# **FROM 1st June TO 7th June**

# **Project ID:**

# **2021J\_BV01\_BCI Browser**

# **Project Title:**

# **Design and development of Brain Computer Interface Browser on Web and Mobile**

# **Summary:**

* A **P300 brain-computer interface** (**BCI**) is a paradigm, where text characters are decoded from event-related potentials (ERPs). In a popular implementation, called **P300** speller, a subject looks at a display where characters are flashing and selects one character by attending to it.
* P300-based BCI is one of the most common BCI systems, as the P300 potential is easy to be stimulated.
* In the P300 EEG signals preprocessing part, the PCA is used to retain the data features of the original P300 EEG signals, which reduced the dimension of the original signals and reduced the computational cost of subsequent algorithms
* Result of the developed paradigm is that the developed program accurately implements the stimulus sequencing required of the P300 Speller. However, stimulus flash times and ISIs were obtained to determine compliance with the stimulus timing requirements.
* This work investigates the implementation of the P300 Speller using web development languages. The developed paradigm is executable on both laptop and Android variants of the Google Chrome browser.
* Possible avenues of future work- The developed paradigm must be integrated to data collecting and signal processing elements to allow for a full BCI implementation that includes training and testing sessions.

# **Detail:**

# **Detailed methodology used**

**P300-BASED CHARACTER INPUT SYSTEM**

Brain–computer interfaces (BCI) can provide a direct communication method between the brain and a computer or other external devices. There are several types of electroencephalograms (EEG) signals used in BCI, such as P300 potential, steady state visual evoked potential (SSVEP) , motor imagery (MI) , and so on. Specifically, P300-based BCI is one of the most common BCI systems, as the P300 potential is easy to be stimulated. Compared with other signals, the P300-based BCI system has some advantages:

(1) P300 signal is extremely easy to measure and non-invasive;

(2) less training time;

(3) suitable for most subjects, including those with severe neurological diseases; and

(4) users only need to provide a simple control signal. It can implement a variety of different functions, and can even be used in the home of people with disabilities.

The P300-based character input system contained a 6\*6 matrix of visual stimulation interface, which was composed of English letters, numbers, and spaces. Before the experiment, the subjects were told that a specified character in the visual stimulator was the target character, and each experiment randomly assigned a character. During the experiment, the subjects were asked to keep an eye on the target character position in the visual stimulator, while any row or column in the visual stimulator flashed randomly. When the target character’s row or column was flashing, a positive potential (called P300 ERP) related to the event could be detected in the subject’s scalp (about 300 ms after receiving the stimulus); if not, the detected EEG data were non-P300 event-related potentials (N-P300 ERP) . In addition to this standard speller system, there are other paradigms, such as row-column (RC) paradigm, single character (SC) paradigm, region-based(RB) paradigm, and so on. For all these systems, how to identify quickly and accurately is critical to improving the performance of BCI systems.

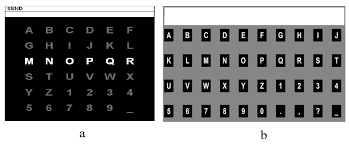
Due to the collected P300 signals often being high dimensional and feature dependent, some methods were proposed to enhance the feature extraction. PCA (Principal Component Analysis), a principal component analysis method, is widely used in feature extraction and data dimensionality reduction. The principle of PCA is to transform the original signal matrix into a covariance matrix through linear transformation , and obtain a new signal matrix by filtering the eigenvalues and eigenvectors of the matrix . The new signal matrix retains some of the most important original signal features in the original signals matrix, and eliminates noise and unimportant features to achieve the purpose of dimensionality reduction. In recent years, many researchers have applied the PCA to reduce the dimensionality of the obtained EEG signals.

**Method**

The Dataset Two sets of experimental data were analyzed. One was the dataset in the BCI Competition III provided by the Wadsworth Research Center NYS Department of Health . The other was provided by South China University of Technology using a different paradigm. There are two and ten subjects, respectively, in the two datasets. All subjects are healthy persons, who were selected randomly.

. Dataset I: The graphical user interface (GUI) of the competition was presented in Figure 1a, which is a 6\*6 character matrix. When the experiment began, each of the 12 rows and columns flashed randomly. A flashing lasts 100 ms and the interval between two flashes is 75 ms. A subject was asked to focus on the target character, and silently count the flashing repetitions of the row and column containing the target character. Each row or column repeats 15 times when outputting one character. The dataset was consisted of one training (85 characters) and one test (100 characters) sets for each of the two subjects A and B. All EEG signals were collected by a 64-electrode scalp, which were bandpass filtered from 0.1–60 Hz and digitized at 240 Hz. The information details can be found in the BCI competition webpage.

Dataset II: The second dataset was collected in the laboratory from South China University of Technology using a 4\*10 paradigm (see Figure 1b). Different from the first dataset, each character flashed separately and randomly. A flashing lasts 100 ms and the interval is 30 ms. A subject was asked to focus on the target character, and silently count the flashing repetitions of the target character. Each character in the paradigm repeats 10 times when outputting one character. The dataset consisted of one training (20 characters) and one test (30 characters) set for each of ten subjects. All EEG signals were collected by a 32-electrode scalp, which were bandpass filtered from 0.1–60 Hz and digitized at 250 Hz.



**Data Preprocessing**

As the raw EEG signals are weak and mixed with non-EEG signals or background noise, the raw data should be preprocessed first. In order to remove the influence of this interference information , we used the 8th-order bandpass Butterworth filter to filter the collected raw data and intercept the potential frequency to 0.1–20 Hz.

**PCA Algorithm**

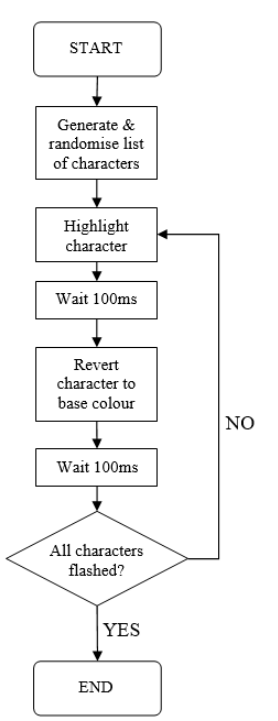
PCA transforms the original data into a set of linearly independent data vectors in various dimensions through linear transformation, which can be used to extract the main feature components of the data and often be used for dimensionality reduction of high-dimensional data. Therefore, before EEG signals were input into the neural network, we used the PCA algorithm to reduce the signal dimension. After preprocessing of the raw signals, a data matrix X is obtained, of which the abscissa and ordinate are the time and space domains, respectively. We input the matrix X into the PCA algorithm and calculated the covariance matrix (Cov(X)); then, the eigenvalue eigenvectors of the covariance matrix were obtained. We could select a matrix of eigenvectors corresponding to the features with the largest eigenvalues. In this way, the data matrix could be transformed into a new space, and the dimension reduction of the data features could be realized. Through dimensionality reduction using PCA, the size of feature vectors changed from 64\*240 and 30\*160 to 64\*120 and 30\*80 (two datasets respectively).

**P300 SPELLER IMPLEMENTATION USING WEB DEVELOPMENT LANGUAGES**

The expression of the P300 Speller in web development languages allows for implementation on all devices with web browsers regardless of operating systems. These devices include desktops, laptops, tablets and smartphones.

The P300 Speller Paradigm itself is an **oddball paradigm** in which rare target stimuli are presented amongst frequently delivered non-target stimuli.

The Speller Paradigm requires subjects to view a single matrix element during the randomly ordered flashing of all matrix elements . When a matrix element the subject is focusing on is flashed, the P300 response is evoked and the detection of the P300 is used to identify user focus .



P300 Speller Paradigm Flow Chart

There are 3 main client-side programming languages that are used to implement webpages:-

1. Hyper Text Markup Language (HTML)

|  |
| --- |
| Content |
| Layout |
| Tables |
| Images |
| Paragraphs |
| Forms |

2. Cascading Style Sheets (CSS)

|  |
| --- |
| Colour |
| Font Size |
| Positioning |
| Margins |
| Alignment |

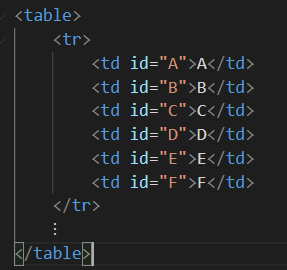
3. JavaScript (JS)

|  |
| --- |
| Dynamic Interaction |
| Control HTML properties |
| Control CSS properties |

The P300 Speller developed is intended to be used offline without the need for an active internet connection therefore it does not involve server-side programming languages.

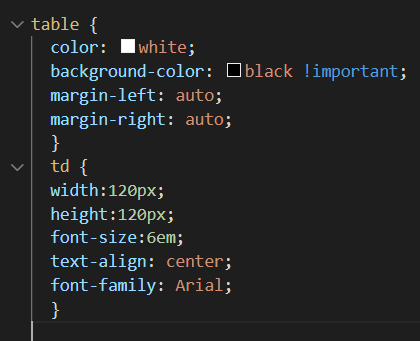
The time-static elements of the P300 speller such as background colour and button positions can be realized using HTML and CSS. However the P300 Speller also has time-dynamic elements such as element colour changes and timing delays. The only client-side language capable of expressing time-dynamic behaviour is JS.(Jquery also used).

**PROGRAM IMPLEMENTATION**

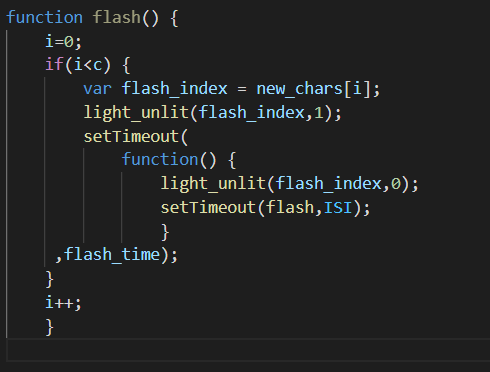
****

This is the HTML code for the first row of Speller element. Subsequently all rows of the matrix are realized in HTML this way.

Each element of the P300 Speller matrix was given a unique id label in HTML. This was done to enable easy targeting by the CSS for character flashing. The CSS code snippet that defines the margins and background/foreground color for the P300 element.



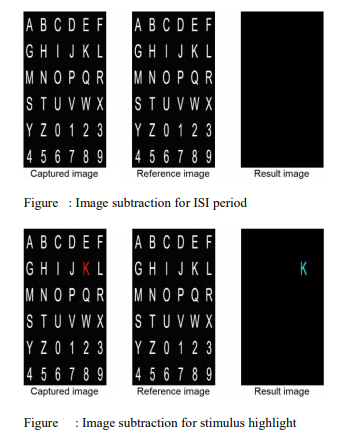
The JS code snippet is responsible for the time dynamic aspects of the P300 Speller Paradigm.



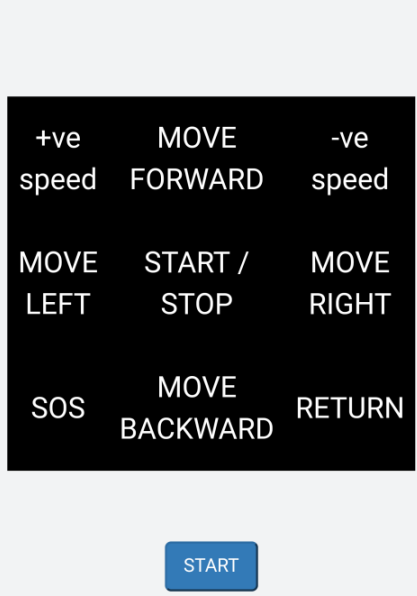
The ‘new\_chars’ variable with length ‘c’ is defined outside of the function and contains the total list of characters to be flashed based on the number of trials. The “light\_unlit” function flashes a P300 matrix element if the second function input is ‘1’ and reverts the element base colour to white if the second input is ‘0’.The “setTimeout” function waits for a certain time before running some specified code. Recursion is then employed to ensure that the code progresses to highlight and then revert character colours until the total character set is flashed.

**RESULTS:**

The developed paradigm works well on google chrome browser windows, internet explorer and mozilla firefox.The developed program accurately implements the stimulus sequencing required of the P300 Speller. However, stimulus flash times and ISIs were obtained to determine compliance with the stimulus timing requirements. The PC monitor was therefore recorded using a software screen recorder which ran concurrently with the paradigm. The recording was done at 60 frames per second (fps) which coincides with the screen refresh rate.



Above figure presents examples of the image processing that was done to determine where a stimulus was flashing or base colour.



A mobile version of P300 Speller can also be developed. It is a modified P300 Speller with a smaller command matrix intended for better viewing on a mobile platform. The commands are tailored for the control of a vehicular platform which is a common P300-based BCI application

**DISCUSSION:**

This work investigates the implementation of the P300 Speller using web development languages. The developed paradigm is executable on both laptop and Android variants of the Google Chrome browser. The paradigm was captured and analysed to determine the compliance of the programmed timing delays.

* The results revealed that the actual flash times and ISIs deviated at most 5.17ms from the programmed time. The successful implementation of the P300 Speller using web development languages provides a useful proof of concept for BCI implementation in a web browser.
* There are also benefits to coding the P300 Speller in web development languages such as HTML/CSS/JS instead of device standard languages such as C++ and Java. Web pages can be executed on all modern devices with a web browser.
* This allows the benefit of a single program which can run on every operating system that implements a web browser without considerations for library and hardware constraints.

# **Conclusion:**

This report investigates the implementation of the P300 Speller Paradigm using web development languages(HTML, CSS, JavaScript). The results demonstrate that the developed paradigm complied with the timing delays required of the P300 Speller. The paradigm is executable on both a laptop and Android smartphone. This paper therefore provided an important proof of concept for web browser based BCI paradigm presentation.

**FUTURE ASPECT-** The developed paradigm must be integrated to data collecting and signal processing elements to allow for a full BCI implementation that included training and testing sessions. User interface can be improved to allow for selectable parameters. This can be further expanded to allow for a testbed based approach.

# **Reference:**

1.Article-A Novel P300 Classification Algorithm Based on a Principal Component Analysis-Convolutional Neural Network

2.P300 SPELLER IMPLEMENTATION USING WEB DEVELOPMENT LANGUAGES R.E.S. Harnarinesingh1 and C.S. Syan2 1 Department of Electrical and Computer Engineering, The University of the West Indies, St. Augustine, Trinidad 2 Department of Mechanical and Manufacturing Engineering, The University of the West Indies, St. Augustine, Trinidad