# Documentation
# Wadsworth BCI Dataset (P300 Evoked Potentials)
**Data Acquired Using BCI2000's P3 Speller Paradigm**
(http://www.bci2000.org)

## BCI Competition III Challenge 2004

Organizer: Benjamin Blankertz (benjamin.blankertz@first.fraunhofer.de)

**Contact:**
Dean Krusienski (dkrusien@wadsworth.org; 518-473-4683)
Gerwin Schalk (schalk@wadsworth.org; 518-486-2559)

## Summary

This dataset represents a complete record of P300 evoked potentials recorded with BCI2000[1] using a paradigm described by Donchin et al., 2000, and originally by Farwell and Donchin, 1988. In these experiments, a user focused on one out of 36 different characters. The objective in this contest is to predict the correct character in each of the provided character selection epochs.

## The Paradigm

The user was presented with a 6 by 6 matrix of characters (see Figure 1). The user's task was to focus attention on characters in a word that was prescribed by the investigator (i.e., one character at a time). All rows and columns of this matrix were successively and randomly intensified at a rate of 5.7Hz. Two out of 12 intensifications of rows or columns contained the desired character (i.e., one particular row and one particular column). The responses evoked by these infrequent stimuli (i.e., the 2 out of 12 stimuli that did contain the desired character) are different from those evoked by the stimuli that did not contain the desired character and they are similar to the P300 responses previously reported (Farwell and Donchin, 1988, Donchin et al., 2000).

[1] BCI2000 is a flexible Brain-Computer Interface research and development platform. It supports a variety of brain signals, signal processing methods, and user applications, and is available free of charge for research purposes (http://www.bci2000.org). It is currently used by 30 research groups for a variety of studies.

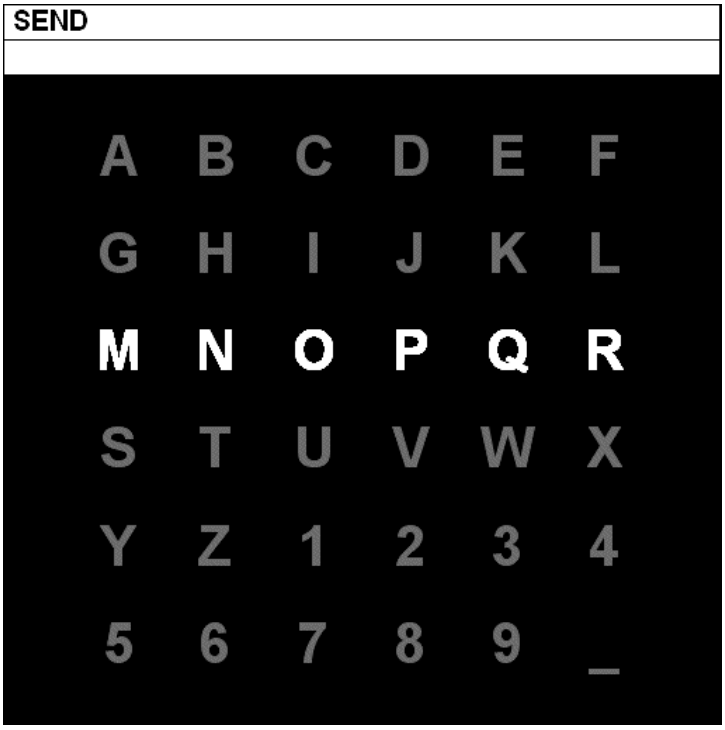


**Figure 1:** This figure illustrates the user display for this paradigm. In this example, the user's task is to spell the word "SEND" (one character at a time). For each character, all rows and columns in the matrix were intensified a number of times (e.g., the third row in this example) as described in the text.

## Data Collection

We collected signals (bandpass filtered from 0.1-60Hz and digitized at 240Hz) from two subjects in five sessions each. Each session consisted of a number of runs. In each run, the subject focused attention on a series of characters. For each character epoch in the run, user display was as follows: the matrix was displayed for a 2.5 s period, and during this time each character had the same intensity (i.e., the matrix was blank). Subsequently, each row and column in the matrix was randomly intensified for 100ms (i.e., resulting in 12 different stimuli – 6 rows and 6 columns). After intensification of a row/column, the matrix was blank for 75ms. Row/column intensifications were block randomized in blocks of 12. The sets of 12 intensifications were repeated 15 times for each character epoch (i.e., any specific row/column was intensified 15 times and thus there were 180 total intensifications for each character epoch). Each character epoch was followed by a 2.5 s period, and during this time the matrix was blank. This period informed the user that this character was completed and to focus on the next character in the word that was displayed on the top of the screen (the current character was shown in parentheses).

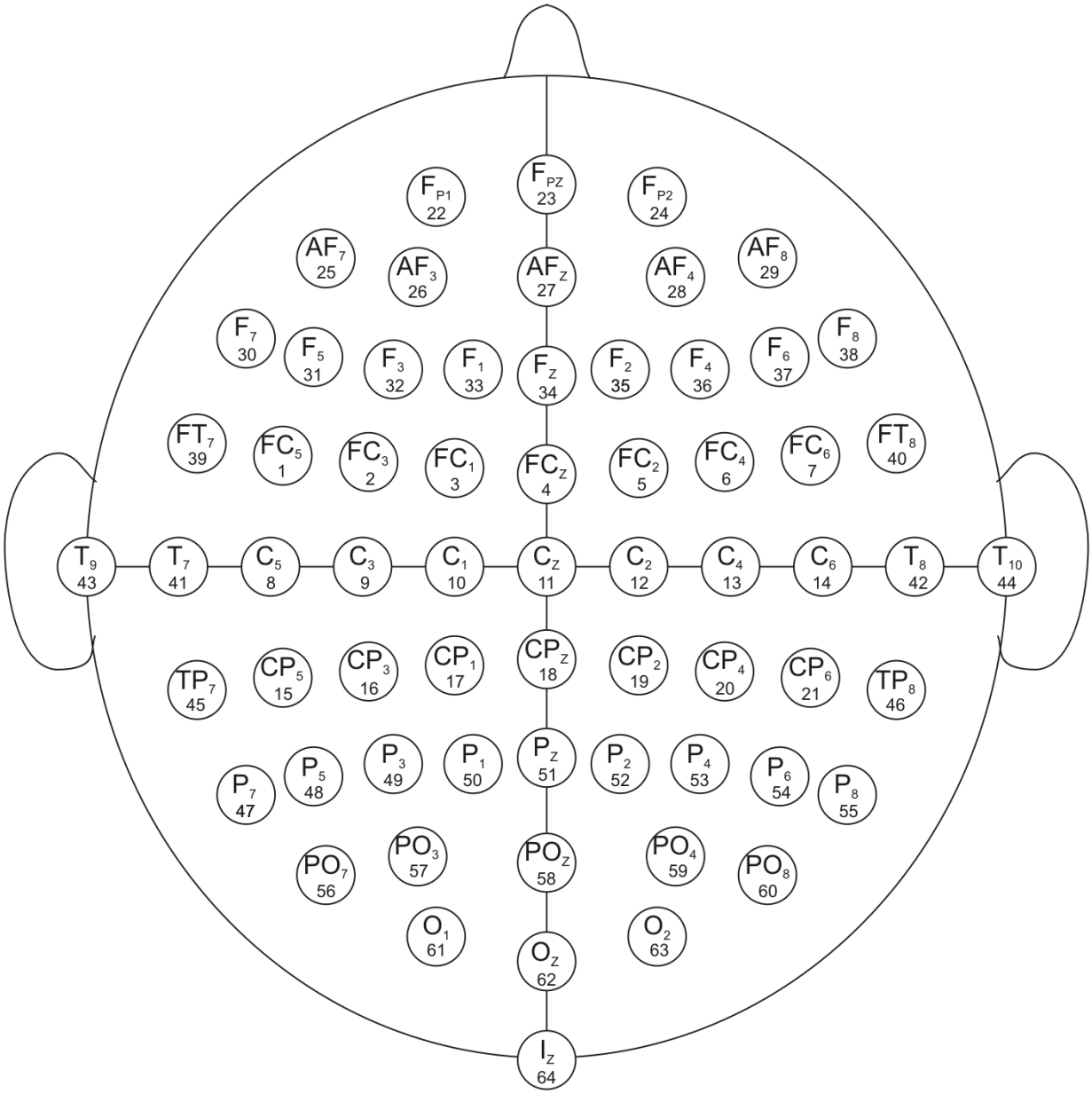


**Figure 2:** This diagram illustrates electrode designations (Sharbrough, 1991) and channel assignment numbers as used in our experiments.

## The Competition Data Set

For the competition data set the recorded data has been converted into 4 Matlab *.mat files, one training (85 characters) and one test (100 characters) for each of the two subjects A and B. All of the data is stored in *single precision* and may need to be converted to double precision (using the *double* command) depending on the version of Matlab used. For each *.mat file, the recorded 64 channel EEG signal is organized in one big matrix (*Signal*) as illustrated in Figure 3. Note that, because the subjects' task was to spell actual words in each run, the character epochs have been scrambled in the training and test sets to prevent identification of the correct test set characters by the participants. The other relevant variables included are described below. The structure of each variable is given in Figure 3.

For each sample in the *Signal* matrix, associated events are coded using the following variables:

| | |
|---|---|
| *Flashing*: | 1 when row/column was intensified, 0 otherwise |
| *StimulusCode*: | 0 when no row/column is being intensified (i.e., matrix is blank)<br>1…6 for intensified columns (1 … left-most column)<br>7…12 for intensified rows (7 … upper-most row)<br>See Figure 4 for details. |
| *StimulusType*: | 0 when no row/column is being intensified or intensified row/column does not contain desired character<br>1 when intensified row/character does contain the desired character<br>This variable provides an easy access to the labels in the training sets in that it can be used to separate the responses that did contain the desired character from the ones that did not. (Obviously, this could also be done using the variable *StimulusCode* in conjunction with the *TargetChar* that the user focused on.) |
| *TargetChar*: | The correct character label for each character epoch in the training data. |

It only takes a few steps to extract the signal waveforms associated with the intensification of a particular row/column:

- For one or more channels, collect a period of signal samples at the start of each intensification, i.e., whenever *Flashing* changes from 0 to 1 (note: each character epoch of the data set starts at the first flash, i.e. *Flashing=1* for the first data sample in each epoch).
- Accumulate the signal samples in 12 separate buffers, according to the *StimulusCode* of the corresponding stimulus. For each character epoch, each buffer should contain the 15 sample periods – one for each intensification of the given row/column. Each character in the matrix is represented by the row/column intersection as illustrated in Figure 4.

| Variable | Dimension 1 | Dimension 2 | Dimension 3 |
|---|---|---|---|
| ***Signal:*** | **Character Epoch** | **X Samples** | **X Channels** |
| ***Flashing:*** | **Character Epoch** | **X Samples** | |
| ***StimulusCode:*** | **Character Epoch** | **X Samples** | |
| ***StimulusType:*** | **Character Epoch** | **X Samples** | |
| ***TargetChar:*** | **Character Epoch** | **X Samples** | |

**Figure 3:** This figure illustrates the content of each Matlab file. (The test data set does not contain *StimulusType* and *TargetChar*.) Channel numbers in the variable *Signal* correspond to numbers in Figure 2.

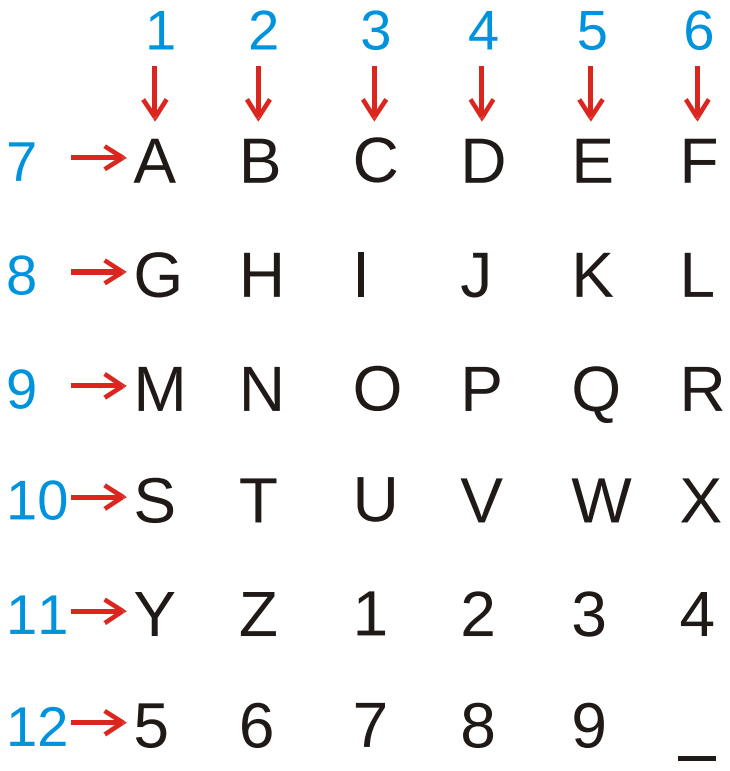


**Figure 4:** This figure illustrates the assignment of the variable *StimulusCode* to different row/column intensifications.

## Demonstration Code

- example.m
  This program extracts the stimuli from channel Cz and uses a very simplistic "peak picking" algorithm to predict the target for each character epoch in the training data for Subject A. This example uses the average of all 15 intensifications to perform the classification. To execute the code, change the Matlab directory to the directory containing the files and type "example" in the command window.

## Demonstration Analyses

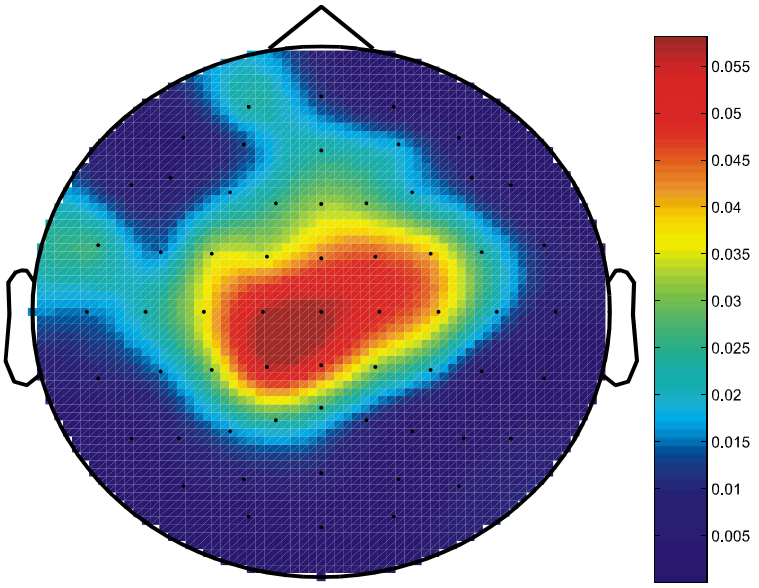


**Figure 5:** This figure shows a topography of values of $r^2$ (i.e., the proportion of the signal variance that was due to whether the row/column did or did not contain the desired character), calculated for one sample at 310ms after stimulus presentation. This topography shows that there is a spatially fairly wide-spread difference at 310ms after intensification of a row/column that is different for rows/columns that did vs. ones that did not contain the desired character.

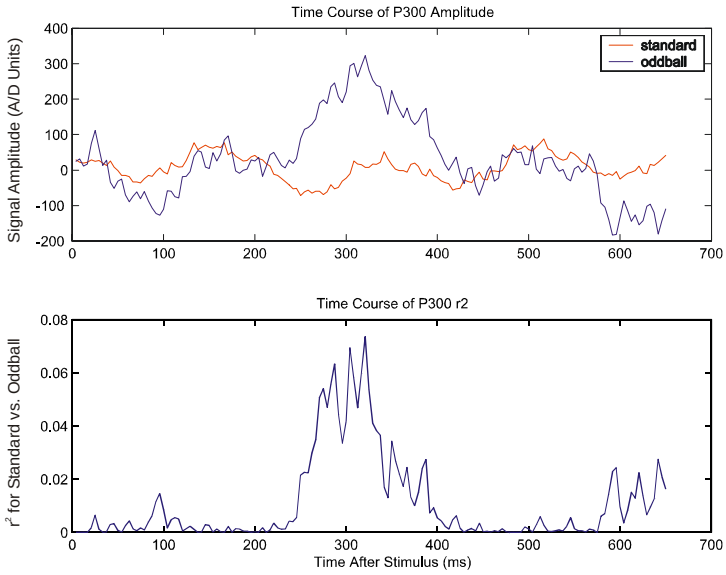


**Figure 6:** This figure shows an example time course of average signal waveforms (at Cz) and of $r^2$ (i.e., the proportion of the signal variance that was due to whether the row/column did (oddball) or did not contain the desired character (standard)).

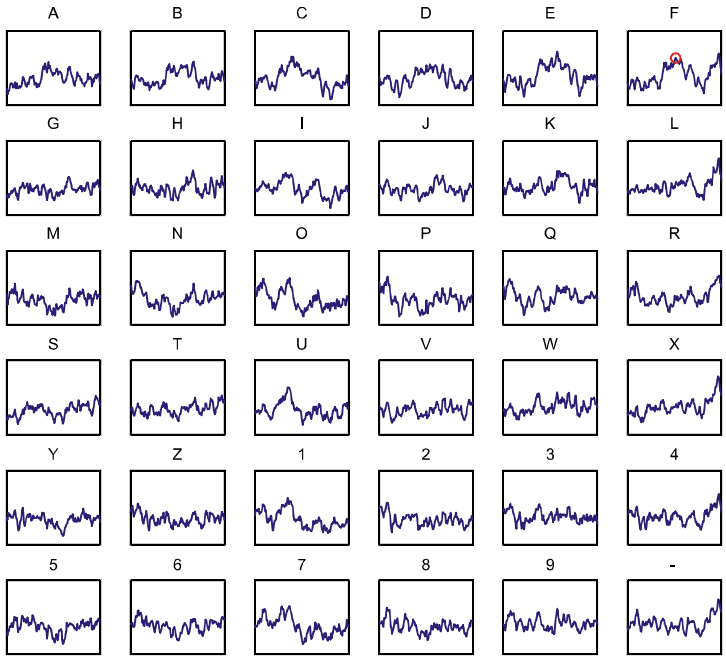


**Figure 7:** This figure shows averaged responses for each character (each character response is calculated as the average of the corresponding average row and column responses), and the character that was predicted for these data using a very simple classification procedure.

## The Goal in the Contest

The goal in this competition is to use the labeled training data (i.e., files Subject_A_Train.mat and Subject_B_Train.mat for subject A and B, respectively) to predict the character sequences in the test set (i.e., files Subject_A_Test.mat and Subject_B_Test.mat for subject A and B, respectively). The deliverables are a total of four character vectors (each 100 elements long; one element for each character epoch in the test data set), two for each subject. The first character vector for each subject shall contain the classification of the test data using all 15 intensifications of each row/column to perform the classification. The second character vector shall be in the same format and contain the classification of the test data using only the first 5 intensifications of each row and column (i.e., a total of 12(stimuli)*5(presentations)=60 intensifications) to perform the classification. The latter results will only be used in the case of a tie. The format for the files to submit is as follows:

One Matlab file 'results.mat' that contains the four character vectors of the predicted characters (2 for each subject and all characters in upper case) in the same format as the variable *TargetChar*. The names for each variable within 'results.mat' shall be as follows:

**SA15** – resulting classification character vector for all 15 intensifications of Subject_A's test set.
**SB15** – resulting classification character vector for all 15 intensifications of Subject_B's test set.
**SA5** – resulting classification character vector for the first 5 intensifications of Subject_A's test set.
**SB5** – resulting classification character vector for the first 5 intensifications of Subject_B's test set.

We will compare the submitted characters to the actual target characters to determine % correct. The submission with the highest % correct for the 15 intensification results wins the competition. In the case of a tie, the submission with the highest % correct for the 5 intensification results will win the competition. *For comparison, the simplistic "peak picking" algorithm implemented in the example.m program gives a percent correct of between 20-40% for the test data sets.*

## Bibliography

Farwell L.A., Donchin, E. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. Electroencephalography & Clinical Neurophysiology. 70(6):510-23, 1988.

Donchin, E., Spencer, K.M., Wijensinghe, R. The mental prosthesis: Assessing the speed of a P300-based brain-computer interface. *IEEE Trans. Rehab. Eng*. 8:174-179, 2000.