

UMA-BCI Speller: an Easily Configurable P300 Speller Tool for End Users

Francisco Velasco-Álvarez¹, Salvador Sancha-Ros², Esther García-Garaluz², Álvaro Fernández-Rodríguez¹, M. Teresa Medina-Juliá¹, Ricardo Ron-Angevin^{1*}

¹ Departamento de Tecnología Electrónica, Universidad de Málaga, 29071 Málaga, Spain

² ENESO Tecnología de Adaptación S.L., Parque Tecnológico de Andalucía, Málaga, Spain

* Corresponding author: rra@dte.uma.es

Abstract

Background and Objective. Some neurodegenerative conditions can severely limit patients' capability to communicate because of the loss of muscular control. Brain-computer interfaces may help in the restoration of communication with these patients, bypassing the muscular activity, so that brain signals can be directly interpreted by a computer. There are many studies regarding brain-controlled spellers; however, these systems do not usually leap out of the lab because of technical and economic requirements. As a consequence, the potential end users do not benefit from these scientific advances in their daily life. The objective of this paper is to present a novel brain-controlled speller designed to be used by patients due to its versatility and ease of use.

Methods. The brain-computer interface research group of the University of Málaga (UMA-BCI) has developed a speller application based on the well-known P300 potential which can be easily installed, configured and used. The application supports the common P300 paradigms: the Row-Column Paradigm and the Rapid Serial Visual Presentation Paradigm. The inner core of the application is implemented with a widely used and studied platform, BCI2000, which ensures its reliability and allows other researchers to apply modifications at will in order to test new features. Ten naïve volunteers carried out exercises using the application and completed usability tests for evaluation purposes.

Results. New subjects using the application managed to set up and use the proposed speller in less than an hour. The positive results of the evaluation through the usability tests support this application's ease of use.

Conclusions. A new brain-controlled spelling tool has been presented whose aim is to be used by severely paralyzed patients in their daily lives, as well as by researchers to test new spelling features.

Keywords: Brain-computer interfaces, P300 speller, Graphical user interface, BCI2000, Usability

1.Introduction

1.1. Brain–computer interfaces

Human communication is based on messages that the sender emits to the receiver. The nature of these messages is quite broad; senders can emit spoken words or sounds (that the receiver will hear), written text or gestures (to be seen by the receiver) or even messages based on physical contact (that the receiver will feel). One point in common to the mentioned messages is that all of them need the sender to perform some kind of muscular movement (in order to speak, write or make a gesture).

On the other hand, there are several conditions that can limit patients' ability to perform movements, and therefore their possibility to communicate, while their cognitive capabilities remain intact. In such cases, patients enter the so-called locked-in state (LIS), as they may be conscious but they do not have the capability to express any thought due to the muscular impairment. When the movement limitation is not absolute, i.e., when patients retain some residual muscular capability (such as gaze control), they can use systems that restore the communication based on the mentioned residual movement (e.g. using an eye tracker, blow sensors or tongue press pad systems that help them to spell words [1]). There are some neurodegenerative diseases, such as amyotrophic lateral sclerosis (ALS), that in advanced stages can even limit the residual eye movements, thus leaving the patients in a state in which they lose all muscular control; in these cases the patients enter the complete locked-in state (CLIS). However, even when no muscular movement is possible, there are alternative ways of establishing communication [2]: systems that decode the senders' messages directly from their mental activity, which are called brain–computer interfaces (BCI).

BCI systems usually analyze the electroencephalographic (EEG) signals in the scalp in order to differentiate some simple mental activities. Several kinds of EEG signals can be detected, resulting in different kinds of BCI [3]. There are BCIs that process the generated EEG resulting from voluntary thoughts, such as the sensorimotor rhythms (SMR) or slow cortical potentials (SCP). Other BCIs depend on external stimuli in order to elicit a brain response, such as: i) the P300 signal, a positive peak that appears in the EEG approximately 300 ms after the expected presentation of a rare stimulus; and ii) the steady-state visual evoked potentials (SSVEP), changes in the neural activity located at the visual cortex that occur at the same frequency as a blinking stimulus.

The common implementation of a BCI is shown in Figure 1, distinguishing three main parts:

- Signal acquisition. This part includes the hardware that is necessary to register the brain signals, such as electrodes, amplifiers and data acquisition devices.
- Signal processing. This includes the algorithms that take the signals as inputs (feature extraction) and deliver as outputs one among a set of classes that are already known to the system (classification).
- Control. After the results of the classification, the system interacts with the environment in some way or simply gives the subjects some feedback in return that allows them to assess their control over the system.

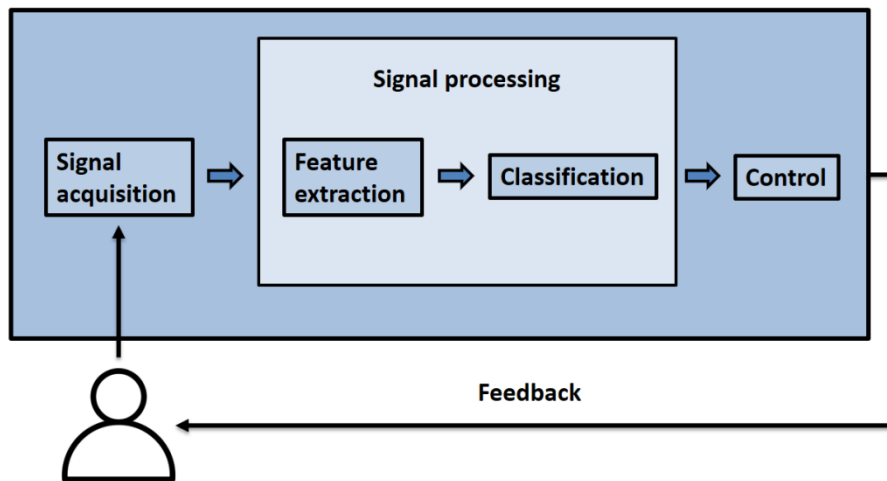


Figure 1. General BCI scheme. The input signal is acquired from the users brain, this signal is processed and the control module provides the user with the corresponding feedback

The usual applications of BCI systems include the control of spelling devices [4], prostheses [5], adapted wheelchairs [6], smart homes [7] or even web browsers [8] and videogames [9].

1.2. P300-based speller

One of the most important applications of BCI systems is to restore the communication of severely paralyzed patients. The review in [4] shows a complete survey on BCI applications focused on spelling devices. Several signals are used as inputs for BCI spellers, mainly P300, SSVEP, SMR and hybrid systems based on more than one input signal. The two main parameters when describing a speller are the accuracy and the information transfer rate (ITR). The accuracy of spellers based on different input signals remains above 80% in most cases, independently of the signal used [4]. The systems that achieve the highest ITR are those based on SSVEP (one group reported an average ITR of 325.33 bits/min using a combination of frequency and phase variations on their targets [10]). However, SSVEP systems need adequate gaze control and have been reported to be tiring and annoying for users and can possibly trigger epileptic seizures [11]. The most popular signal is still the P300 evoked potential, as the first BCI speller was based on it [12] and it has been studied for many years, with researchers proposing many variations to the paradigm. Besides, the P300 system can be controlled with little training and it offers a notably high ITR (a study using a P300 paradigm with two simultaneous stimuli achieved 80 bits/min [13]). This particular signal is a well-known reaction located in the parietal lobe due to expected and rare stimuli [14]. Usually, an oddball paradigm with multiple options is presented to the user, who has to pay close attention to one of these options. Every option keeps being stimulated pseudo-randomly, in such a way that when the option that the user is focused on is highlighted, the P300 potential is elicited in the brain and detected by the BCI system (Figure 2), so that the system determines which option the user wanted to select. Even when the visual interface is the most used in P300 systems, this evoked potential can be elicited by different stimulations as well (e.g. auditory or tactile) in order to make P300 gaze-independent [15].

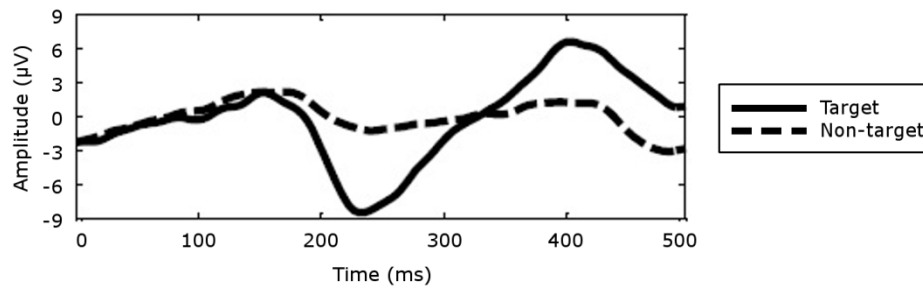


Figure 2. Target and non-target waveforms. The signal corresponding to target stimuli presents a differentiable P300 effect

This concept was adopted by Farwell and Donchin [12] to propose a paradigm to control a text speller (Figure 3). In their proposal, the oddball paradigm consists of the random stimulation of the rows and columns of a matrix of characters (the stimulation affects one row or one column each time). In this way, when the system completes a sequence of stimulations (i.e., all rows and all columns have been stimulated), every character has been highlighted twice. Now the system analyzes the users' EEG to detect what column and row stimulation elicited a P300 response, which defines the selection of a single character. Commonly, several stimulation sequences are needed to correctly detect the desired symbol.

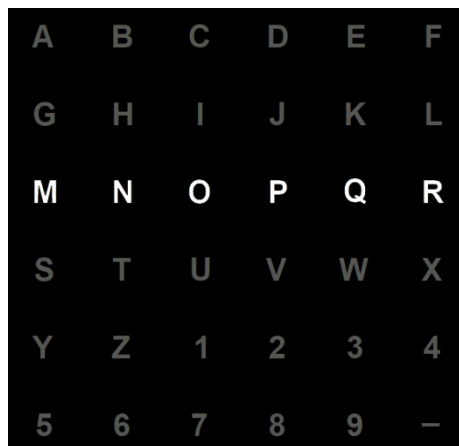


Figure 3. Common P300 speller matrix. Rows and columns are randomly highlighted while the users focus their attention in one specific symbol

The most common use of this paradigm is to spell text as in the original proposal, but it can be configured to contain other elements than characters, in order to be used for selecting different options, such as functions of a domotic system [16] or positions to move a robotic wheelchair [17].

The use of a P300 speller BCI generally consists of three phases:

1) Implementation. The operator must decide and set up a matrix with a certain number of rows and columns containing the elements to be selected. It is necessary to consider that a bigger matrix makes it possible to select a larger number of elements, but it takes longer to select a particular item, as the sequence to highlight all the rows and columns has more elements. Another important aspect to consider is the timing of the interaction.

The operator fixes a temporization that determines how long the highlighting lasts and how long is the pause between stimulations.

2) Calibration or training. Before the BCI can recognize any pattern in the user EEG, it is necessary to train the system in a calibration phase in which the EEG is recorded while the user performs a specific task. This task consists of a series of prefixed items to which users must pay attention in a particular order; a commonly used strategy is to ask the subjects to mentally count how many times that item is highlighted. In this way, the recorded EEG can be analyzed afterwards as corresponding to a target or non-target item. After recording the EEG, an offline analysis of it determines a classifier which enables online communication. As a result of this analysis, an optimum number of sequences is obtained as well, so the number of stimulations that the users count in order to select an item can be reduced in the third phase.

3) Online test. Finally, the online spelling tasks can be performed using the classifier obtained in the previous stage. In the online test, the users themselves choose what item they want to select, they pay attention to it (again, usually counting the number of times it is stimulated) and, as a consequence, the desired item is selected. Then, the users can continue to select a new item and so on in order to spell a text.

1.2.1. Rapid serial visual presentation paradigm

Up to this point, we have described the most commonly used P300 speller paradigm, which is based on the stimulation of the rows and columns of a matrix of characters. This is usually called the row-column paradigm (RCP). However, in this paradigm the symbols are placed in different positions on a screen, so the users need to have a certain gaze control in order to pay attention to one of these symbols and select it. In other words, the performance of the RCP speller depends on gaze control [18]. An alternative P300 paradigm that does not depend on the gaze direction consists of showing the symbols one by one in the centre of the screen. So, instead of paying attention to a present symbol of the matrix and mentally counting how many times it is stimulated, with this alternative the users pay attention to the centre of a screen where different symbols keep appearing, in such a way that they should mentally count how many times the desired symbol appears. This paradigm is called the rapid serial visual presentation (RSVP), as shown in Figure 4.

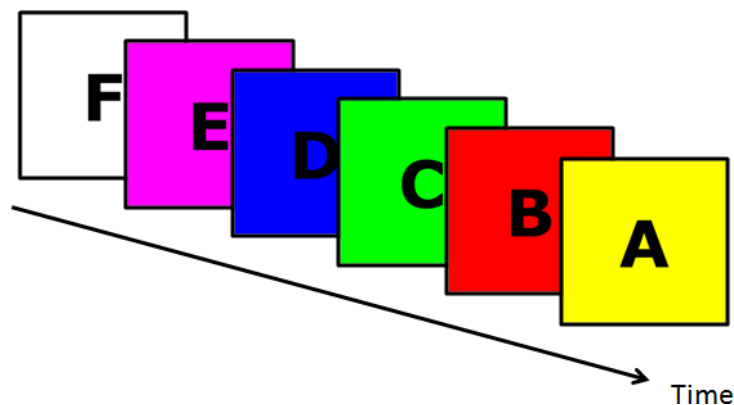


Figure 4. Rapid serial visual presentation (RSVP) paradigm example. The stimuli are presented in a random sequence, with only one stimulus at a time

As explained in the review of BCI spellers in [4], this kind of paradigm has the advantage of being gaze-independent, but the information transfer rate (ITR) that can be achieved is lower than in the case of the RCP. The slower speed of the RSVP is caused by the fact that symbols need to be presented serially, while in the RCP several symbols are stimulated simultaneously (in the same row or column).

This paradigm is easily adapted to be audio-visual, by adding different sounds to each symbol presentation, and to be only auditory, i.e., the rapid serial auditory presentation (RSAP) paradigm. Through this last proposal, users do not need gaze control, and do not even need to look at the screen.

1.3. Objective of this study

Most of the studies regarding BCI research are carried out in experimental environments with healthy subjects. Even when patients suffering from different kinds of paralysis participate as subjects, they usually do it only for a few sessions in order to validate the proposed BCI systems. Several authors have also studied the use of BCIs taking into account the perspective of patients, caregivers, and professionals:

- A recent study by Wolpaw et al. [19] found that, although there are more than 4000 research studies regarding BCI systems, only three cases corresponded to independent home use of a BCI for communication. Encouraged by these case reports, they carried out research with 37 ALS patients who initially began to use a BCI system at home; when the study ended (after up to 18 months), only 7 patients kept the BCI for further use. The authors mention several reasons that explain this difference: 13 patients died during the study or abandoned it because of a rapid disease progression, 9 patients could not use it, 4 lost interest and 2 preferred another assistive communication device, among other minor reasons.
- The results of Taherian [20] showed that BCIs are not suitable for independent use outside experimental environments. For them, "the hardware needs to be configurable, comfortable and accommodate physical support needs. The training approach needs to be less cognitively demanding, motivating and support personalized mental tasks". They also cite how the reliability should improve and the need for adequate technological support.
- Liberati et al. [21] concluded that four reasons were fundamental to understand the low intensity use of BCI: "i) lack of information on BCI and its everyday applications; ii) importance of a customizable system that supports individuals throughout the various stages of the disease; iii) relationship between affectivity and technology use; and iv) importance of individuals retaining a sense of agency".

Taking into account these results, the BCI research group of the University of Málaga (UMA-BCI) decided to implement a BCI system based on a P300 speller that helped to solve some of the previously mentioned issues in order to be used by patients and caregivers at home. It should be an easy to install and use tool and it should be modular and configurable to support personalization. At the same time, the developed tool should be flexible enough to be used by researchers in the field in order to study variations and alternative paradigms. Both RCP and RSVP P300-based paradigms should be supported. The target tool should include the possibility of providing dynamic

spellers, i.e., spellers that change the items in their layout depending on previous selections. One last specification of the speller is that it should be free and open source software.

Three options were available: i) to fully develop a custom BCI system; ii) to use a general purpose platform to implement the desired speller; and iii) to partially develop a custom BCI system based on a BCI platform. We declined the first option because it would require a lot of development and testing in order to get a reliable application. The second option would avoid these processes, but it would be limited in its functionality and it would require advanced technical skills, as the reader will see in the next section. Finally, we decided to benefit from the extended use of platforms to use reliable signal acquisition and signal processing modules, thus using a general purpose platform as the internal core of a custom application. The next section addresses this subject.

1.4. Popular BCI platforms

BCI systems have been the focus of attention from many research groups for more than three decades so, as expected, several multi-purpose platforms for BCI analysis and development are available for scientists to carry out their research. A complete survey on these platforms can be found in [22]. As the present paper is focused on the development of a P300 speller, we will next present a brief description (extracted from the mentioned survey) of some platforms that could be used to implement the speller. This section ends with a discussion of the pros and cons of each platform related with the objective of this research.

1.4.1. BCI2000

BCI2000 [23] is a general-purpose free software platform for BCI research. It provides support for different data acquisition hardware, signal processing algorithms and experimental paradigms. It offers highly customizable auditory and visual stimulation which is synchronized with the acquisition of brain signals. Its modular design allows the customization of the system through the adaptation of parameters from an operator GUI (without recompilation). It is composed of reusable software blocks that support the creation of new modules (for example, for a particular data acquisition hardware). The used programming language is C++ but it supports online signal processing code in MATLAB and it includes a layer of Python compatibility. Since its presentation, it has had a great impact on the BCI research field, with hundreds of papers reporting BCI experiments using BCI2000.

1.4.2. OpenViBE

OpenViBE [24] is a general-purpose software platform for the design and implementation of BCI systems. It is free and open-source. The platform consists of different software modules that can be integrated to develop fully functional BCIs. One of the important characteristics of OpenViBE is that it offers a GUI for non-programmers, consisting of a box-based design concept (a box is a graphical representation of a functional module). Boxes can be created and connected in order to develop a complete BCI system. OpenViBE is designed for different types of users, including researchers, developers, and clinicians. Different tools are available for each

user type, depending on their needs, programming skills and knowledge of brain physiology.

1.4.3. BCILAB

BCILAB [25] is an open source MATLAB-based toolbox for advanced BCI research. The design of BCILAB is focused less on clinical or commercial deployment, but on the advancement of the BCI technology and methodology itself. However, compiled versions of BCILAB are available to run standalone versions of BCI methods.

1.4.4. BCI++

BCI++ [26] is an open source framework based on a sophisticated graphics engine. The platform provides a set of tools for the rapid development of BCIs and human-computer interaction (HCI) in general. The aim of this framework is to allow developers to focus on the design of the HCI using the advanced graphics engine. The framework is very flexible, and the large set of debugging tools simplifies the debugging and testing of new systems. The most relevant aspect of BCI++ is the possibility for unskilled developers to develop and test their own work and to actively help to increase the number of available instruments in the framework. However, it provides little support for third party hardware and it has had a limited impact on BCI research groups.

1.4.5. BF++

BF++ [27] is a framework whose aim is to provide tools for the implementation, modelling and data analysis of BCI and HCI systems. It is highly scalable, cross-platform, and programmed by adopting only well-established technologies, such as C++ as the programming language, XML for storage, and UML (unified modelling language) for description and documentation. It provides a collection of small applications specifically designed for the analysis, evaluation, simulation and optimization of BCIs.

1.4.6. Summary of BCI platforms

It is worth mentioning that other BCI tools exist that help with the processing of EEG signals but that do not meet the needs of a complete BCI system. This is the case of EEGLAB [28] (integrated with BCILAB), BioSig [29] (whose real-time features need MATLAB and Simulink) or xBCI [30] (deprecated). On the other hand, the Pyff platform [31] does not support data acquisition or signal processing, but it is focused on the feedback and stimulus presentation.

As previously mentioned, the aim of this work is to provide end users with an easy to use P300 speller that is flexible enough to be useful for researchers as well. From this point of view, BCI2000 and OpenViBE are the most interesting among the mentioned BCI platforms as they are both widely used platforms, with up-to-date software releases, documentation and support. Regarding the other mentioned platforms, they do not seem appropriate for different reasons: i) BCILAB is focused on the development and testing of new BCI methods and it requires licensed software (MATLAB); ii) BCI++ has not enough support of data acquisition hardware, iii) BF++ is mostly oriented towards BCI evaluation and not on efficient systems development.

Even though BCI2000 and OpenViBE are intended to build end user BCI applications, they still require technical skills in order to implement a P300 speller. This may be the reason why these platforms are not present enough in end user scenarios, for example for clinical uses. As both are general-purpose platforms, with a high degree of configurability, it is complex to parameterize them in order to obtain the desired speller. Besides, the tool proposed in the present paper is intended to be easily configurable for researchers as well, in such a way that the testing of new spelling modalities using alternative stimuli may help in their research (e.g. through the use of images or the independent customization of each cell). BCI2000 is used notably more by researchers in their works [22], so a speller based on it could be easier to use and modify by the scientific community than one based on OpenViBE.

As explained in [4], many publications have recently explored new possibilities in order to improve the speller's performance (or the ERP amplitude). The proposed variations tend to change the way the stimuli are presented to the user: by changing colours, showing images or altering the classic row-column paradigm. If we focus on variations of the classic P300 matrix speller [13,32–53], 8 out of 23 papers used BCI2000 for data collection and signal processing, but only 6 used this platform for stimulus presentation. This fact points to the idea that alternative ways of configuring the speller are not easily done with BCI2000. As mentioned before, this limitation in the layout variation possibilities was the main reason for discarding the implementation of the desired speller using a general purpose platform as it is.

To sum up: the objective of the UMA-BCI group was to provide an easily configurable speller for end users, but it should offer multiple configuration options in order to be useful also for BCI researchers, who should be able to modify the system at will. The developed system should not depend on licensed software and it should be compatible with multiple data acquisition hardware. With these conditions, the UMA-BCI group decided to develop a custom application based on BCI2000 as the software core of the speller implementation.

1.5. BCI2000

The most widely used BCI platform is BCI2000 [23], a general-purpose software platform for BCI research with GPL license. This platform allows a system to be implemented using three modules: signal source, signal processing and application, which correspond to the main parts of a BCI, as mentioned. Each one of these modules can be independently configured in order to adjust them to the desired system. The general scheme of a BCI2000 implementation is shown in Figure 5. The user can modify many parameters (more than 150) in order to configure the three modules:

- The signal source module supports multiple data acquisition hardware that can be selected. Besides, manufacturers can create their own C++ modules to be included.
- The signal processing module allows the processing of the most common EEG signals (P300, steady-state visual evoked potential (SSVEP), sensorimotor rhythms (SMR), slow cortical potentials (SCP) etc.) with predefined or custom-implemented algorithms.
- The application module contains several basic graphical applications which use the data coming from signal processing as input.

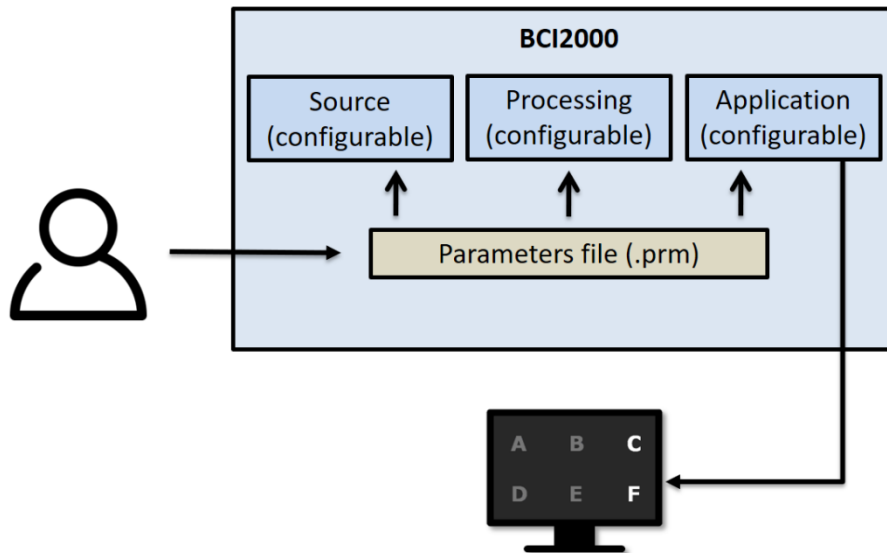


Figure 5. BCI2000 general scheme. The three platform modules correspond to the three main parts of a BCI system: Signal acquisition, Signal processing and Control. The user configures each module through a parameters file (.prm)

All these parameters can be modified and saved through a tab-based GUI provided together with BCI2000 and they are stored in a parameters file with the extension "*.prm". The use of parameterization allows a high degree of customization, but at the same time it makes the application harder to configure.

Among the pre-defined BCI2000 applications, the platform includes one implementation of the P300 speller called P3Speller. As the reader will see through the next section, the UMA-BCI Speller here presented is built using this BCI2000 application.

2.Methods

2.1. Description of the application

This paper presents the UMA-BCI Speller, a P300-based speller application that is built using BCI2000, but that simplifies the use of it and extends its functions acting as a wrapper of BCI2000.

It simplifies its use because the user downloads and runs a single installable file that deploys UMA-BCI Speller and BCI2000. However, the presence of BCI2000 is not necessarily visible as it is wrapped inside UMA-BCI Speller. Figure 6 shows the way users interact with UMA-BCI Speller and how it wraps BCI2000.

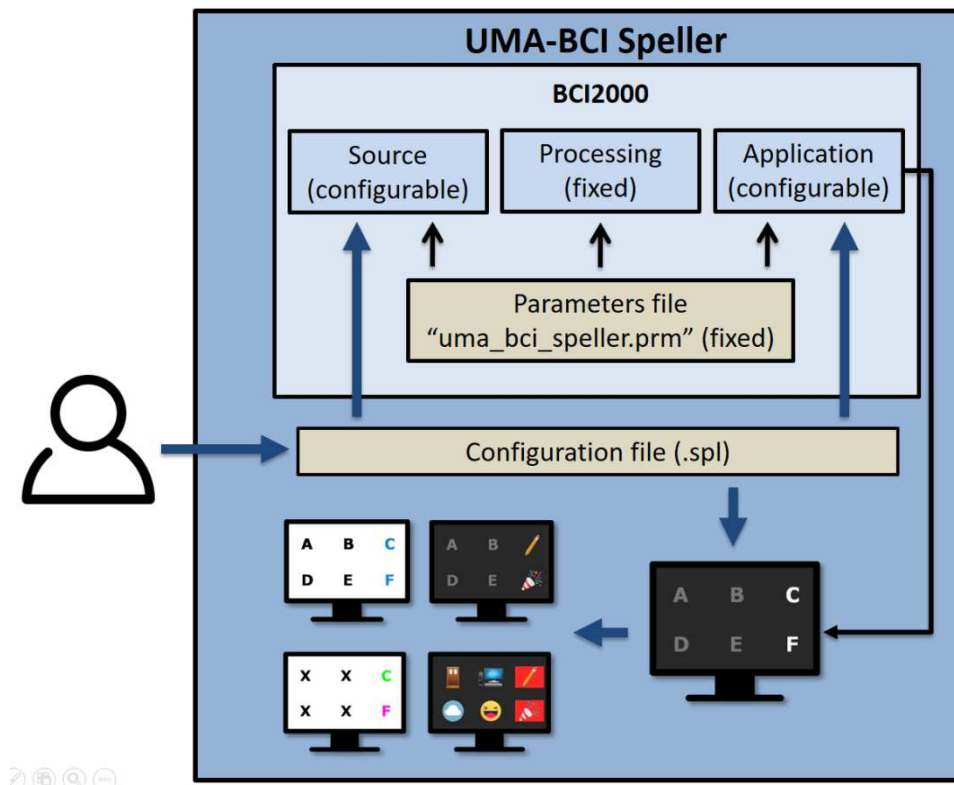


Figure 6. UMA-BCI Speller scheme. The application acts as a wrapper of BCI2000, a fixed parameters files (.prm) configures most part of BCI2000, while a configuration file (.spl) modifies the BCI2000 output to add new features

Most of BCI2000's configurability is intentionally limited because UMA-BCI Speller's aim is to easily implement a P300 speller. For this reason, the BCI2000 signal processing module is fixed to "P3SignalProcessing", as well as the majority of the application module parameters (fixed to be "P3Speller" or "StimulusPresentation" for RCP or RSVP respectively) and the source module (which loads the common electrode positions for a P300 analysis). These predefined settings are applied to BCI2000 through a parameters file ("uma_bci_speller.prm" or "uma_bci_speller_rsvp.prm") that is loaded automatically when users start the application UMA-BCI Speller.

Through the UMA-BCI Speller GUI, users can indirectly modify some BCI2000 parameters that affect the source signal selection and the number of elements of the speller matrix (in Figure 6, the blue arrows pointing up). Besides, users can control the visual appearance of the speller through new options outside BCI2000 that let them change colours and add images (in the same figure, the blue arrow pointing down), as the reader will see next. The set of configuration parameters that users can modify is grouped in a new Configuration file "*.spl".

Thus, users only control a reduced set of parameters in order to use and personalize their own P300 speller keyboard.

Even though BCI2000 is not necessarily visible when using UMA-BCI Speller, it is running in the background in the system. Advanced users (i.e., users who already know how to manage BCI2000) can take advantage of it by opening the mentioned "uma_bci_speller.prm" file and modifying it in order to customize the application beyond what UMA-BCI Speller offers by default (e.g. parameters regarding the timing

of the speller). Another way to modify BCI2000 parameters without altering this file has been included (see section "BCI2000 tab in Settings menu").

2.1.1. UMA-BCI Speller basics

The main advantage of UMA-BCI Speller is the high degree of customization that the speller can present. This customization is achieved through the GUI configuration, which can be modified in two ways: i) by changing the values of the parameters that are related to the general appearance of the speller (number of rows and columns, space in between them, background colour...) and to all the cells equally; and ii) by particularizing the appearance and function of each individual cell. There are two menus to modify these two sets of parameters, the "Grid Parameters" and the "Cell Parameters", which will be described in their respective sections below. The UMA-BCI Speller configuration files (with extension ".spl") save both types of parameters.

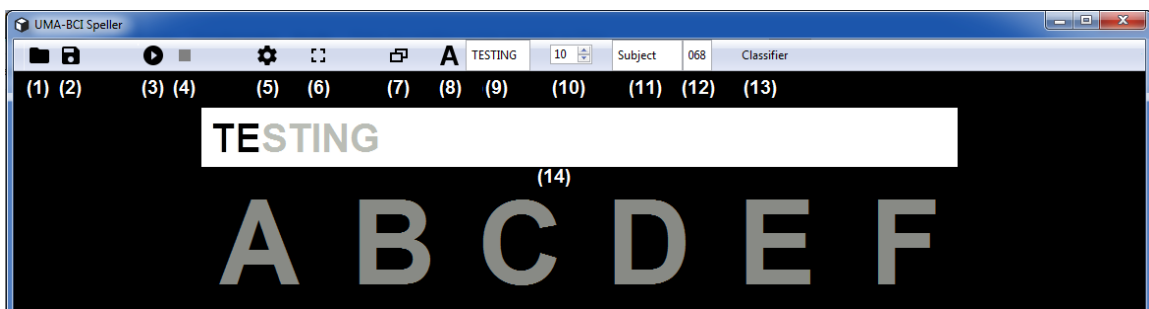


Figure 7. UMA-BCI Speller partial screen capture. The functionality options can be accessed through the items present in the control bar. Numbers in parenthesis are included as references to the main text

Figure 7 shows the top partial view of the main screen; numbers 1 to 14 in parenthesis are included in this Figure below each element of the toolbar as a reference which will be explained next. A helpful pop-up message is displayed with the meaning of each element when the users move the mouse over it.

(1) This button allows a previously saved GUI configuration to be loaded. A configuration determines the speller's final appearance and, as mentioned, it includes two types of parameters (grid and cell parameters).

(2) This button saves the current GUI configuration.

(3) This button starts the application

(4) This button stops the current run.

(5) By clicking this button, the user accesses a Settings menu consisting of two tabs that allow selection of the source signal hardware, a change of language and other minor functions. See "BCI2000 tab in settings menu" section

(6) This button turns on the full screen mode.

(7) This button toggles from the RCP (the button is coloured black) to RSVP (the button turns grey) paradigm and vice versa.

(8) The button marked with an "A" toggles the training or calibration mode (the button is coloured black) to a free spelling mode (the button turns grey) and vice versa.

(9) In this text field the user inputs the text to spell for calibration. In the example shown in Figure 7, the calibration is enabled and the corresponding training text was "TESTING".

(10) The number of sequences to highlight every row and column in RCP (or every element in RSVP) can be determined in this text field both for calibration mode and for free spelling mode. The way a speller usually works includes a fixed number of highlighting repetitions for all elements during training. However, after the classifier is set up, the system is able to recognize a selected item with a lower number of repetitions in the free spelling mode (this number is obtained in the P300 Classifier process; see section "P300 Classifier automatic calling").

Fields (11) and (12) allow the user to indicate the subject identifier and the session to be stored, respectively.

(13) The button with the text "Classifier" opens a menu that allows the user to process the training data and to obtain a signal classifier to be applied in the free spelling mode (see section "P300 Classifier automatic calling").

(14) In a second line, a new and bigger text field is present. This text field is the typing bar; it is not an input field, but the place where the spelled text is shown. In the free spelling mode, any character detected by the system will appear there. Otherwise, in the case of the calibration mode, the text here is the same as the input above for the text to spell (9), but the already selected characters are represented more strongly than the characters still to spell, as a way to indicate to the users what character they should pay attention to (in the example in Figure 7 the letters "TE" have already been marked, so the user should be paying attention to the cell with the letter "S").

2.1.2. Grid parameters

This set of parameters (Figure 8, left) is accessed by double-clicking in the idle part of the GUI (i.e., the speller background). There, users can modify the number of rows and columns of the display and specify the size of the cells, as well as the empty space in between. The typing bar layout is defined here as well. There are some other parameters that allow the user to make changes on the keyboard letters' font type and size, as well as the cells' background colour for either the idle or highlighted state. Through this menu, users can fix also the size of the stimulus in the case of activating the RSVP paradigm. These parameters affect all cells equally; the configuration of a unique cell is explained in the following section.

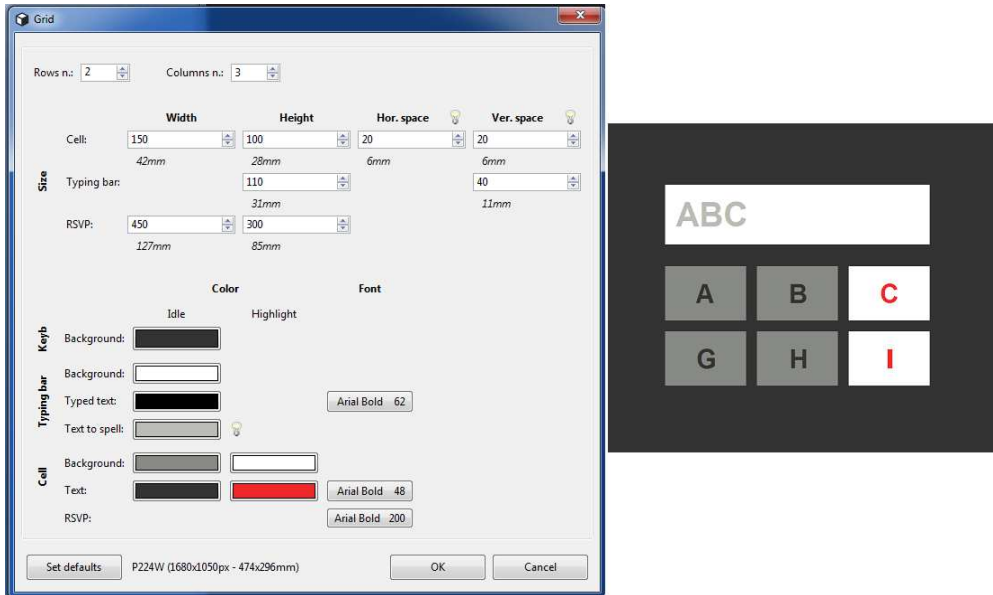


Figure 8. Grid parameters example. Left, grid configuration; right, grid result.

Through this menu, users can easily customize their P300 speller layout, changing the number of elements and the way these elements are represented. In the example shown in Figure 8 (right), the speller consists of a 2×3 matrix with black background, where each cell is 150 pixels wide and 100 high and it has a grey background and black text on it (idle state); when each cell is highlighted it changes its background to white and its content to red (in the example, the third column is being highlighted).

2.1.3. Cell parameters

The capability to customize each individual cell beyond the general appearance of the grid parameters is one of the main aspects of the UMA-BCI Speller. If users want to change a single cell's appearance, they can do it by double-clicking on a particular item; a cell-specific menu will appear (Figure 9, left). In this menu, two columns of parameters are present, "Idle" and "Highlight", for the case of the cell being in idle or highlighted state, respectively. The cell text and background colour for both states can be modified; if not marked in this menu, the values defined in the general grid parameters menu will apply.

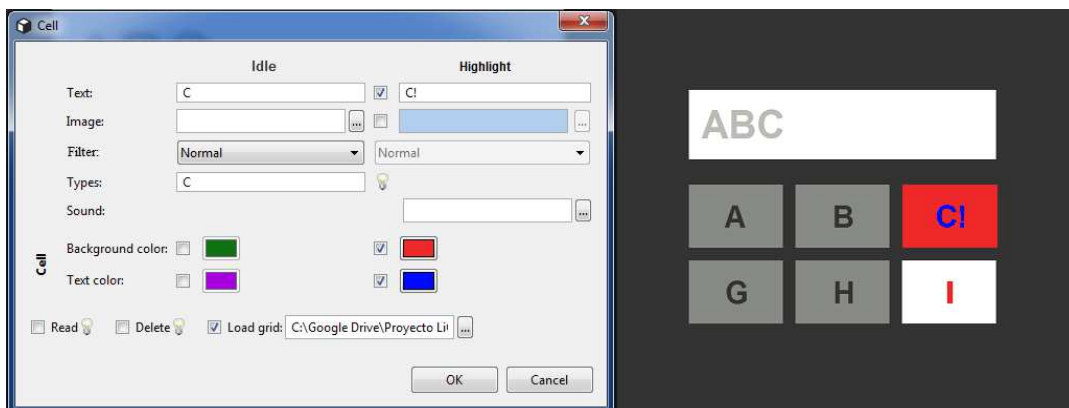


Figure 9. Cell parameters example. Left, cell configuration; right, cell result.

In the example shown in Figure 9 (right, where the third column is being highlighted), modifications to the upper-right cell have been made: the text and background colour have been changed, but this will only apply in the highlighted state (as these fields are marked). In this way, this cell will be highlighted with blue text over a red background, while it keeps the idle settings similar to the rest of cells, as the grid parameters apply (if the corresponding idle fields were marked, this cell would have a green background with purple text). The text for this cell in the highlighted state has changed as well, from "C" to "C!". Please note that the other cells do not change, in either the idle or the highlighted state.

Another important feature is that images can be included here instead of text, for both idle and highlight states. Images to be added can be applied via a couple of predefined filters in order to automatically invert the colours or desaturate them (changing the image colours into grayscale).

In the field "Types", users can set the text to be written in the typing bar when the cell is selected. In the common case of a speller matrix containing characters, the "Types" text should be a single character corresponding to the one actually shown. However, in other situations the "Types" field may be a whole word or sentence; that would be the case of cells containing pictograms that could mean "water", or "turn on the light", for example.

The field "Sound" allows a sound to be loaded to be heard when this cell is presented in the RSVP paradigm.

Two checkable options are present, "Read" and "Delete". With the first one, the user indicates that when this cell is selected, the system "reads" with a synthesized voice all the characters spelled to this point. The other option is used to include a "Delete" cell that removes the last spelled character.

Finally, the "Load grid" option allows the speller elements to be changed to a previously saved grid. In this way, the elements in the speller are updated depending on previous selections, so users can navigate through several keyboards, as in [54].

In the case of using the RSVP paradigm, only the "Highlight" parameters apply, as the symbols are presented serially in their highlighted state.

2.1.4. P300 Classifier automatic calling

One way of making the speller easier to use is to simplify the signal processing procedure. Usually, in a BCI2000 P3Speller experience, the operator uses a BCI2000 tool called "P300 Classifier" to analyze the EEG of a training session and to obtain the classifier parameters which will determine what items are selected by the user afterwards. It is not a complicated procedure, but it requires a few steps. In the UMA-BCI Speller application, this part is hidden, so users do not need to perform more actions after finishing the training. With UMA-BCI Speller, once the training session is finished, one single action groups all the subject's EEG raw signals and performs the necessary steps to obtain the optimal classifier. Clicking on the button Classifier (button 13 in Figure 7) causes the system to open a window to select the files with the calibration data (Figure 10, left). Once selected, an internal process uses the BCI2000 tool "P300 Classifier" to obtain the classifier parameters to be used in the free spelling

session. Another window with the results of the process is then shown (Figure 10, right). The results indicate if a classifier could be obtained with the supplied data and the classification accuracy corresponding to the number of sequences (set of row and column flashes). This information can be used to determine the optimal number of sequences to be applied in the free spelling mode (button 10 in Figure 7). In the example in Figure 10, four files with calibration data were used, resulting in 100% accuracy from the fifth sequence. The classifier parameters are saved in the "Output parameter file" and automatically loaded in the speller after closing the "Results" window.

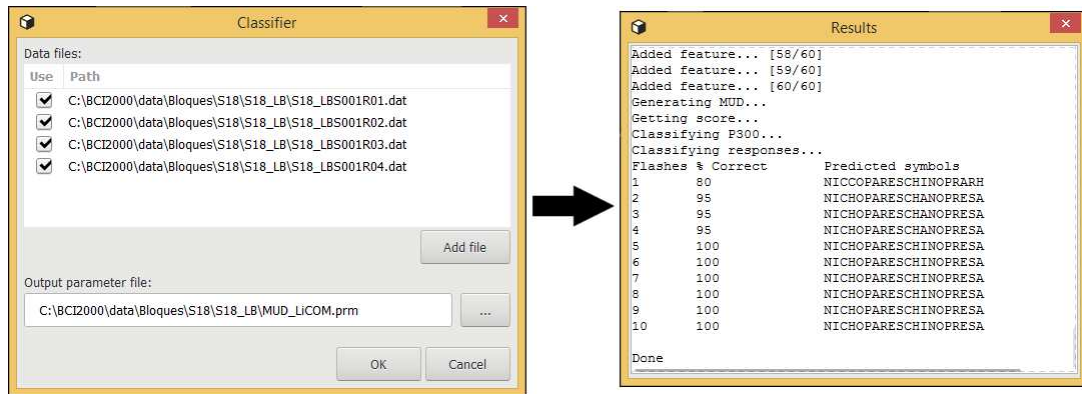


Figure 10. Classifier windows. Left, list of recent files for selecting the calibration data; right, calibration results

2.1.5. BCI2000 tab in Settings menu

This tab in the Settings menu makes it possible to modify some miscellaneous aspects that are directly related to the BCI2000 core in the UMA-BCI Speller application (Figure 11).

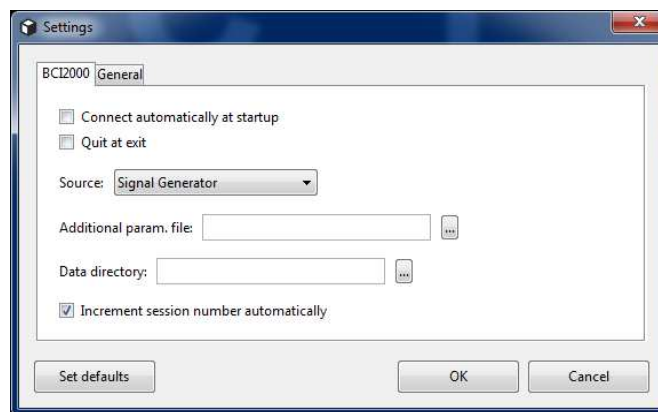


Figure 11. BCI2000 tab in Settings menu. Several options regarding the BCI2000 configuration are present in this settings tab

As in the case of common use of BCI2000, with UMA-BCI Speller different signal amplifiers can be used to register the EEG. The process for using them in UMA-BCI Speller is the same as in BCI2000: a specific module needs to be provided by manufacturers, which may be a ".dll" file or source code to be compiled. In both cases, once the BCI2000 system is ready to use them, UMA-BCI Speller just needs the user to select the desired option in the "Source" selector.

We stated before that users do not need to manage any BCI2000 parameters file in order to control the UMA-BCI Speller application as the basic configuration parameters are automatically loaded when the application is started from the "uma_bci_speller.prm" file in the case of the RCP paradigm (or the "uma_bci_speller_rsvp.prm" in the case of the RSVP paradigm). However, if users want to, they have the opportunity to load an additional parameters file ".prm" through the Settings "Additional parameters file". The parameters loaded in this way overwrite those automatically loaded from "uma_bci_speller.prm", but they do not persist for the next runs. If the users want to persistently update these parameters, the "uma_bci_speller.prm" file needs to be directly modified with a text editor or, in a safer way, through the BCI2000 Configuration tabs menu and overwriting it. This tab can be accessed through BCI2000 or by simply clicking on the "BCI2000 Config" button. Once pressed, it opens the original BCI2000 Configuration window, where the user can modify any parameter and save the changes in the default parameter file "uma_bci_speller.prm" (or "uma_bci_speller_rsvp.prm" for the RSVP paradigm).

2.2. GUI configuration examples

In this section we will show some examples of configurations obtained by modifying grid and cell parameters. In all these examples in Figure 12, the last column is highlighted.

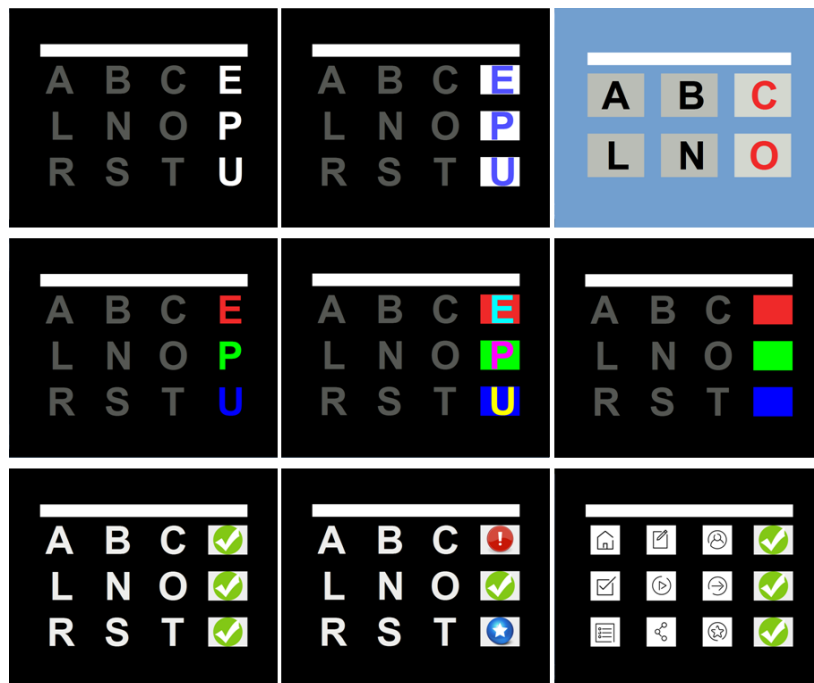


Figure 12. GUI configuration examples. Variations that show several possibilities of UMA-BCI Speller, from the change of text and background colour to the use of images

In the first row, three variations were obtained by changing the Grid parameters, so these changes affect all cells equally; this means that all idle cells are similar and that all the cells in a column or row that are highlighted are similar too. In the first case, grey letters over black background are highlighted by changing their colour to white, which is the most commonly used configuration, following the proposal of Farwell and Donchin [12]; most papers use this layout, possibly due to the complexity of the

platforms used regarding the customization of the speller. Next, in the second example, the highlighting is achieved with blue text over a white background on each cell. Finally, in the third example, the number of elements has been reduced to six in a 2×3 matrix, the speller background has been turned to pale blue, each cell now has black text over grey background, and the highlighting consists of turning the cells text to red.

The second row shows another three variations, this time using the Cell parameters; in this way, each cell is treated independently. In the examples, the idle cells are set identically to be grey without any background, while the highlighting of the cells is done with different colours (font and background) for each one: red, green and blue fonts in the first case; red, green and blue background with the inverted colour for the font (pale blue, purple and yellow) in the second example; red, green and blue background and font colour in the third example.

Finally, the third row shows three examples of cell parameters using images. In the first one, a unique image is set to highlight all cells, while the second example uses a different highlighting image for each cell. The last example shows the case of cells that contain not characters, but pictograms. The highlighting of the cells in this case is done with a unique image. As mentioned, the selection of a pictogram could involve the spelling of a string of characters (e.g. "go home"). Images with an alpha channel can be used, which could be especially important in the case of showing pictograms. In future updates of the software, it is intended that the selection of a cell causes some action in connection with external devices, such as performing a given task in a domotic environment.

2.3. Hardware and software requirements

The UMA-BCI Speller application does not require any special features. As it wraps BCI2000, the whole system requirements would be those of BCI2000. The BCI2000 platform is not a heavy software, it only requires a Windows-based operating system (Windows 2000 or newer) and one of the data acquisition devices that BCI2000 supports [55]. BCI2000 can also run on Linux and OS X systems, but there is no binary distribution for these systems (users would need to compile BCI2000 before using it). Besides, only a minority of source modules support operating systems other than Windows (those modules that receive data through a TCP connection [56]).

2.4. Licensing and availability

The UMA-BCI Speller application is implemented using BCI2000, which is available under the GNU General Public License (GPL). As a consequence, the use and distribution of UMA-BCI Speller is protected by the same GPL license. This means that anyone can use UMA-BCI Speller, that the source code is publicly available, and that anyone can develop and distribute derived software products under any terms, provided that the full source code of the derived product is made publicly available for download under the terms of the GPL.

The full source code of UMA-BCI Speller can be accessed in:
https://proyectos.diana.uma.es/umabci_speller

The same URL contains a detailed User Manual with all the information regarding the installation, configuration and use of the UMA-BCI Speller.

3. Usability tests

In order to check the feasibility of the UMA-BCI Speller, two aspects should be considered: its reliability and usability.

On the one hand, as the inner core of the application is BCI2000, the UMA-BCI Speller is as reliable as BCI2000, which has been widely tested and accepted.

On the other hand, regarding the ease of configuration and use, we carried out usability tests with the aim of quantifying it. Ten subjects were asked to complete two exercises and to evaluate the difficulty through a System Usability Scale (SUS) test [57]. These exercises were proposed in such a way that most of the functionalities of the system were covered. All subjects were first grade students of the Graduate School of Cognitics (Bordeaux, France) with ages between 21 and 23 years. About a week before the test, the manual for the application (around 3300 words) was sent to participants so they could read it and have a first contact with the application; however, they did not use the tool until the day of the evaluation. The participants declared that they invested an average time of 26 ± 16.47 minutes to read the manual.

The first exercise consisted of changing the speller layout in a nine-step sequence which included most of the configuration options available in both the Grid and Cell menus (see Figure 13). The users took 12.6 ± 1.96 minutes on average to perform this task, this time includes a possible period of familiarization with the system.

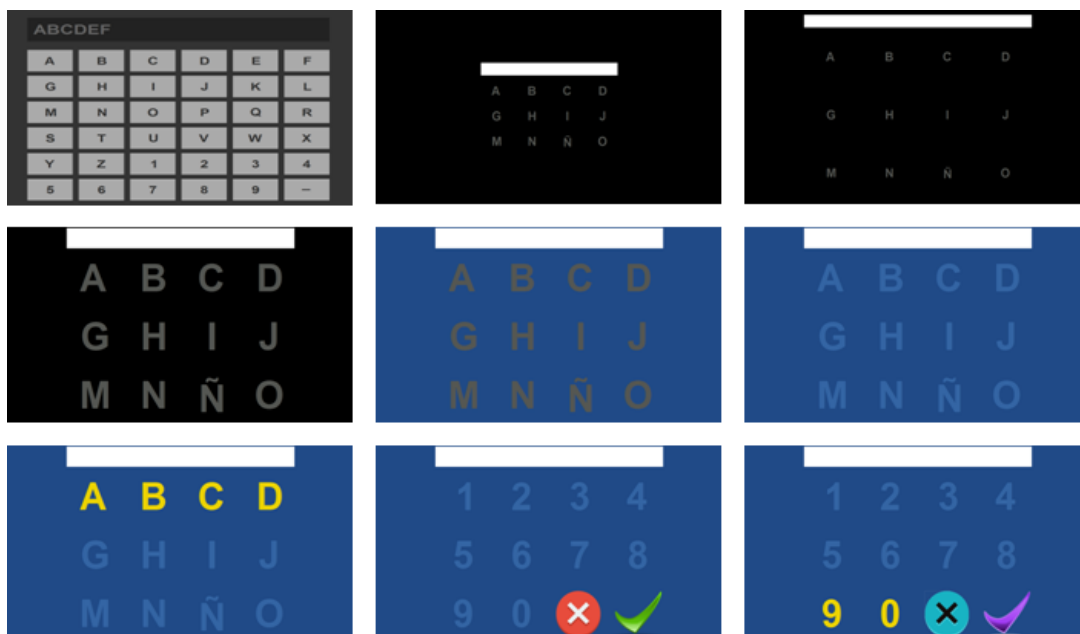


Figure 13. Sequence of steps of the proposed exercise for evaluation purposes. The users were asked to change the elements layout, size, colour, content and highlighting stimulus

After that, the volunteers carried out a simulation of an experiment as a second exercise, using the same layout to perform a calibration of the system (using the BCI2000 Signal

Generator as signal source) and loading a previously recorded file with real EEG data to get the classifier. This last task took them 23.4 ± 8 minutes. Although this task was quite simple (it could have been done in less than 5 minutes), all subjects made a mistake that made it impossible to complete the exercise and it took them a long time to realize the error (because it was not correctly explained in the manual). Thanks to this mistake, we corrected the manual in order to warn about it.

Participant	Time (min)			SUS score
	Read the manual	Complete task 1	Complete task 2	
P1	20	12	27	95
P2	30	12	40	90
P3	15	13	30	80
P4	20	12	30	85
P5	20	9	15	85
P6	25	15	18	70
P7	60	11	17	90
P8	25	15	20	72.5
P9	0	15	19	60
P10	45	12	18	90
Mean	26 ± 16.47	12.6 ± 1.96	23.4 ± 8	81.75 ± 11.06

Table 1. Usability assessment of the UMA-BCI Speller.

The SUS test consists of ten simple questions that are answered in a 1–5 scale and offer a global vision of the usability of the system in a 0–100 range. The average result of the questionnaires was 81.75 ± 11.06 , which can be considered a score between "good" and "excellent" according to [58]. The results regarding the time needed to complete each task and the SUS test are shown in Table 1.

4. Discussion and conclusion

We have presented in this paper a novel BCI tool focused on a speller application: the UMA-BCI Speller. The aim of this tool is to provide end users with an easy to use open source P300 speller. It is based on the widely used platform BCI2000, so it takes advantage of the reliability that such a platform offers. The UMA-BCI Speller wraps BCI2000 in such a way that its configuration and use is much more visual and easier. In this way, we think that the use of BCI systems at end users' homes will be enhanced. At the same time, the UMA BCI Speller allows new features to be added in order to test different variations of the speller layout, a capability that BCI researchers can exploit. The UMA-BCI Speller supports two P300 stimulations: RCP and RSVP. Users can configure their speller more appropriately using characters, images or sound cues, and they can navigate through different layouts, thus opening the door to complex speller configurations.

The positive results of the SUS tests support the use of this tool for non-experienced users. Ten subjects, in an hour approximately, could read the application manual and perform two common tasks: to implement an RCP speller and to test it. Based on our experience, these same tasks could have required much more time and technical skills in

the case of using the general purpose platform BCI2000. The average results of the usability questionnaires score the application in between "good" and "excellent". The lowest score was given by a subject who could not read the manual before the tests; so if we do not consider this subject, the average score would be 84.17, still closer to the excellent limit (85.5) obtained in [58].

The UMA-BCI group continues to work on new features to be added to the next software releases. One of these improvements is to allow the modification of the row-column paradigm in order to use other proposals, such as the Checkerboard paradigm [42]. The RSVP paradigm can also include new features in order to improve the ITR, such as the Triple RSVP described in [59]. It would be interesting also to adapt the interface in order to extend its functionality beyond spelling, for example, making it possible to use it to browse the internet or to activate external actuators. More improvements to be considered consist of the use of generic models or adaptive calibration techniques that can reduce the calibration time needed to use the system. UMA-BCI Speller uses the current release of BCI2000 (v3.0.5); however recent beta versions (e.g. BCI2000 v3.6) include new features that enable a "quasi-asynchronous" control and let the system dynamically change the number of stimulus repetitions through the use of the parameters `MinimumEvidence` and `AccumulateEvidence` [60]. Upcoming versions of UMA-BCI Speller will be updated to include the new features of BCI2000 beta releases.

Acknowledgements. This work was partially supported by the Spanish Ministry of Economy and Competitiveness through the project LICOM (DPI2015-67064-R) and by the European Regional Development Fund (ERDF).

Conflict of interest statement. Authors declare no conflict of interest.

References

- [1] P. Augustyniak, Z. Mikrut, Interfaces for tetraplegic people - Review of solutions supporting activities of daily living, *Bio-Algorithms and Med-Systems*. 11 (2015) 183–194. doi:10.1515/bams-2015-0015.
- [2] A.R. Murguialday, J. Hill, M. Bensch, S. Martens, S. Halder, F. Nijboer, B. Schoelkopf, N. Birbaumer, A. Gharabaghi, Transition from the locked in to the completely locked-in state: A physiological analysis, *Clin. Neurophysiol.* 122 (2011) 925–933. doi:10.1016/j.clinph.2010.08.019.
- [3] L.F. Nicolas-Alonso, J. Gomez-Gil, Brain computer interfaces, a review, *Sensors*. 12 (2012) 1211–1279. doi:10.3390/s120201211.
- [4] A. Rezeika, M. Benda, P. Stawicki, F. Gembler, A. Saboor, I. Volosyak, Brain–computer interface spellers: A review, *Brain Sci.* 8 (2018). doi:10.3390/brainsci8040057.
- [5] J. Pascual, F. Velasco-Alvarez, K.R.K.-R. Muller, C. Vidaurre, First study towards linear control of an upper-limb neuroprosthesis with an EEG-based Brain-Computer Interface, in: *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS, 2012*: pp. 3269–3273. doi:10.1109/EMBC.2012.6346663.
- [6] R. Ron-Angevin, F. Velasco-Álvarez, Á. Fernández-Rodríguez, A. Díaz-Estrella, M.J. Blanca-Mena, F.J. Vizcaíno-Martín, Brain-Computer Interface application: auditory serial interface to control a two-class motor-imagery-based wheelchair, *J. Neuroeng. Rehabil.* 14 (2017). doi:10.1186/s12984-017-0261-y.
- [7] W. Alrajhi, D. Alaloola, A. Albarqawi, Smart home: Toward daily use of BCI-based systems, in: *2017 Int. Conf. Informatics, Heal. Technol. ICIHT 2017, 2017*. doi:10.1109/ICIHT.2017.7899002.
- [8] M. Bensch, A.A. Karim, J. Mellinger, T. Hinterberger, M. Tangermann, M. Bogdan, W. Rosenstiel, N. Birbaumer, Nessi: An EEG-controlled web browser for severely paralyzed patients, *Comput. Intell. Neurosci.* 2007 (2007) 71863. doi:10.1155/2007/71863.
- [9] R. Scherer, E.C. V Friedrich, B. Allison, M. Pröll, M. Chung, W. Cheung, and C.N. R. P. N. Rao, Non-invasive brain-computer interfaces: Enhanced gaming and robotic control, *Adv. Comput. Intell. Gence, Lect. Notes Comput. Sci.* 6691/2011 (2011) 362–369. <http://www.springerlink.com/content/aw7782349378k206/>.
- [10] M. Nakanishi, Y. Wang, X. Chen, Y. Te Wang, X. Gao, T.P. Jung, Enhancing detection of SSVEPs for a high-speed brain speller using task-related component analysis, *IEEE Trans. Biomed. Eng.* 65 (2018) 104–112. doi:10.1109/TBME.2017.2694818.
- [11] U. Hoffmann, E.J. Fimbel, T. Keller, Brain-computer interface based on high frequency steady-state visual evoked potentials: A feasibility study, *2009 4th Int. IEEE/EMBS Conf. Neural Eng.* (2009) 466–469. doi:10.1109/NER.2009.5109334.
- [12] L.A. Farwell, E. Donchin, Talking off the top of your head: toward a mental prosthesis utilizing event-related potentials, *Electroencephalogr. Clin. Neurophysiol.* 70 (1988) 510–523.
- [13] T. Kaufmann, A. Kübler, Beyond maximum speed - A novel two-stimulus paradigm for brain-computer interfaces based on event-related potentials (P300-BCI), *J. Neural Eng.*

- 11 (2014). doi:10.1088/1741-2560/11/5/056004.
- [14] J. Polich, Updating P300: An integrative theory of P3a and P3b, *Clin. Neurophysiol.* 118 (2007) 2128–2148. doi:10.1016/j.clinph.2007.04.019.
- [15] E. Yin, T. Zeyl, R. Saab, D. Hu, Z. Zhou, T. Chau, An Auditory-Tactile Visual Saccade-Independent P300 Brain-Computer Interface, *Int. J. Neural Syst.* 26 (2016) 1650001. doi:10.1142/S0129065716500015.
- [16] F. Aloise, F. Schettini, P. Aricò, L. Bianchi, A. Riccio, M. Mecella, F. Babiloni, D. Mattia, F. Cincotti, Advanced brain computer interface for communication and control, *Proc. Int. Conf. Adv. Vis. Interfaces - AVI '10.* (2010) 399. doi:10.1145/1842993.1843076.
- [17] I. Iturrate, J.M. Antelis, A. Kübler, J. Minguez, A noninvasive brain-actuated wheelchair based on a P300 neurophysiological protocol and automated navigation, *IEEE Trans. Robot.* 25 (2009) 614–627. doi:10.1109/TRO.2009.2020347.
- [18] P. Brunner, S. Joshi, S. Briskin, J.R. Wolpaw, H. Bischof, G. Schalk, Does the “P300” speller depend on eye gaze?, *J. Neural Eng.* 7 (2010). doi:10.1088/1741-2560/7/5/056013.
- [19] J.R. Wolpaw, R.S. Bedlack, D.J. Reda, R.J. Ringer, P.G. Banks, T.M. Vaughan, S.M. Heckman, L.M. McCane, C.S. Carmack, S. Winden, D.J. McFarland, E.W. Sellers, H. Shi, T. Paine, D.S. Higgins, A.C. Lo, H.S. Patwa, K.J. Hill, G.D. Huang, R.L. Ruff, Independent home use of a brain-computer interface by people with amyotrophic lateral sclerosis, *Neurology.* (2018) 10.1212/WNL.0000000000005812. doi:10.1212/WNL.0000000000005812.
- [20] S. Taherian, T.C. Davies, Caregiver and special education staff perspectives of a commercial brain-computer interface as access technology: a qualitative study, *Brain-Computer Interfaces.* 5 (2018) 73–87. doi:10.1080/2326263X.2018.1505191.
- [21] G. Liberati, A. Pizzimenti, L. Simione, A. Riccio, F. Schettini, M. Inghilleri, D. Mattia, F. Cincotti, Developing brain-computer interfaces from a user-centered perspective: Assessing the needs of persons with amyotrophic lateral sclerosis, caregivers, and professionals, *Appl. Ergon.* 50 (2015) 139–146. doi:10.1016/j.apergo.2015.03.012.
- [22] C. Brunner, G. Andreoni, L. Bianchi, B. Blankertz, C. Breitwieser, S. Kanoh, I.P. Susila, B. Venthur, BCI Software Platforms, in: *Toward. Pract. Brain-Computer Interfaces, 2013:* pp. 303–331. doi:10.1007/978-3-642-29746-5_16.
- [23] G. Schalk, D.J. McFarland, T. Hinterberger, N. Birbaumer, J.R. Wolpaw, BCI2000: A general-purpose brain-computer interface (BCI) system, *IEEE Trans. Biomed. Eng.* 51 (2004) 1034–1043. doi:10.1109/TBME.2004.827072.
- [24] C. Arrouët, M. Congedo, J.-E. Marvie, F. Lamarche, A. Lécuyer, B. Arnaldi, Open-ViBE: A three dimensional platform for real-time neuroscience, *J. Neurother.* 9 (2005) 3–25. doi:10.1300/J184v09n01_02.
- [25] C.A. Kothe, S. Makeig, BCILAB: A platform for brain-computer interface development, *J. Neural Eng.* 10 (2013). doi:10.1088/1741-2560/10/5/056014.
- [26] BCI++, (n.d.). <http://www.sensibilab.lecco.polimi.it/bci>.
- [27] L. Bianchi, F. Babiloni, F. Cincotti, S. Salinari, M.G. Marciani, Introducing BF++: A C++

- framework for cognitive bio-feedback systems design, *Methods Inf. Med.* 42 (2003) 104–110.
- [28] A. Delorme, S. Makeig, EEGLAB: An open source toolbox for analysis of single-trial EEG dynamics including independent component analysis, *J. Neurosci. Methods.* 134 (2004) 9–21. doi:10.1016/j.jneumeth.2003.10.009.
- [29] a. Schlogl, C. Brunner, BioSig: A Free and Open Source Software Library for BCI Research, *Computer (Long. Beach. Calif.)*. 41 (2008) 44–50. doi:10.1109/MC.2008.407.
- [30] I.P. Susila, S. Kanoh, K.I. Miyamoto, T. Yoshinobu, xBCI: A generic platform for development of an online BCI system, *IEEJ Trans. Electr. Electron. Eng.* 5 (2010) 467–473. doi:10.1002/tee.20560.
- [31] B. Venthur, S. Scholler, J. Williamson, S. Dähne, M.S. Treder, M.T. Kramarek, K.-R. Müller, B. Blankertz, Pyff - A pythonic framework for feedback applications and stimulus presentation in neuroscience, *Front. Neuroinform.* 4 (2010). doi:10.3389/2010.00100.
- [32] S.K. Yeom, S. Fazli, S.W. Lee, P300 visual speller based on random set presentation, 2014 Int. Winter Work. Brain-Computer Interface, BCI 2014. (2014). doi:10.1109/iww-BCI.2014.6782567.
- [33] S.K. Yeom, S. Fazli, K.R.M. Ller, S.W. Lee, An efficient ERP-based brain-computer interface using random set presentation and face familiarity, *PLoS One.* 9 (2014) 1–13. doi:10.1371/journal.pone.0111157.
- [34] W. Speier, A. Deshpande, L. Cui, N. Chandravadia, D. Roberts, N. Pouratian, A comparison of stimulus types in online classification of the P300 Speller using language models, *PLoS One.* 12 (2017) 1–11. doi:10.1371/journal.pone.0175382.
- [35] I. Käthner, A. Kübler, S. Halder, Rapid P300 brain-computer interface communication with a head-mounted display, *Front. Neurosci.* 9 (2015) 1–13. doi:10.3389/fnins.2015.00207.
- [36] S.T. Ahi, S. Member, H. Kambara, Y. Koike, A Dictionary-Driven P300 Speller, *IEEE Trans. Neural Syst. Rehabil. Eng.* 19 (2011) 6–14.
- [37] Y. Li, C.S. Nam, B.B. Shadden, S.L. Johnson, A p300-based brain-computer interface: Effects of interface type and screen size, *Int. J. Hum. Comput. Interact.* 27 (2011) 52–68. doi:10.1080/10447318.2011.535753.
- [38] J. Jin, E.W. Sellers, X. Wang, Targeting an efficient target-to-target interval for P300 speller brain-computer interfaces, *Med. Biol. Eng. Comput.* 50 (2012) 289–296. doi:10.1007/s11517-012-0868-x.
- [39] Y. Sakai, T. Yagi, Alphabet matrix layout in P300 speller may alter its performance, 4th Biomed. Eng. Int. Conf. BMEiCON-2011. (2011) 89–92.
- [40] R. Fazel-Rezai, Human Error in P300 Speller Paradigm for Brain-Computer Interface, 29th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (2007) 2516–2519. doi:10.1109/IEMBS.2007.4352840.
- [41] D.B. Ryan, K.A. Colwell, C.S. Throckmorton, L.M. Collins, K. Caves, E.W. Sellers, Evaluating Brain-Computer Interface Performance in an ALS Population: Checkerboard and Color Paradigms, *Clin. EEG Neurosci.* 49 (2018) 114–121.

doi:10.1177/1550059417737443.

- [42] G. Townsend, B.K. LaPallo, C.B. Boulay, D.J. Krusienski, G.E. Frye, C.K. Hauser, N.E. Schwartz, T.M. Vaughan, J.R. Wolpaw, E.W. Sellers, A novel P300-based brain-computer interface stimulus presentation paradigm: Moving beyond rows and columns, *Clin. Neurophysiol.* 121 (2010) 1109–1120. doi:10.1016/j.clinph.2010.01.030.
- [43] Q.T. Obeidat, T.A. Campbell, J. Kong, Introducing the Edges Paradigm: A P300 Brain-Computer Interface for Spelling Written Words, *IEEE Trans. Human-Machine Syst.* 45 (2015) 727–738. doi:10.1109/THMS.2015.2456017.
- [44] B.N. Cuthbert, H.T. Schupp, M.M. Bradley, N. Birbaumer, P.J. Lang, Brain potentials in affective picture processing: Covariation with autonomic arousal and affective report, *Biol. Psychol.* 52 (2000) 95–111. doi:10.1016/S0301-0511(99)00044-7.
- [45] C. Guger, R. Ortner, S. Dimov, B. Allison, A comparison of face speller approaches for P300 BCIs, in: 2016 IEEE Int. Conf. Syst. Man, Cybern. SMC 2016 - Conf. Proc., 2017: pp. 4809–4812. doi:10.1109/SMC.2016.7844989.
- [46] B. Rozenkrants, J. Polich, Affective ERP processing in a visual oddball task: Arousal, valence, and gender, *Clin. Neurophysiol.* 119 (2008) 2260–2265. doi:10.1016/j.clinph.2008.07.213.
- [47] Y. Liu, Z. Zhou, D. Hu, Comparison of stimulus types in visual P300 speller of brain-computer interfaces, *Proc. 9th IEEE Int. Conf. Cogn. Informatics, ICCI 2010.* (2010) 273–279. doi:10.1109/COGINF.2010.5599730.
- [48] J. Shi, J. Shen, Y. Ji, F. Du, A submatrix-based P300 brain-computer interface stimulus presentation paradigm, *J. Zhejiang Univ. Sci. C.* 13 (2012) 452–459. doi:10.1631/jzus.C1100328.
- [49] J.S. Eom, H.R. Yang, M.S. Park, J.H. Sohn, P300 speller using a new stimulus presentation paradigm, 2013 Int. Winter Work. Brain-Computer Interface, BCI 2013. (2013) 98–99. doi:10.1109/IWW-BCI.2013.6506644.
- [50] J. Jin, P. Horki, C. Brunner, X. Wang, C. Neuper, G. Pfurtscheller, A new P300 stimulus presentation pattern for EEG-based spelling systems, *Biomed. Tech.* 55 (2010) 203–210. doi:10.1515/BMT.2010.029.
- [51] C. Polprasert, P. Kukieattikool, T. Demeechai, J.A. Ritcey, S. Siwamogsatham, New stimulation pattern design to improve P300-based matrix speller performance at high flash rate, *J. Neural Eng.* 10 (2013). doi:10.1088/1741-2560/10/3/036012.
- [52] T. Kaufmann, S.M. Schulz, C. Gr??nzinger, A. K??bler, Flashing characters with famous faces improves ERP-based brain-computer interface performance, *J. Neural Eng.* 8 (2011). doi:10.1088/1741-2560/8/5/056016.
- [53] Q. Li, S. Liu, J. Li, O. Bai, Use of a green familiar faces paradigm improves P300-speller brain-computer interface performance, *PLoS One.* 10 (2015) 1–15. doi:10.1371/journal.pone.0130325.
- [54] G. Pires, U. Nunes, M. Castelo-Branco, GIBS block speller: Toward a gaze-independent P300-based BCI, *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS.* (2011) 6360–6364. doi:10.1109/IEMBS.2011.6091570.

- [55] G. Schalk, J. Mellinger, *A Practical Guide to Brain-Computer Interfacing with BCI2000.*, Springer London Dordrecht Heidelberg New York, 2010. doi:10.1007/978-1-84996-092-2.
- [56] BCI2000 Binaries - BCI2000 Wiki, (n.d.).
https://www.bci2000.org/mediawiki/index.php/BCI2000_Binaries#Non-Windows_Operating_Systems (accessed January 31, 2019).
- [57] J. Brooke, SUS - A quick and dirty usability scale, in: *Usability Eval. Ind.*, 1996. doi:10.1002/hbm.20701.
- [58] A. Bangor, P. Kortum, J. Miller, *Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale*, 2009.
- [59] Z. Lin, C. Zhang, Y. Zeng, L. Tong, B. Yan, A novel P300 BCI speller based on the Triple RSVP paradigm, *Sci. Rep.* 8 (2018) 1–9. doi:10.1038/s41598-018-21717-y.
- [60] User Reference:P3SpellerTask - BCI2000 Wiki, (n.d.).
https://www.bci2000.org/mediawiki/index.php/User_Reference:P3SpellerTask#MinimumEvidence (accessed January 31, 2019).