the number of stimuli a menu based scenario is used. In this mode, stimuli roles are changed to Select, Horizontal, Vertical and Reverse. The user has to navigate a cursor over the menu items and make a selection.

Despite the mode, context information if available is incorporated making decisions. However, the context information in the immediate control mode is in a lower layer using information about the surrounding objects and floor map in addition to possible directions and frequency they are used. Figure 4.1 shows three snapshot of the low level information extracted from the environment using a LiDAR sensor. In destination based navigation, context information mostly consists of information about the destinations, when and how often they are chosen, objects and locations of interest and user habits, the low level information is used by the autonomous navigation system. Destination based navigation is preferred since it can decrease the cognitive load on the user taking away the need for constant attention. However, immediate control give the user the ability to make personal maneuvers to places that are not considered among the destinations. The application area in the user interface can present a real-time video feedback illustrating the robot perception or a map of environment displaying the location of the wheelchair on the map.

In a simple experiment, a participant used FlashNav™ with immediate commands to control a
CHAPTER 4. APPLICATIONS

Figure 4.1: Three snapshots of the low level context information extracted from the environment [3].

<table>
<thead>
<tr>
<th>Route</th>
<th>Duration W.O. Context (sec)</th>
<th>Duration W. Context (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>$D_3 \rightarrow D_2$</td>
<td>60</td>
<td>64</td>
</tr>
<tr>
<td>$D_2 \rightarrow D_1$</td>
<td>60</td>
<td>54</td>
</tr>
<tr>
<td>$D_1 \rightarrow D_3$</td>
<td>109</td>
<td>145</td>
</tr>
</tbody>
</table>

Table 4.1: Experiment Task Durations.

wheelchair going through the points $D_1$, $D_2$ and $D_3$ three times with the help on context information and three times without the help of context information. Relative distance of the points were $D_3 \rightarrow D_2 \rightarrow D_1 \rightarrow D_3$. Total distance traveled were 192 meters. Table 4.1 presents the time spent to travel between the target points in seconds. As the results in Table 4.1 shows, even use of basic context information can decrease the duration of the task significantly.

4.2 FlashGrab™

FlashGrab™ deals with grabbing and manipulating objects. Object manipulation has been done with Baxter, a low cost humanoid robot [60]. Objects and more specifically, handle shape parts on the objects are identified with an image processing algorithm [61]. Then detected handle parts are numbered and presented
CHAPTER 4. APPLICATIONS

to the user as overlays on a video feed illustrating the perception of Baxter to the user. Depending on the number of graspable handles, a direct or a multistep decision will be made by the user. If the number of detected handles is less than or equal to the number of stimuli, stimulus labels will be associated with the overlaid handle numbers, otherwise the user has to make a selection among the handle numbers presented in a menu. In addition, information about the detected objects if available will be considered by the classifiers while detecting the intended object. After a selection, the robotic arm picks up the object. At this point the robotic arm can move the object to a predefined location or, if desired, a menu can be presented to the user to choose a destination. Stimulation is active only when a selection needs to be made. A video feed at the center of the screen presents the robot perception and the target destination to the user. This is a very simplistic implementation of an object manipulation application. More sophisticated object detection algorithms along with object tagging methods and more detailed contextual information about the objects can be applied, however, the stimuli and BCI doesn’t need to change. As an example, the location of an object, the sequence of chosen objects and the time of the day could be considered to estimate the probability of being the target for each detected object.

4.3 FlashType™

FlashType™ is a context aware language independent typing brain interface [62]. It provides the user with a cursor, capable of navigating throughout a grid of symbols. The number of symbols is adjustable based on user preferences; default setting provides 28 symbols including the 26 English alphabet letters and space and backspace symbols. This keyboard consists of three main parts, Static Keyboard, dynamic Character Suggestion, and Word Prediction. By default, using a 6-gram language model and typing history, 7 highly probable characters and 3 most probable words are estimated and presented to the user. The separation among the keyboard and the stimuli makes the keyboard language independent. Hence, FlashType™ can be used to type in any language even to choose among symbols or icons. The arrangement of the characters in the Static Keyboard can be optimized based on the frequency of appearance of the characters or based on the user preferences. In the initial study, novice users have been able to reach rates of 6 seconds per character and build the habit of using different parts of the keyboard in just a few minutes.

4.3.1 User Interface Design

A user interface containing Stimuli, Visual Target, Visual Feedback and Stimuli Role Labels has been designed and presented on a computer display. Square checkerboards at the four corners of the screen, play the stimuli role. A keyboard section in the center has the role of the Visual Target and a section on the top center of the screen provides the user with the Visual Feedback. Figure 4.2 shows a sample of the screen. During the Calibration mode, defined in Section 2.4 only the stimuli and the yellow frame are presented to the user. A yellow frame around the stimulus, is an indication of the stimulus being the target of the epoch.
CHAPTER 4. APPLICATIONS

Figure 4.2: An example of the screen in the initial state of FlashType™. The top center rectangle provides the user with a visual feedback of what has been typed so far.

during Calibration mode, and the stimulus being selected as the intended stimulus during the other modes of operation. This specific arrangement of the user interface parts is chosen to utilize most of the screen and also maximize the inter stimuli distance. In addition, having the Visual Target and the Visual Feedback at the center makes it easier for the user to take a glance and see them all at once.

FlashType™ uses the cursor mode selection mechanism described in Section 2.6.2 where selection is done by moving and making a selection on a grid. In FlashType™, the grid consists of the three parts, Word Predictions, Character Suggestions and Static Keyboard. The starting point of the cursor in every epoch is the most probable character estimated by the language model. The most probable character appears as the left most character on the Character Suggestions row, the second row from the top of the keyboard.

4.3.2 Keyboard

Keyboard consists of three sections, Static Keyboard, Character Suggestions and Word Predictions. Figure 4.3 shows the Keyboard in the starting state.

- Static keyboard provides the user with the access to all the 26 letters in the alphabet, space and backspace, organized as a 4 by 7 matrix. While this is the default keyboard, options can be added or removed based on the user request and needs. The letters are ordered alphabetically for the purpose of illustration in this paper. However, the ordering can be changed according to user preferences and also be optimized based on language models on overall character usage. A typical extension of the Static Keyboard consists of 5 rows and 8 columns resulting in 40 options which can also include numerals or accented characters. The specific ratio of the rows and columns is chosen based on the resolution of the wide screen monitors. In general, there are no restrictions on the number of rows and columns. Consequently, this keyboard is not limited to the English alphabet, alphabet from any language, or even symbols and icons can be used.
Figure 4.3: The default keyboard provided by the FlashType™ in the initial state. The first part from the top, presents three predicted words. The second part, presents the top seven most probable characters based on the language model estimates. The third part, consisting of four rows presents the static part of the keyboard.

- **Character Suggestions** are shown in a row on top of the Static Keyboard. Character are suggested based on the conditional probability of the letters in the alphabet given what has been typed so far. Probabilities are calculated using an n-gram language model trained on the New York Times corpus. Language models can also become personalized based on the user activities, life style and common vocabulary. At every point in a sentence, 7 letters with the highest conditional probability are presented from left to right in the Character Suggestions row.

- **Word Predictions** at every point in a sentence, up to 3 predicted words are presented in a row above the Character Suggestions row. Word prediction utilizes the same n-gram language model to predict the top 3 probable words given what the user has typed so far. Words are predicted using a Viterbi [63] like search algorithm constrained on reaching a space or a backspace character.

### 4.3.3 Language Model Information Extraction

Language model provides a set of probabilities for all the letters in the alphabet, based on what has been typed so far. Here an n gram language model is used where the model estimates the conditional probability of each character based on the n – 1 characters that has been typed so far. In this study, a 6-gram model trained using a one-million-sentence (210M character) sample of the NY Times portion of the English Gigaword corpus is used. Corpus normalization and smoothing methods are described in [64]. Most importantly for this work, the corpus was case normalized, and Witten-Bell smoothing was used for regularization [65].

Here, after receiving the language model estimated probabilities for every character, probabilities are marginalized based on the four available options at the time. Similar to the available options, marginalizing the probabilities, also depends on the active dimension and active direction of the cursor. At every point on the keyboard, targets are divided among the available options. For example, when the vertical dimension and the down direction are active, the default start settings for every selection, available options are, Select/Switch,
Right, Down and Up. So based on the cursor location, the targets are reachable directly or by a combination of going down, up, right and the target at the current location of the cursor is reachable by Select.

Every target is assigned a reachability probability based on its relative location compared with the cursor location. Targets that are reachable from more than one direction, share their probability equally between different paths. At every point in time, the opposite directions are only available for the active dimension. So the targets are divided between the two directions of the same dimension. Every target belongs to the direction in which it is closer to the current cursor location. The prior probabilities for the four available options are calculated based on the probability of each target estimated by language model and its reachability probability.

In addition to priors calculated based on the current state of the cursor, the effect of the previous choices that the user has made to reach to the current state in each word can also be incorporated. Incorporating such information in the priors for the next decision will help the participant move towards their actual goal. Here, we have incorporated the history of the moves for each selection from the start point until a selection has been made. For example, starting with the first letter of a word, the history of the moves towards the selection of every character is considered. After a character or word is selected, the history is cleared and the new moves towards the next character or word are incorporated.

Figure 4.4 shows the graphical model representing the generative model of the collected data in epoch $e$. The goal here is to estimate the next command $s_{e_t}$. In this graphical model,

- $w_e$ is the context information based on the $n$ previously typed characters. Here $n$ is the order of a Markov model that is utilized by the available language model.
- $y_{e_t}$ is the desired cursor location at iteration $t$ of epoch $e$.
- $A$ represents the assignment. More specifically, user may assign a particular character to $y_{e_t}$, from the options available in the direction of $y_{e_{t-1}} \rightarrow y_{e_t}$.
- $L$ represents the layout of characters on the screen.

Figure 4.4: System graphical model in epoch $e$. 
CHAPTER 4. APPLICATIONS

- \( s_{t_e} \) is the decision at iteration \( t \) of epoch \( e \).
- \( X_{t_e} \) is the EEG evidence for decision \( s_{t_e} \).

Here we estimate the system state at each iteration in a maximum A-posteriori (MAP) inference mechanism. Hence, we need to calculate the posterior PMF over all available commands. Assume that \( \{X_{t_e}\}_{t=1}^t, X_{t_e}, w_e \) and \( L \) are observed. Then according to the assumptions presented in the probabilistic graphical model of the system we have,

\[
P(s_{t_e} | \{X_{t_e}\}_{i=1}^{t-1}, X_{t_e}, w_e, L) \propto P(s_{t_e}, X_{t_e} | \{X_{t_e}\}_{i=1}^{t-1}, w_e, L)
\]

\[
= P(X_{t_e} | s_{t_e}) \sum_{y_{t_e}} P(s_{t_e}, y_{t_e} | \{X_{t_e}\}_{i=1}^{t-1}, w_e, L) \tag{4.1}
\]

And also

\[
P(y_{t_e} | \{X_{t_e}\}_{i=1}^{t-1}, w_e, L) = \sum_{A_{t_e}} \left[ P(y_{t_e} | A_{t_e}, w_e, \{X_{t_e}\}_{i=1}^{t-1}, L)P(A_{t_e} | L) \right]
\]

We can write

\[
P(y_{t_e} | \{X_{t_e}\}_{i=1}^{t-1}, w_e, A_{t_e}, L) = \sum_{\{s_{t_e}\}_{i=1}^{t-1}} P(y_{t_e}, \{s_{t_e}\}_{i=1}^{t-1} | A_{t_e}, w_e, \{X_{t_e}\}_{i=1}^{t-1}, L)
\]

\[
= \sum_{\{s_{t_e}\}_{i=1}^{t-1}} P(y_{t_e}, \{s_{t_e}\}_{i=1}^{t-1}, \{X_{t_e}\}_{i=1}^{t-1} | A_{t_e}, w_e, L) \tag{4.3}
\]

Note that

\[
P(\{s_{t_e}\}_{i=1}^{t-1}, \{X_{t_e}\}_{i=1}^{t-1} | A_{t_e}, w_e, L) = P(\{X_{t_e}, X_{t_e}\}_{i=1}^{t-1} | \{s_{t_e}\}_{i=1}^{t-1}) P(\{s_{t_e}\}_{i=1}^{t-1} | A_{t_e}, w_e, L)
\]

\[
= \prod_{i=1}^{t-1} P(X_{t_e}, X_{t_e} | s_{t_e}) \sum_{\{y_{t_e}, A_{t_e}\}_{i=1}^{t-1}} P(\{s_{t_e}\}_{i=1}^{t-1}, y_{t_e}, A_{t_e}, \{X_{t_e}\}_{i=1}^{t-1} | A_{t_e}, w_e, L) \tag{4.4}
\]

The last term in eq. 4.4 can be simplified as

\[
\sum_{\{y_{t_e}, A_{t_e}\}_{i=1}^{t-1}} P(\{s_{t_e}\}_{i=1}^{t-1}, \{y_{t_e}\}_{i=1}^{t-1}, A_{t_e}, \{X_{t_e}\}_{i=1}^{t-1} | A_{t_e}, w_e, L)
\]

\[
= \sum_{\{y_{t_e}, A_{t_e}\}_{i=1}^{t-1}} P(\{s_{t_e}\}_{i=1}^{t-1} | \{y_{t_e}\}_{i=1}^{t-1}) P(\{y_{t_e}\}_{i=1}^{t-1} | \{A_{t_e}\}_{i=1}^{t-1}, w_e, L) P(\{A_{t_e}\}_{i=1}^{t-1} | L) \tag{4.5}
\]

Hence we have,

\[
P(y_{t_e} | \{X_{t_e}\}_{i=1}^{t-1}, w_e, A_{t_e}, L)
\]

\[
\propto P(y_{t_e} | A_{t_e}, w_e) \sum_{\{s_{t_e}\}_{i=1}^{t-1}} \prod_{i=1}^{t-1} P(X_{t_e} | s_{t_e}) \sum_{\{y_{t_e}, A_{t_e}\}_{i=1}^{t-1}} \prod_{j=1}^{t-1} \left[ P(s_{t_e} | y_{t_e}) P(y_{t_e} | A_{t_e}, w_e) P(A_{t_e} | L) \right] \tag{4.6}
\]
CHAPTER 4. APPLICATIONS

In this set of equations,

- \( P(X_{e_t} | s_{e_t}) \), represents the EEG observation likelihood for a particular command.
- \( P(s_{e_t} | y_{e_t}) \), is the probability by which the user selects a particular command to take the cursor to position \( y_{e_t} \).
- \( P(A_t | L) \), represents the probability of a particular character assignment by user, given the layout of the targets and the current cursor position.
- Assuming a particular assignment, and the context, \( P(y_{e_t} | A_{e_t}, w) \), would represent the probability of a character which is assigned to \( y_{e_t} \) according to the context information.

At most one decision is made for every epoch. Decisions are made using the MAP estimate decision criterion described in Section 3.5.2. Experiment results described in Section 5.4 illustrate how the language model information incorporation helps the users as they maneuver over the keyboard parts. In addition, the experiment also shows how the participants make use of different keyboard parts.

4.3.4 Operation Modes

FlashType™ has three different modes of operation. In general, all modes of operation, except the Calibration mode, depend on the data collected during a Calibration session.

- **Calibration**: is the same routine discussed in Section 2.4 is followed here. In the Free Run and Copy Phrase modes, the stimulation will continue until a decision is made by the classifier.
- **Copy Phrase**: is the mode in which the user is given a sentence with a highlighted word. The phrases are chosen based on the language model estimates for the target word in the phrase. The phrases are categorized based on the difficulty level of their target word. The difficulty level is calculated based on the probabilities estimated by a language model. The more likely a word gets the lower the mastery level will be. In other words, words that are more frequently used are assigned higher probabilities by the language model compared to words that are rarely used. This method implements the Mastery task training method described in Section 3.4.2 in the concept of words and phrases. The user has to type the highlighted word to complete a level and proceed to the next. Every user, starts by typing the words that are frequently used and upon successful completion, may proceed to the next word which would be of the same or higher difficulty.
- **Free Run**: is the mode in which the user has the freedom of writing any arbitrary character leading to words or sentences. This mode is usually used after the user has successfully passed a few Copy Phrase tasks.
4.4 FlashPlay™

FlashPlay™ provides a simple brain interface for training and entertainment. Here we illustrate a simple game, maze, in which the goal is to pass through a highlighted path guiding a mouse indicator to a cheese icon. The application presents a virtual environment such as a maze or a floor map. Using a virtual environment makes the setup much simpler. While playing the maze game, stimuli roles are Go Left, Go Right, Go Down and Go Up. Figure 4.5 illustrates sample screenshot of two maze games with black & white and black & white stimuli. While performing this task, if the users make a mistake and move the mouse indicator in to an incorrect path, highlighted in pink, they have to correct their mistake by bringing the mouse indicator on to the correct path and move towards the cheese icon at the end of the path. Mazes can be generated randomly every

![Sample screenshots of two maze games](https://example.com/maze-screenshots.png)

Figure 4.5: Sample screenshots of two maze games using red & green and black & white stimuli. In each maze the correct path is highlighted in yellow and the incorrect paths are highlighted in pink.

time and parameters such as the number and frequency of changes in the directions, the total length of the correct path, the frequency and length of the incorrect paths along the way can be utilized to generate simple or hard mazes. In addition, taking advantage of the probabilistic classifier, the difficulty of the same maze can also be altered using the prior probabilities assigned to the possible choices at each step.

Adjusting the difficulty level is done by keeping the ratio of the prior probability of the correct option and the maximum prior probability of the incorrect choices between an upper and lower bound. Table 4.2 shows the boundaries considered to generate 5 difficulty levels. Lower level Mastery task are easier to complete, in the sense that the system adjusts the prior probability of the correct choice to make it more probable. The help provided by the system decreases as the Mastery task levels increase and after level 3 the system starts to make the correct choice less likely than at least one of the incorrect choices. Completing these Mastery tasks helps users to build the habit of using the system during the easy levels and also get used to shifting their attention and focus on the stimuli to generate strong responses during the harder levels.

An experiment described in Section 5.1 demonstrates the effectiveness of the Mastery tasks on user training and performance.
### Mastery Task Level

<table>
<thead>
<tr>
<th>Mastery Task Level</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>0.5</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table 4.2: Mastery task probability boundaries.
Chapter 5

Experimental Results

Here we present the results of a series of experiments exploring the effects of different system parameters such as confidence threshold value, static vs dynamic confidence threshold, visual cue and different input modalities. FlashPlay™ is utilized as the test bed for comparing input modalities and also evaluating the simulation based performance estimation capability. In addition, some experimental results are also presented utilizing the typing application FlashType™. In all the experiments participants consented and participated in a data collection session that followed a protocol approved by Northeastern IRB.

5.1 Confidence Threshold & Visual Cue

An experiment was designed utilizing FlashPlay™ in order to examine the effects of the confidence threshold value and visual cue on the user performance. Confidence value effect is examined by using two confidence threshold values of 0.75 and 0.9 and two methods of Static and Dynamic adjustments. Visual cue effect is examined by including and excluding the first trial in every epoch. Stimuli with black and white color pair with bit presentation rate of 60 pbs and control bit sequences of length 31 bit were used resulting in trial length of about half a second. So excluding first trial in every epoch translates to considering a half a second visual cue population time.

Ten healthy individuals, 5 male aging from 26 to 31 and 5 females aging from 23 to 29, with normal or corrected to normal vision participated in the experiment. Each individual participated in two data collection sessions lasting less than two hours each and collected on two separate days. Every data collection session started with a Calibration session. Four black & white stimuli were used mapping to Go Left, Go Right, Go Down and Go Up commands. During each data collection session, after a qualified Calibration data was collected, every individual was asked to complete eight mazes and then finish the session with a post Calibration. Mazes were chosen to be from 4 difficulty levels, covering Mastery tasks levels 1 to 4. Figure 5.1 shows the mazes used in this experiment and their difficulty levels.
Mazes presented in the top row were used during the first half of each session and the ones presented in the bottom row were used in the second half of each session. The same mazes with the same order were used during the second data collection session.

Mazes were ordered to have difficulty levels of 1, 2, 3, 4, 3, 4, 2, 1. Mazes were randomly generated following the same statistical criteria in terms of frequency of changes in the commands the incorrect paths. Difficulty levels were adjusted by adjusting the prior probabilities adjusted at each step using the method described in Section 4.4. The special order for the first half of the tasks, was chosen to help participants get used to the system starting with easier Mastery tasks and proceed to the harder ones. During the second half, the order was chosen to keep the difficulty higher at first and decrease the difficulty level towards the end of the session. The difference between the first and the second day were only in the system parameters. The base confidence threshold was set to 0.75 during both data collection sessions. However, during the second day, Dynamic confidence threshold adjustment was activated. Recall from Section 3.4.1 that dynamic confidence threshold considers two factors while adjusting the confidence threshold for each epoch, the minimum base confidence threshold and the maximum context prior probability. Resulting in a higher confidence threshold when the prior probability obtained from the context information exceeds the base confidence threshold.

Stimuli were specifically adjusted to present at least 3 trials in every epoch during the Mastery task. The upper bound on the number of trials presented in each epoch was controlled by the classifier. The minimum 3 trials per epoch was considered to provide the capability of assessing actual user performance under different classifier parameters. Actual user performances have been assessed under 8 different conditions. Two base confidence thresholds of 0.75 and 0.9 has been examined under static and dynamic confidence thresholds and
including and excluding the first trial of each epoch. In addition, the post Calibration was used to have a
coarse estimate of user fatigue. Actual user performance is calculated using the EEG evidence collected during
each epoch and at each step on the maze to make an intent detection having the classifier parameters set based
on different targeted values. Actual user performance is defined as ratio of the number of correct decisions and
total number of decisions made for each task. Figure 5.2 shows the actual accuracy achieved over 4 different
Mastery task levels using the PreCalibration and PostCalibration data and the average number of trials per
epoch during the first data collection session. Figure 5.3 shows similar results for the second data collection
session. Based on the results, typically, base confidence threshold of 0.9 and Dynamic confidence threshold
adjustment with visual cue consideration produces the best performance and at the same time uses a moderate
among of trials per epoch. Eye tracking data collected during the experiment and analysed offline showed
most participants spent about half a second searching over the screen and then focused on the target stimulus
at the beginning of each epoch. This time duration was in addition to the pause time considered between
task execution and stimulation for the next decision. So, although the participants were given two seconds to
see the execution of the task associated with the last decision and decide which command and accordingly
which stimulus to focus on, they had a wondering period during the first half a second of stimulation for
each epoch. The actual duration was different from one participant to the other, but, half a second seems
to be an accurate average estimation. Hence, the first trial having approximately half a second duration is
considered contaminated. In addition, the time required to fill up the visual cue would also fall in to the same
time window. These figures also clearly demonstrate the effectiveness of Mastery tasks were as the level
increases the accuracy or in other words the number of mistakes made increases, at the same time classifier
also uses more sequences to come to confident decisions.

As expected performance decreases a little as participants get tired, hence the performance decreases
when PostCalibration data is used to train the classifiers and assess user performances. But, most of the
time, when the Calibration session and the testing sessions are closer in time, user behaviour tends to stay
more similar. Furthermore, participants develop their skills as they use the system more and more. This
effect is more clear comparing performance of every individual during the first and the second day of data
collection. Overall, the main increase in the performance comes from visual cue consideration. The second
factor affecting the performance is the base confidence threshold value, 0.9 being more effective than 0.75.
And finally the dynamic confidence threshold adjustment can increase performance, especially during the
higher Mastery levels or in situations where context information is biased towards one of the targets.
CHAPTER 5. EXPERIMENTAL RESULTS

Figure 5.2: Actual participant accuracy and number of sequences used for 8 task under difficulty levels during the first data collection session. SC stands for static confidence threshold, DC stands for dynamic confidence threshold and VC stands for visual cue consideration. (Part I)
CHAPTER 5. EXPERIMENTAL RESULTS

Figure 5.2: Actual participant accuracy and number of sequences used for 8 tasks under difficulty levels during the first data collection session. SC stands for static confidence threshold, DC stands for dynamic confidence threshold and VC stands for visual cue consideration. (Part II)
Figure 5.2: Actual participant accuracy and number of sequences used for 8 task under difficulty levels during the first data collection session. SC stands for static confidence threshold, DC stands for dynamic confidence threshold and VC stands for visual cue consideration. (Part III)
CHAPTER 5. EXPERIMENTAL RESULTS

Figure 5.3: Actual participant accuracy and number of sequences used for 8 task under difficulty levels during the second data collection session. SC stands for static confidence threshold, DC stands for dynamic confidence threshold and VC stands for visual cue consideration. (Part I)
Figure 5.3: Actual participant accuracy and number of sequences used for 8 task under difficulty levels during the second data collection session. SC stands for static confidence threshold, DC stands for dynamic confidence threshold and VC stands for visual cue consideration. (Part II)
Figure 5.3: Actual participant accuracy and number of sequences used for 8 task under difficulty levels during the second data collection session. SC stands for static confidence threshold, DC stands for dynamic confidence threshold and VC stands for visual cue consideration. (Part III)
CHAPTER 5. EXPERIMENTAL RESULTS

5.2 Eye Tracking vs Code-VEP

An experiment was designed utilizing FlashPlay™ with two input modalities. The experiment consisted of two main parts, in which, the input modalities were different. During the first half, c-VEP was used as the input modality and during the second half eye tracking system described in Chapter 2.9. In each part, participants used the system to complete four different mazes. The mazes are designed randomly beforehand and chosen to have almost the same number of decisions per stimulus. Experiments were designed to be extremely similar with input modality being the only difference.

Participants used the c-VEP based system first and then used the eye tracking to perform the exact same tasks. Based on the stability of the Calibration, the calibration process might have had to be repeated a few times. A single EEG electrode placed at Oz, with the ground electrode on Fpz and reference electrode placed on the left earlobe was used to capture EEG activity at the rate of 256Hz. For the c-VEP based system, different strategies were used to achieve a Calibration data with good performance. Once a well performing Calibration data was achieved, it was used throughout the c-VEP tasks. Red and green color pair was used for the stimuli with bit presentation rate of 110 bps and control bit sequences of length 63.

However, for the eye tracking system, the system was so sensitive to even slight movements that there was a need to run the Calibration routine before every single task. In addition, the estimated location of the gaze was presented on the screen and participants were given the freedom of making slight adjustments to their head to match their gaze point to the estimated gaze point. The eye tracking parameters were saved for each participant and were loaded prior to each task to decrease the eye tracker calibration time, but, they couldn’t eliminate the need to make slight adjustments to the cameras and to run the on screen Calibration before every task.

The tasks were the same for both input modalities, having gone through them during the c-VEP part, participants were more familiar with the tasks during the eye tracking part. While the c-VEP system is able to incorporate the information about the context to boost the performance and the accuracy of the decisions, in this study, we have not used the context information to have a fair comparison.

Figure 5.4 illustrates four mazes that were presented to the participants. The mazes were presented to the participants in the same order when using different input modalities. In each maze the start point is marked with a red block and the end point is marked with a green block. The correct path is unique, 20 or 21 blocks long and highlighted in yellow. There are incorrect, dead-end paths randomly placed along the way and highlighted in pink. The task is to move the mice indicator from the the start point to get the piece of cheese placed at the end point. In the event that the participant made a mistake and led the mice into and incorrect path, the mistake had to be corrected by coming back to the correct path.

Stimuli roles for this task were, Go Left, Go Right, Go Down and Go Up starting from the top left corner and moving clockwise. Figure 5.5 illustrates the screen during the first maze task. A total timeout of 10 minutes were considered for each task.
5.2.1 User Experience

Participants sat in front of an 22 computer display at a distance of 80 cm. The height of the chair, head fixed setup and the computer display were adjusted such that looking straight the participant’s eyes were pointing at the top 1/3 of the computer display. After the experimentation session, every participant was asked to fill out a survey with a few questions about their preferred input modality and their reasoning. Although the participants in this study were all healthy individuals with normal or corrected to normal vision, we tried to include individuals with a variety of vision statuses such as wearing eye glasses or contact lenses.

Ten healthy individuals with normal or corrected to normal vision, 6 females aging from 23 to 30 and 4 males aging from 25 to 27 participated in this study. Among the 10 participants, 8 voted in favor of the c-VEP system and 2 were in favor of the eye tracking system. Participants were asked to fill out a questionnaire after completing all the tasks which included the following questions. Q1: Which input modality you prefer? Q2: Which system you think was following your commands better? Q3: Which system you found faster? Q4: Which system was easier to use? Table 5.1 summarizes participants’ responses and vision statuses. Participants particularly were not comfortable with the head fixed setup for the eye tracker.
Figure 5.5: Illustration of the screen when task Maze 1 starts.

Table 5.1: User response summary, B stands for Brain Interface (c-VEP), E stands for Eye Tracker, G stands for glasses and CL stands for contact lenses. Calibration accuracy is estimated based on the Calibration data.

<table>
<thead>
<tr>
<th>Participant ID</th>
<th>Vision Status</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>c-VEP Calibration Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>Corrected (CL)</td>
<td>B</td>
<td>E</td>
<td>B</td>
<td>B</td>
<td>98.5</td>
</tr>
<tr>
<td>P2</td>
<td>Normal</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>96.5</td>
</tr>
<tr>
<td>P3</td>
<td>Normal</td>
<td>E</td>
<td>E</td>
<td>B</td>
<td>B</td>
<td>99.5</td>
</tr>
<tr>
<td>P4</td>
<td>Corrected (G)</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>95.5</td>
</tr>
<tr>
<td>P5</td>
<td>Normal</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>91.5</td>
</tr>
<tr>
<td>P6</td>
<td>Normal</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>86.5</td>
</tr>
<tr>
<td>P7</td>
<td>Normal</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>93</td>
</tr>
<tr>
<td>P8</td>
<td>Corrected (G)</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>93.5</td>
</tr>
<tr>
<td>P9</td>
<td>Corrected (CL)</td>
<td>B</td>
<td>E</td>
<td>B</td>
<td>B</td>
<td>93.5</td>
</tr>
<tr>
<td>P10</td>
<td>Normal</td>
<td>E</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>93.5</td>
</tr>
</tbody>
</table>

5.2.2 Results & Discussion

All participants except the sixth participant were able to finish all the tasks using both input modalities. Our sixth participant, couldn’t finish maze number 2 using c-VEP input modality and made only 30% progress. The same participant had an even harder time using the eye tracker. This participant could not finish mazes 1 and 3 using the eye tracking input modality and made 0% and 50% progress respectively. To have a comparison between the two input modalities some statistics were extracted from the data. To summarize the results better, we averaged some of the factors with in the four tasks for each participant. First, we start with the duration of setup and calibration. Due to sensitivity of the eye tracker, although the data was saved after the first calibration, there was a need to make some adjustments before every task. However, except for participants P6 and P9, the same Calibration data was used for the c-VEP input modality for all the tasks. These two participants needed a second Calibration mostly due to the changes in their behaviour. Figure 5.6 shows the
CHAPTER 5. EXPERIMENTAL RESULTS

Figure 5.6: Average setup and calibration duration for different input modalities. Two average setup durations are calculated for c-VEP method including and excluding the one time calibration duration. The average setup and calibration duration for each input modality. Since the c-VEP input modality mostly uses only a single Calibration data, collected over a duration of 181 seconds with the settings used, the average is provided including and also excluding this time duration. The duration of the Calibration becomes negligible as the number of tasks increase. Before every task, there has been a quick EEG signal monitoring to make sure of an acceptable electrode connection. For all then participants except P9, the total calibration time for the eye tracker has been longer than than c-VEP. While the setup times excluding the c-VEP Calibration sessions which rarely require repetition, are considerably less than eye tracking setup time.

Next, is the average time each participant spent to complete the tasks with the same input modality. Figure 5.7 shows the average time duration of the completed tasks performed with the same input modality.

Figure 5.7: Average completed task duration for different input modalities.
The results show that the average time required to complete a task using c-VEP input modality is less than eye tracking input modality except for participant P7. In this calculation only the completed tasks were considered.

Next factor is the accuracy of the decisions made by the two input modalities. Figure 5.8 shows the average accuracy of the decisions made by the two different input modalities performing the four tasks. The average accuracy for the c-VEP system is consistently above about 80% and even most of the time above 90%. For most users the accuracy of both input modalities is close, however, there are some exceptions like participants P1, P5, P6 and P7 for whom the difference is more.

Next, we would like to present the number of infeasible decisions made by the two input modalities. An infeasible decision is defined as a command that is not executable, i.e. when the mice indicator is facing a wall it is not possible to pass through the wall. In such situations, a command to move the mice indicator through the wall is considered as an infeasible command. We should mention that the c-VEP system, using the context information, is capable of avoiding such situations by decreasing the prior probability of the infeasible options, however, in this study we did not use this functionality to focus more on the differences of the input modalities. For the eye tracking system, choosing a point out of the stimuli area is also considered as an infeasible command. Figure 5.9 shows the average number of infeasible decisions made by each input modality.

Between the two classification methods used for intent detection using the eye tracker, in this study fixation method was only able to detect an intent for seven times and the rest of the intent detections (> 100 per participant) were done by the second eye tracker based classification method. Participants in general had a hard time fixating on the points and even with the relaxed fixation parameters, such as drift speed and distance still the second method was able to perform faster.

It is hard to find a definite answer about the best input modality. Our study shows that even healthy individuals, who have complete control over their body and more specifically eyes, might prefer c-VEP.
CHAPTER 5. EXPERIMENTAL RESULTS

Figure 5.9: Average number of infeasible decisions made by the two input modalities.

over eye tracking. This is considering the fact that the participants were given the freedom of making slight adjustments to their position to match the estimated gaze point with their actual gaze point during the tasks. In this study, we tried to keep the application and most of the settings the same to make the experiments differ just in the input modality. While one might claim that there might exist better eye tracking or EEG based interfaces, here, we have tried to make use of a fast and reliable system to make the comparison as fair as possible. Factors such as system reaction time, the irritation from the stimuli and tiring effect of staying still were the top factors affecting participant preferences. Adjusting the eye tracker for individuals wearing eye glasses was specially very hard. Eye tracking system capable of tracking head movements are expected to provide ease of use and more comfort.

Based on the results, setup and calibration times for the c-VEP system are considerably less than the eye tracking system. The short setup time of the c-VEP system partly comes from the fact that the system uses only a single EEG electrode. The Calibration was more stable for the c-VEP input modality, while, the eye tracking system was extremely susceptible to even slight head movements. Code-VEP system performed faster for most of the tasks, even though, the flickering effect made some users feel it took longer time than the eye tracking tasks.
CHAPTER 5. EXPERIMENTAL RESULTS

5.3 Simulation based Performance Estimation

This section illustrates the results of an experiment conducted to verify the Simulation tool described in Section 2.8.1. Performance estimation tool can estimate the average participant performance. Hence, to have a reasonable comparison, there is a need to have many samples of the actual user performances. However, gathering multiple repetition of the tasks becomes infeasible quickly when the number of repetitions increases. In this study, FlashPlay™ is utilized to provide a simple, easy to setup task. Every participant completed four mazes illustrated in Figure 5.4. Ten healthy individuals with normal or corrected to normal vision, 4 males aging from 25 to 27 and 6 females aging from 23 to 30 participated in the study. Red & green stimuli with bit presentation rate of 110 bps and m-sequences of length 63 bit were used. Participants started the session with a Calibration session. A minimum of 85% Calibration performance was required to consider the Calibration successful. Participants were given at most 10 minutes to complete each task.

Using the Calibration data and the Simulation tool, Monte Carlo simulations with 10000 iterations were ran for each task. The actual and estimated performance were compared in terms of the average accuracy, average number of trials used per decision and average number of mistakes made. All the participants were able to complete all the four tasks except one participant during one of the tasks who proceeded only about 30%. Figure 5.10 shows the actual and the estimated average performance for each participant, sorted based on the participants’ Calibration performance. Single actual performance measures for each task performed by each participant is marked with an asterisk. In addition, mean and median of the performance measures are also displayed to visualize the overall behaviour. Like every sampling method, a single performance sample might be an outlier. The 90% confidence intervals are estimated based on the performance estimations performed by Monte Carlo simulations performed by the Simulation tool described in Section 2.8.1.

Actual performance marks spread around the 90% confidence interval with their median closer to the lower bound. A closer look also indicates that the actual performance marks and the estimated confidence interval overlap more for individuals with the higher Calibration performances. From Figure 5.10 comparing the actual performance results with the estimated ones, we observe that under and over performance estimations are possible. We conducted a two tailed Wilcoxon signed-rank test between simulation and experimental results. Since median is more robust to outliers, we compared the median of the actual performance with the estimated performance values. The test results showed that with p-value of 0.01, the simulation over estimated the actual accuracies, but with p-value 0.43 the actual accuracies are not significantly different than the lower 90% confidence level of the estimated values. Moreover, with p-value of 0.11, estimated number of trials per target are not significantly different than the actual values. One of the factors playing a crucial role in the accuracy of the performance estimates is the user behaviour. Simulation tool expects a similar behaviour from the user during the Calibration and the actual tasks. Since the Simulation tool uses the Calibration data to generate new samples during the Monte Carlo simulations, a deviation in the behaviour during the actual task will result in decreased performance. This behaviour consists of the factors such as the way users focus on their target stimulus and the strategy they follow as well as how fast they react and are able to focus and pay
attention. The inconsistencies in the behaviour also affect the actual performance. If a user changes their behaviour during the actual task, classifiers expecting the same behaviour would not be able perform as well as expected. Hence, in such situations, a new Calibration data should be acquired and users should always be encouraged to keep their behaviour the same. In addition, fatigue can also affect actual user performance.
Figure 5.10: Actual and estimated user performances using the Simulations over 4 different tasks. (a) Average accuracy of the decisions, (b) Average number of trials per decision and (c) Average number of mistakes.
CHAPTER 5. EXPERIMENTAL RESULTS

5.4 FlashType™

FlashType™ was utilized in a simple experiment in which participants had to complete 10 Copy Phrase tasks. Three healthy participants, a 26 years old male and two 24 and 26 years old females with normal or corrected to normal vision, consented and participated in a data collection session. The data collection session started with a Calibration session. Based on the user performance, estimated from the calibration data, and their level of familiarity with the system, the Calibration session might be repeated one or two times. For this study, bit presentation rate of 110 bits/second and 63 bit long m-sequences have been used, which, translate to a trial duration of 0.57 seconds.

Once a qualified Calibration performance was reached, 85% or more, the session proceeded with a training session. The main purpose of the training session was to familiarize the participants with the role of each stimulus during the task. During the training session, a couple of example tasks were performed with a longer time gap between the decisions. The longer time gap, helped participants to recognize the task performed by the system, decide what they want to do next, and how they should act towards their next choice. Like any other user interface, the training phase duration and the level of familiarization with the system is user dependent. However, once the user gets used to the system, they can easily remember the role of each stimulus. During the training phase, healthy individuals, were provided with the control of the mouse cursor to move the cursor over their target stimulus representing their choice.

Every user had to type 10 words, each word in an individual sentence. The words have been chosen from 5 different difficulty levels to evaluate the effect of the language model. The following 10 words were presented to the participants “the, and, with, will, seat, between, seen, please, buys, makeup”. The difficulty level increases by one after every 2 words.

5.4.1 Results & Discussion

Calibration performance estimated for the three participants were, 95.5, 96 and 97 percent. All the participants were able to type all the words. The number of cursor movements towards every selection, depends on the location of the target character in the Static keyboard, whether it has been among the suggested characters or predicted words and also the route chosen by the participant to reach their desired target. On average, the time spent to type a character during the given 10 tasks, by the three participants have been 6.2, 7.6, and 11 seconds. However, the longer a user uses the system the better they get using the Reverse option and moving in the opposite direction to reach their target location faster. Figure 5.11 shows the overall performance of the prior probabilities for each word and for each participant. For each word, prior probabilities corresponding to the correct choice, and the maximum of the prior probabilities corresponding to the incorrect choices, provided at each step, are averaged and displayed. In addition, the number of times that the prior assigned to the correct choice had been greater than the rest of the priors, has been counted and displayed as the constructive prior percentage. Figure 5.11 shows the number of decisions each participant had to make to type each word. There is a clear direct correlation between the number of moves needed to type each word and
CHAPTER 5. EXPERIMENTAL RESULTS

Figure 5.11: Parts (a), (b) and (c), show the average value of the prior probabilities for the correct choice, the average value of the the maximum prior probability of the incorrect choices and the percentage of the number of times when the prior assigned to the correct choice has been greater than the maximum of the priors assigned to the other choices for typing each word. Part (d) shows the number of decisions each participant made to complete the task.
CHAPTER 5. EXPERIMENTAL RESULTS

the word’s difficulty level estimated by the language model. Figure 5.12 shows how the prior probability of the

![Figure 5.12](image)

Figure 5.12: Prior probabilities estimated by the language model for every character on the screen at every stage of typing the word "seen". Each plot demonstrates how the prior probabilities evolve as the cursor moves towards the correct target. The solid thick line shows the prior probability of the correct choice and the dashed lines show the prior probability of the other choices.

characters evolves after every movement of the cursor. Each plot demonstrates the prior probability variation towards writing one character. In this example, "seen" has an estimated difficulty level of 4, thus, it hasn’t been predicted as a probable word. Note that, depending on the sentence, the same word might have a different difficulty level. Figure 5.13 shows the prior probabilities assigned to each target on the keyboard at every move towards typing the word With. Every row belongs to the selection of a character except the last row, which belongs to the selection from the predicted words. As the colors indicate, the adaptive prior assignment based on the location and route of the cursor and the language model estimates, helps to increase the estimated probability of the correct target character as the cursor gets closer and closer to its location. Here, with, has a difficulty level of 2 based on the language model estimates, and it has been predicted by the word prediction

86
CHAPTER 5. EXPERIMENTAL RESULTS

Figure 5.13: Prior probabilities estimated by the language model for every character on the screen at every stage of typing the word “with”. Every row belongs to the moves towards choosing a letter, except the last row, which belongs to choosing the complete word from the predicted words row.

The algorithm after the second character is selected. To better evaluate the performance of the estimated priors and probabilistic incorporation of the information from every movement towards the target character, snapshots of the probability estimates, during selection of the word *buys* with the difficulty level of 4 are illustrated in Figure 5.14. Despite the fact that, the word *buys* is not among the few most probable words, and the user has to choose every character, the incorporation of the route taken by the user in the past moves helps to adjust the prior probabilities towards the correct choices. Looking at the plots, every row in Figure 5.14 shows that as the cursor gets closer to the correct target, the estimated prior probability of the correct target character also increases making the selection easier for the user.

Figure 5.15 shows the participants’ usage of different parts of the keyboard, *Static, Character Suggestions*, and *Word Predictions* to write the 10 required words. Although, the sample size in this experiment is limited to only 10 words, uniformly selected from different difficulty levels, it can provide a good measure about the effectiveness of each part of the keyboard. *Word Predictions*, can be more helpful when the intended words are frequently used, and therefore, more predictable by the language model. However, since the number of the characters suggested is greater, they can cover a broader range and be more helpful when the intended word is less frequent. The *Static Keyboard* also plays an important role, while visually static, the prior probabilities assigned to each target are dynamically changed based on the location of the cursor. One of the main advantages of the *Static Keyboard* is that all the targets are available at predefined fixed locations where it would be constant and easily memorizable by the user. FlashType™ uses only one electrode placed on top of the visual cortex at Oz and 4 stimuli placed at the four corners of the display, clearly demonstrating the
CHAPTER 5. EXPERIMENTAL RESULTS

Figure 5.15: The overall usage of the keyboard parts, Static Keyboard, Character Suggestions, and the Word Predictions by each participant.
CHAPTER 5. EXPERIMENTAL RESULTS

performance of the cursor based selection described in Section 2.6.2. The separation of the keyboard and stimuli makes FlashType™ invariant to the alphabet and language. Although, FlashType™ uses a cursor based hierarchical selection method, due to the high accuracy of every action, Character Suggestions and Word Predictions, the average required time per character is shorter or comparable to that of similar systems on one of tests like those reported here [66]. The hierarchical decision making mechanism also makes the system more robust to errors. Having multiple decisions involved in one selection, unless the error happens while selecting a target, there is always a recovery chance.

All the participants were able to type all the words, with an average time per character of 6 to 11 seconds. Our third participant needed 11 seconds per character, since he preferred using the right and down directions rather than switching the direction and approaching targets from the opposite direction. The adaptation of the user to the system plays a significant role on overall performance. For example, a feature such as reversing the direction of the horizontal or vertical movement, in the specific configuration of the keyboard in this study, can decrease the time required to reach the target 50% to 85% varying based on the location of the target.
Chapter 6

Discussion & Future Work

In this dissertation, we have designed a noninvasive, context aware BCI system. Taking advantage of code-VEP responses, fast, robust, reliable and effective stimuli are designed and optimized. The stimuli have been tested with trials as short as 281(ms). Different opposite color pairs have been evaluated and shown the red and green color pair produces stronger responses as opposed to blue and yellow or black and white in participants with normal vision. However, the black and white color pair stays a general viable choice for individuals who are color blind. Generative models are used to model EEG responses to the designed stimuli and resulted in performances above 90% or even higher in most cases.

The inference engine in FlashLife™ uses a recursive Bayesian mechanism to compute the posterior probability mass function (PMF). In this framework one can use a prior PMF obtained from the context information (that can be defined based on the target application) to compensate for variations in likelihood estimates imposed by noisiness of EEG. These priors would be probabilistically fused with the likelihoods obtained from distributions defined over EEG observations to calculate the posterior PMF. The recursive nature of this framework would allow adaptively collecting more data until reaching a confident inference from a maximum a posteriori state estimation.

Moreover, the generative model proposed for class conditional EEG distribution, provides the capability of running Monte-Carlo system simulations using actual supervised data collected from human subjects. These models are used in a Simulation tool to estimate user performance based on an actual Calibration data without the need for the actual user to perform the task. The Simulation tool can save costly and time consuming experiments and help fine tune system parameters and validate models.

In addition, separating the stimuli and the target applications and two modes of Immediate and Cursor based selection have made it possible to employ FlashLife™ in different applications to address the communication and control needs of an individual in the everyday life. These applications include,

- FlashNav™, a context aware navigation brain interface; It can be used to navigate a wheelchair or control a robot remotely. Information such as environment map, objects and locations of interest and
user habits can be used to boost the probabilistic decision making performance. In addition, destination selection along with autonomous navigation and collision avoidance mechanisms can decrease the cognitive load on the user.

• **FlashType™**, a context aware language independent typing brain interface; It provides the user with a cursor, capable of navigating throughout a grid of symbols. The number of symbols is adjustable based on user preferences; default setting provides 28 symbols including the 26 English alphabet letters and space and backspace symbols. This keyboard consists of three main parts, Static Keyboard, Character Suggestion, and Word Prediction. By default, using a 6-gram language model and typing history, 7 highly probable characters and 3 most probable words are estimated and presented to the user. FlashType™, incorporates all the EEG collected from the user while navigating throughout the keyboard to make every selection. The separation among the keyboard and the stimuli makes the keyboard language independent. Users can rearrange the symbols in the Static Keyboard as they prefer. In the initial study, novice users have been able to reach rates of 6 seconds per character and build the habit of using different parts of the keyboard in just a few minutes.

• **FlashGrab™**, a context aware object manipulation brain interface; Using Baxter, a low cost humanoid robot, and image processing techniques, graspable objects are detected and labeled with numbers. A video feed shows the robot perception with the overlaid labels to the user. Depending on the number of graspable handles, an immediate or cursor based decision will be made by the user.

• **FlashPlay™**, an interface to a virtual environment such as a maze or a floor map; Training and entertaining the user are the main goals. Using a virtual environment makes the setup much simpler. A series of Mastery tasks have been designed with different difficulty levels, taking advantage of the probabilistic classifier and the virtual environment, to help the users to build the habit of using the system and attending to the stimuli effectively.

In FlashNav™, context information can come from sources such as a map of the environment and user habits. In FlashType™, context information comes from a language model, and other possible contextual information available. In FlashGrab™, context information can come from the surroundings, location, time of the day. In FlashPlay™, context information comes from the virtual environment. A series of Mastery tasks with different difficulty levels have been designed to train potential users. Mastery task ranging from easy to hard, help the user to start using the system and build the habit of focusing on the stimuli.

The stimuli being visual, an eye tracker has been employed to provide eye tracking as an alternative input modality. Experimental results using a head fixed eye tracking system have shown, even healthy individuals might prefer code-VEP to eye tracker. The preference mostly comes from the sensitivity of the eye tracking system to head movements and longer setup time. The two input modalities, code-VEP and eye tracking, both have shown to be effective and provide similar performances.

In FlashLife™, we have shown the possibility of having application independent, robust and fast stimuli capable of supporting different applications covering basic communication and control needs in everyday life.
CHAPTER 6. DISCUSSION & FUTURE WORK

Although, different brain responses might be better suited for specific applications, having the same stimuli in different applications makes it easier for an individual to build up the habit of using that stimuli.

6.1 Future Work

Future work can be separated into stimulus design and classification method improvement categories. While the stimuli in FlashLife™, checkerboards, are independent from target applications, they have a dependency on the potential user’s vision capabilities. Based on the symmetric visual field assumption, the default settings for stimuli produce a symmetric stimuli. However, in cases of asymmetric visual fields, it is possible to modify the size and the position of each stimulus to compensate for visual deficits. Hence, it would be best to have the results of an in-depth eye examination of the potential users before they start using the system. Eye examinations such as Multifocal Electroretinography (mfERG) and Multifocal Visual Evoked Potentials (mfVEPs) would be among the most informative studies for the purpose [67]. As an example, it would be possible to avoid damaged parts of the retina by changing the position of the stimuli. Furthermore, considering gaze dependencies, attention monitoring could enhance the stimuli strength.

Building a general visual response model to the stimuli using the results of multifocal studies would help to build and adapt estimated template responses and possibly shorten the Calibration duration even further. Classification performance can also be enhanced by adding an attention monitoring information source during the joint inference. Evaluation of the stimuli with covert attention for individuals suffering from gaze control issues would be a great step towards providing them with communication and control means. Lastly, multimodal inputs, stimulating audio and visual cortex for example, can be explored in order to enhance and broaden the classification and the potential target user population.
Bibliography


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


