A Predictive Speller Controlled by a Brain-Computer Interface Based on Motor Imagery

TIZIANO D'ALBIS, ROSSELLA BLATT, ROBERTO TEDESCO, LICIA SBATTELLA, and MATTEO MATTEUCCI, Politecnico di Milano

Persons suffering from motor disorders have limited possibilities for communicating and normally require assistive technologies to fulfill this primary need. Promising means of providing basic communication abilities to subjects affected by severe motor impairments include brain-computer interfaces (BCIs), that is, systems that directly translate brain signals into device commands, bypassing any muscle or nerve mediation. To date, the use of BCIs for effective verbal communication is yet an open issue, primarily due to the low rates of information transfer that can be achieved with this technology. Still, performance of BCI spelling applications could be considerably improved by a smart user interface design and by the adoption of natural language processing (NLP) techniques for text prediction. The objective of this work is to suggest an approach and a user interface for BCI spelling applications combining state-of-the-art BCI and NLP techniques to maximize the overall communication rate of the system. The BCI paradigm adopted is motor imagery, that is, when the subject imagines moving a certain part of the body, he/she produces modifications to specific brain rhythms that are detected in real-time through an electroencephalogram and translated into commands for a spelling application. By maximizing the overall communication rate, our approach is twofold: on one hand, we maximize the information transfer rate from the control signal, on the other hand, we optimize the way this information is employed for the purpose of verbal communication. The achieved results are satisfactory and comparable with the latest works reported in literature on motor-imagery BCI spellers. For the three subjects tested, we obtained a spelling rate of respectively 3 char/min, 2.7 char/min, and 2 char/min.

Categories and Subject Descriptors: I.5.4 [**Pattern Recognition**]: Applications

General Terms: Algorithms, Human Factors

Additional Key Words and Phrases: BMI, BCI, NLP, speller, brain-computer interface, text prediction

ACM Reference Format:
D'Albis, T., Blatt, R., Tedesco, R., Sbattella, L., and Matteucci, M. 2012. A predictive speller controlled by a brain-computer interface based on motor imagery. ACM Trans. Comput.-Hum. Interact. 19, 3, Article 20 (October 2012), 25 pages.
DOI = 10.1145/2362364.2362368 http://doi.acm.org/10.1145/2362364.2362368

20

1. INTRODUCTION

The ability to communicate with other people is one of the main factors that makes the life of any human being enjoyable. Individuals suffering from motor disorders may have limited possibilities for communicating and may require assistive technologies to fulfill this primary need. Some people may have completely lost control over voluntary muscles while still being fully conscious and aware of what is happening in their

Authors' addresses: T. D'Albis, R. Blatt, R. Tedesco, L. Sbattella, and M. Matteucci, Politecnico di Milano, Dip. de Electronica e Informazione, Via Ponzio 34/5, I-20133, Milan, Italy; corresponding author's email: matteucci@elet.polimi.it.

DOI 10.1145/2362364.2362368 http://doi.acm.org/10.1145/2362364.2362368

environment. This is the case of the so-called "locked-in" syndrome, which is normally caused by lesions to the brainstem or by neurodegenerative diseases [Plum and Posner 1966; Bauer et al. 1979].

Brain-computer interfaces (BCIs) are a promising means of providing basic communication abilities to such people [Wolpaw et al. 2002; Graimann et al. 2011]. A brain-computer interface is a system that bypasses any muscle or nerve mediation and translates signals acquired from the brain into commands for an external device [Migliore et al. 2008]. The most common approach of acquiring brain signals non-invasively is electroencephalography (EEG) in which the electrical activity induced by cortical neurons is sensed through a set of electrodes placed on the scalp of the subject. Performing appropriate cognitive tasks, users may learn to produce voluntary modifications on their EEG signals that can be detected in real-time and used to generate a control signal [Curran and Stokes 2003].

BCI systems are normally characterized by low rates of information transfer and low accuracies [Wolpaw et al. 2002], while communication is a complex process that requires a large set of language symbols for conveying messages. Mapping this large set of symbols into a limited number of input commands is one of the major challenges that assistive communication devices need to face. On the other hand, natural language encodes messages in a redundant way, and verbal communication is characterized by recurrent structures imposed by grammatical and syntactical rules. Therefore, natural language processing (NLP) techniques [Manning and Schütze 1999] may be employed to exploit language redundancies and improve the performance of assistive communication devices.

The objective of our work is to suggest an approach and an interface for BCI spelling applications, adopting state-of-the-art BCI and NLP techniques to maximize the overall communication rate of the system. The BCI paradigm we adopt is motor imagery, that is, when the subject imagines moving a certain part of the body (such as a hand or a foot), he/she produces modifications on specific brain rhythms characteristic of the motor cortex [Jeannerod 1995; Pfurtscheller and Neuper 1997]. These modifications are detected in real-time from the EEG signal and mapped into different choices available in the spelling application. By maximizing the overall communication rate, our approach is twofold: on one hand, we maximize the information transfer rate from the control signal; on the other hand, we optimize the way this information is employed for the purpose of verbal communication. Dealing with the first issue, we define novel features to be extracted from the EEG signal and employ state-of-the-art machine learning techniques for feature selection and classification. Dealing with the second issue, we propose an original speller interface in which redundancies in natural language are used to speed up the selection of symbols and word suggestions are provided during the composition.

The article is organized as follows: Section 2 introduces some approaches that can be found in the literature on BCI spellers; Section 3 describes the application protocol we adopted in our BCI speller; and Section 4 gives an overview of the different modules that constitute our system: the BCI module (Section 5), the user interface (Section 6), and the language prediction module (Section 7). In Section 8, we describe the experiments and the measures used to assess the system performance and report the results obtained. Finally, in Section 9, we discuss the results and outline possible lines of improvement for future developments of this work. For the sake of readability, several details regarding the BCI module have been omitted from the main text of this article; interested readers can refer to Online Appendix A for details on the BCI processing pipeline and to Online Appendix B, both available in the ACM Digital Library, for details on the experiments and tests performed with the BCI module alone.

2. RELATED WORK

One of the first BCI spellers that has been developed is the so-called "Thought Translation Device" [Birbaumer et al. 1999]. This system employed slow cortical potentials to generate control signals and achieved a spelling rate of about 0.5 char/min, performing binary decisions for the selection of symbols. The same symbol selection strategy was also combined with a motor-imagery BCI paradigm achieving a spelling rate of about 1 char/min [Neuper et al. 2003]. In 2003, the group led by Gert Pfurtscheller developed the "Virtual Keyboard", a spelling application controlled by a two-class BCI based on motor imagery [Obermaier et al. 2003]. In its first version, the system performed a binary classification on two different mental tasks, and symbol selection was performed with a dichotomous search procedure. With this system, three subjects achieved a spelling rate of, respectively, 0.85, 1.02, and 0.67 char/min [Obermaier et al. 2003]. Later, the same group proposed another version of the Virtual Keyboard controlled by a voluntary asynchronous modulation of the brain rhythms associated with three motor-imagery tasks [Scherer et al. 2004]. With this system, two subjects achieved a spelling rate of, respectively, 3.38 and 2.85 char/min, while a third subject was completely unable to use the spelling application [Scherer et al. 2004].

BCI spelling devices based on paradigms different from that of motor imagery have been developed, too—P300 being one of the most successful. P300 is a positive deflection in the EEG signal, appearing approximately 300 ms after the presentation of rare or surprising task-relevant stimuli [Polich 1997]. To evoke a P300, subjects are asked to observe a random sequence of stimuli such that one stimulus (the oddball stimulus) appears only rarely in the sequence, while the other stimuli (the normal stimuli) appear more often. Most of the BCI spellers based on P300 are inspired by the system proposed by Farwell and Donchin [1988]. This speller consisted of a matrix of alphabetic symbols in which rows and columns are flashed in random order: flashes of a row or column containing the desired symbol constituted the oddball stimulus, while all other flashes constituted the normal stimuli. Using this approach, Farwell and Donchin obtained a spelling rate of 2.3 char/min, and more recent studies adopting the same paradigm reported spelling rates from 4 to 6 chars/min [Donchin et al. 2000; Wang et al. 2005; Dal Seno et al. 2010a]. A BCI based on P300 also has been tested recently in conjunction with a predictive speller [Ryan et al. 2011]. In this study, language prediction increased the overall communication rate (obtaining an average spelling rate of 5.28 char/min) but caused a reduced accuracy of the BCI classifier alone. Although often associated with higher communication rates, P300 spellers may be tedious to use, since the subjects are required to concentrate for a significant amount of time while being repetitively exposed to flashing visual stimuli.

To our knowledge, the only other predictive BCI spelling application controlled by motor imagery reported in the literature is the "Hex-o-Spell" [Blankertz et al. 2006]. In this application, letters are displayed in hexagons and selections are performed rotating and scaling an arrow by means of two different motor-imagery tasks. A language model is used for controlling the arrangement of symbols into the hexagons to speed up the selection process. This application has been tested by two untrained subjects, achieving a spelling rate between 2.3 and 5 char/min for one subject and between 4.6 and 7.6 char/min for a second subject. As with the Hex-o-Spell, we adopt a language model to speed up text composition, but we avoid a context-dependent rearrangement of the language symbols in order to maintain that the user interface is simple and intuitive. Instead, we introduce a novel selection strategy in which only the most probable symbols in the composition context are enabled for selection. Additionally, we provide word-level language predictions and propose a user interface that is designed to minimize the impact of BCI errors on the overall communication rate.

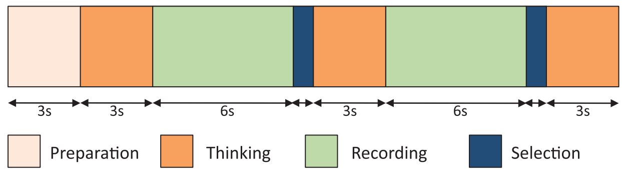
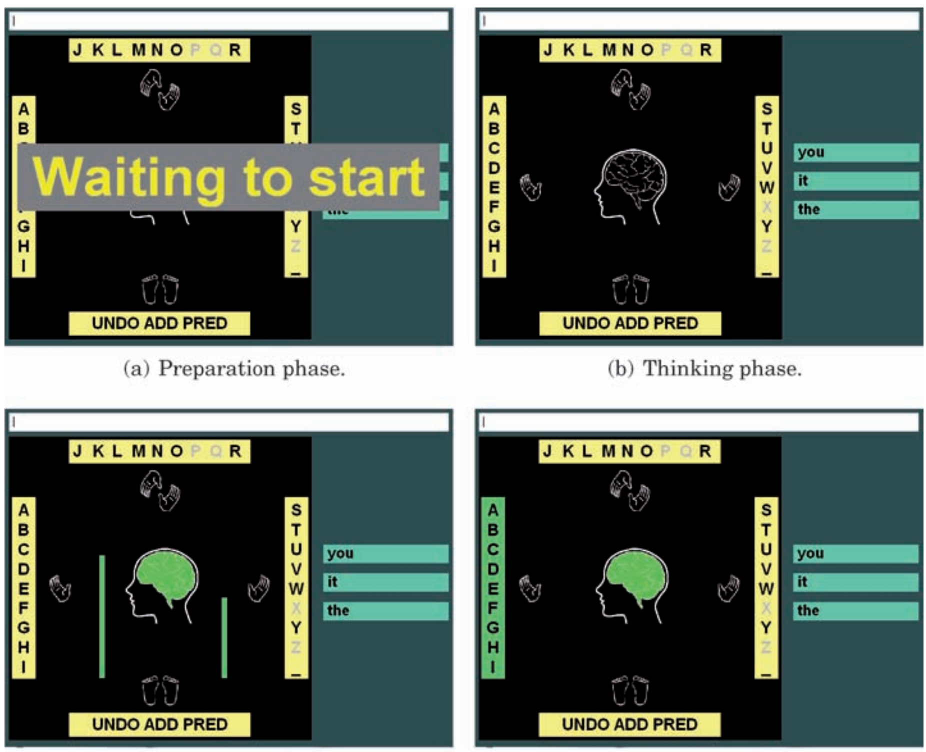


Fig. 1. The four phases of the application protocol.



(a) Preparation phase. (b) Thinking phase.

(c) Recording phase. (d) Selection phase.

Fig. 2. GUI displayed in different phases of the application protocol.

3. APPLICATION PROTOCOL

The BCI spelling application we propose in this article adopts a synchronized control paradigm in which selections are performed according to an application protocol composed of four phases (Figure 1).

Preparation. At the beginning of the preparation phase, the graphical user interface is displayed on the screen and a text message warns the user that a spelling session will start in a few seconds (Figure 2(a)). This phase lasts three seconds.

Thinking. In the thinking phase, the user can see all the options provided by the current state of the user interface and decide which target to select. Next to each target, a small icon indicating the corresponding motor-imagery task is displayed (Figure 2(b)). During this phase, the acquired EEG signal is completely discarded. This phase lasts three seconds.

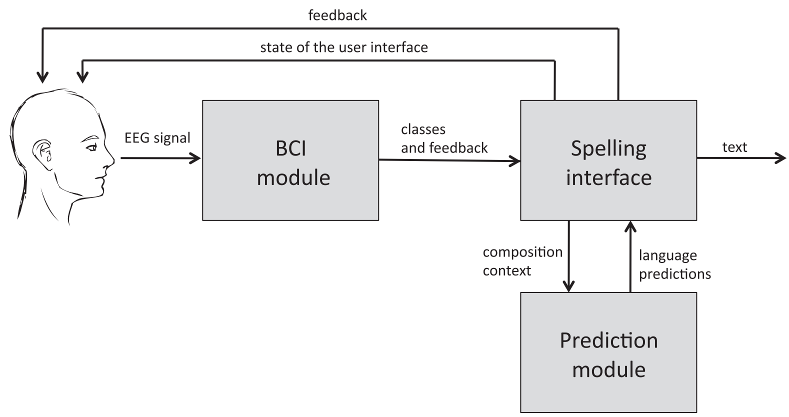


Fig. 3. System overview.

Recording. In the recording phase, the subject performs the motor-imagery task corresponding to the selected target and the EEG signal is buffered for later classification. During this phase, the brain depicted at the screen center is colored in green and two green bars are displayed at the sides (Figure 2(c)). The height of each bar, that is, a visual feedback, is proportional to the strength of the characteristic brain rhythms that the user learns to modulate during motor imagery. This phase lasts six seconds.

Selection. As soon as the recording phase terminates, the signal is classified and the associated control signal is immediately available to the spelling application. During this phase, the selected target is colored in green (Figure 2(d)), providing the user with immediate feedback about the result of the classification. This phase lasts 500 ms. At the end of this phase, the generated control signal is evaluated and the system returns in the thinking phase for a new target selection.

The time durations associated with the different phases have been chosen empirically after a set of test trials with different subjects. The time length of the recording phase is clearly a trade-off between the amount of signal available for classification and the selection speed achievable. We did not specifically investigate how performance varies according to different timings configurations; nevertheless, these parameters can be easily configured within the application, and different solutions can be adopted for different subjects.

4. SYSTEM OVERVIEW

Our BCI spelling application consists of three main modules (Figure 3).

The *BCI module* processes the raw EEG signal acquired from the amplifier, classifies different motor-imagery tasks, and provides as output a control signal for the spelling interface. Additionally, this module provides a feedback signal computed from the acquired EEG data that is displayed in real-time on the user interface (Section 5).

The *spelling interface* presents choices to the user in the form of graphical targets to be selected, that is, rectangular areas at the extreme of the interface (Figure 2(b)). A target may contain a set of letters to be expanded, a word suggestion, or an auxiliary function to be activated. The spelling interface translates the received control signal into target selections and provides as output text in natural language. This module receives from the prediction module the probabilities of symbols and words in the current composition context in order to speed up the symbol selection to provide word suggestions (Section 6).

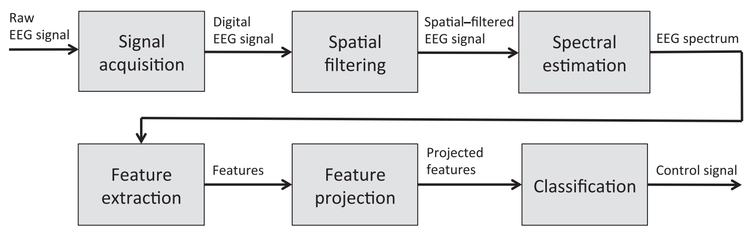


Fig. 4. BCI processing pipeline.

Finally, the *prediction module* computes probabilities of symbols and words in the current compositions context querying a statistical language model. The composition context is kept updated with messages received from the spelling interface (Section 7).

5. THE BCI MODULE

The goal of this module is to translate the EEG signal acquired from the subject's scalp into a control signal that is used to operate the spelling application. By performing this translation, the BCI paradigm adopted is motor imagery [Jeannerod 1995; Pfurtscheller and Neuper 1997]. This paradigm is based on the detection of Sensory-Motor Rhythms (SMRs), that is, characteristic oscillations of the EEG signal generated over the sensory-motor cortex and normally observed in the frequency bands 8–13Hz (*mu*) and 13–25Hz (*beta*) [Kozelka and Pedley 1990]. SMRs have been found to be attenuated during movement and preparation of movement contralaterally with respect to the moved limb (event-related desynchronization); while during post-movement and relaxation, an increase of SMRs is normally observed (event-related synchronization) [Pfurtscheller and Lopes da Silva 1999]. Furthermore, and most relevant for BCIs, the amplitude of SMRs also has been found to be modulated by a simple imagination of the motor action (motor imagery), without the need of any muscular activation [Pfurtscheller and Neuper 1997].

BCI systems employing imagined movements of hands, feet, or tongue have been developed by different research groups around the world, the most important ones being the Wolpaw's group at the Wadsworth Center (Albany, NY) and the group led by Gert Pfurtscheller in Graz (Austria). Wolpaw and his colleagues developed the first motor-imagery BCI for an EEG-based cursor control [Wolpaw et al. 1991] in which the amplitude of the SMRs was directly translated into cursor movement. Their original BCI system has been improved, combining the information extracted from different EEG channels and frequency bands and introducing an adaptive algorithm to generate control signal [McFarland et al. 1997; Wolpaw et al. 2000; Wolpaw and McFarland 2004; McFarland et al. 2008]. The group led by Gert Pfurtscheller at Technische Universität Graz developed a system for discriminating between three imagined movements (left hand, right hand, and feet) using spectral features for classification [Kalcher et al. 1996]. Their system has been later improved, adopting a feature selection algorithm for detecting subject-specific discriminative frequency components [Pregenzer and Pfurtscheller 1999] as well as introducing an adaptive classifier and an asynchronous communication protocol [Pfurtscheller et al. 2003].

In our BCI system, we consider four motor imagery tasks: the motor imagery of the left hand (LH), right hand (RH), both hands (BH), and feet (F) (for two subjects, class F has been excluded; see Section 8). Using signal processing and machine learning techniques, the BCI module classifies these four tasks from the EEG signal acquired in real-time from the subject's scalp. To this end, we adopt a BCI processing pipeline composed of six stages (Figure 4).

(1) *Signal acquisition*. The raw EEG signal is acquired from the subject's scalp, amplified, and sampled by an EEG amplifier.
(2) *Spatial filtering*. The digital EEG signal is spatially filtered, enhancing signal localization and averaging out all the background activity that is irrelevant for classification.
(3) *Spectral estimation*. The EEG signal undergoes spectral estimation.
(4) *Feature extraction*. A number of features, designed to discriminate different motor-imagery tasks, are extracted from the EEG power spectrum. Additionally, a feedback signal is estimated in real-time from the EEG data and provided to the user interface for visualization.
(5) *Feature projection*. The feature vector is projected in a lower-dimensional space.
(6) *Classification*. The projected features are classified, and the corresponding control signal is provided to the spelling interface.

In the following paragraphs, we give a brief explanation of these six processing stages (further details can be found in Online Appendix A available in the ACM Digital Library).

5.1. Signal Acquisition, Spatial Filtering, and Spectral Estimation

At the *signal acquisition* stage, the raw EEG signal is acquired from the subject's scalp, amplified, and sampled by an EEG amplifier. The EEG has been recorded using a prewired electrode cap equipped with 19 electrodes positioned in the standard locations defined by the 10-20 international system [Jasper 1958].

At the *spatial filtering* stage, three Large Laplacian filters [McFarland et al. 1997] centered on channels C3, Cz, and C4 are applied to the EEG data. These channels are particularly relevant for our BCI system since they are located in correspondence with the sensory-motor cortex where SMRs are most prominent. Spatial filtering is used to enhance signal localization, averaging out all background activity that is not relevant for classification.

At the *spectral estimation* stage, the spectral power of the EEG signal is estimated in real-time from a limited number of voltage samples. Since the EEG signal is characterized by a low signal-to-noise ratio and can be affected by abrupt variations over time, statistical spectral estimation techniques are normally employed for this purpose. In this work, we adopt the Maximum Entropy Method (MEM) [Burg 1975] for spectral estimation, as it has been shown to perform well with EEG data [Pardey et al. 1996]. The EEG power spectrum computed at this stage is used to extract the features for classification and to provide a feedback signal to the user.

5.2. Feature Extraction, Projection, and Classification

At the *feature extraction* stage, a number of features are extracted from the estimated EEG frequency spectrum in order to discriminate among the different motor-imagery tasks, and a feedback signal is generated for the user. The feedback is proportional to the EEG power estimated on channels C3 and C4 in the *mu* frequency band (see Online Appendix A.4 available in the ACM Digital Library). Dealing with feature extraction, a common practice of motor-imagery BCIs is to define features in the frequency domain since SMRs are characterized by strong oscillatory components. However, there is not a clear consensus on what are the specific signal features that best discriminate between different motor-imagery tasks, that is, which EEG channels, time intervals, or frequency windows perform best [Lotte et al. 2007]. In this work, we explore a novel set of features covering a broad spectrum of possible feature choices and apply a feature selection algorithm to select the best discriminative ones. We consider two types of features: *average features* and *evolution features*. Average features are computed from the EEG power spectrum time-averaged across the whole duration of the recording phase (see Section 3), and they are designed to represent the average form of the EEG power spectrum while a motor-imagery task is performed. Evolution features, instead, are computed from six EEG power spectra (one for each second of the recording phase), and they represent the temporal evolution of the signal during the task. A detailed description of the features can be found in Online Appendix A.4 available in the ACM Digital Library.

We identified a total of 255 features and applied a feature selection algorithm to select the most discriminative ones. Feature selection has been performed exclusively offline, and only the selected features have been used for online classification. Since the space of all the possible feature subsets is huge, we performed a heuristic search using a genetic algorithm (GA) [Goldberg 1989; Holland 1992] (see Online Appendix A.4.3 available in the ACM Digital Library for further details).

Before proceeding with classification, at the *feature projection* stage, all the features retained by the selection algorithm are projected in $N - 1$ dimensions, where N is the number of classes considered. This operation is performed by applying Fisher Discriminant Analysis (FDA) [McLachlan 2004]. Finally, the *classification* stage applies linear discriminant analysis (LDA) [Bishop et al. 2006] for classification.

6. THE SPELLING INTERFACE

The model of the user interface plays a very important role in the design of the overall spelling application. Indeed, the way choices are presented to the user influences both the efficiency and usability of the system [Dal Seno et al. 2010b]. The main functional requirements we considered in the design of the user interface are summarized next.

—Enable, the composition of plain text in the English language by means of a control signal with three or four different states.
—Provide a number of auxiliary functions that may be activated during composition, for example, speech synthesis, word deletion, etc.
—Deal with errors introduced by incorrect BCI classifications or by wrong selections performed by the user.

Moreover, some important non-functional requirements have been taken into account, too.

—*Usability*. The interface shall be intuitive and predictable. Solutions that may disorient or confuse the user shall be avoided, while repetitive schemes and patterns that are easily recognizable by the user shall be preferred.
—*Formalization*. The interface structure and behavior shall rely on a formal model so that different versions of the user interface can be easily tested and interface performance can be assessed by automatic simulator programs.
—*Predictive capabilities*. The interface shall exploit redundancies in natural language in order to speed up the composition process.

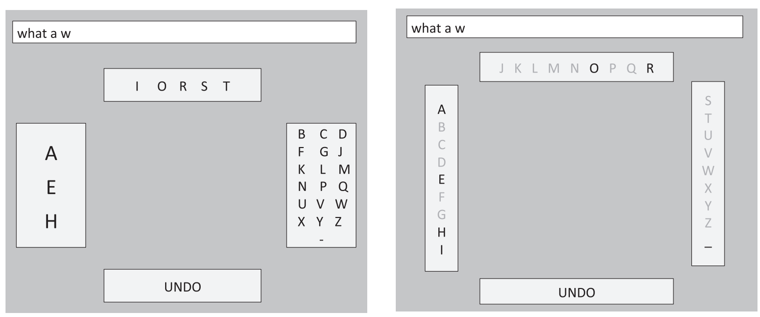
In the following sections, we describe in detail the design of the spelling interface implemented in our system.

6.1. Symbols and Functions

We consider a set of symbols composed by the 26 letters of the English alphabet plus the spacing character. In order to keep the interface simple, we chose not to include any punctuation character in our application. By contrast, we considered the possibility of writing numbers, as they may prove important even in the context of simple conversations. However, numbers are kept separate from letters in the user interface, and they

Table I. Auxiliary Functions for the Spelling Interface

Undo	Recovers the previous state of the interface after an erroneous selection.
Exit menu	Exits from a menu.
Predictions	Allows the selection of an entire word from a set of predictions.
Delete char	Deletes the latest character inserted.
Delete word	Deletes the latest word inserted.
Speak	Activates the vocal synthesizer to reproduce the currently composed text.
Add to dictionary	Enables all symbols that may have been disabled as a result of letter prediction.
Numbers	Switches the symbol set from letters to numbers.
Letters	Switches the symbol set from numbers to letters.
Quit	Terminates the application.



(a) Dynamic symbol arrangement. (b) Static symbol arrangement.

Fig. 5. Examples of two possible interfaces adopting dynamic and static symbol arrangements. In the example shown, the user wants to write the phrase "what a wonderful day" and has already entered the string "what a w". The same letter prediction algorithm is assumed in both cases. The letters associated with nonzero probabilities for the context considered are (in order of probability) "a", "e", "h", "i", "o", "r". The dynamic interface (on the left) groups the 27 symbols, assigning to smaller groups the symbols with higher probabilities. By contrast, the static interface (on the right) shows balanced groups and disables symbols with zero probability. In both cases, selecting the letter "o" requires two group expansions.

are reachable through a dedicated menu. This choice allowed us to minimize the number of selections required for normal text composition while retaining the possibility of entering numbers. The set of auxiliary functions we considered is reported in Table I.

6.2. Symbol Selection Strategy

In our application, symbols and functions are grouped in a hierarchical tree of targets, and the selection of one symbol is performed through recursive target expansions. This can be regarded as an encoding technique, as each symbol is encoded in the sequence of targets that needs to be expanded for its selection.

Since in natural language each symbol has a different probability of being selected, it would be advantageous to adopt an encoding scheme such that the average number of target expansions required during text composition is minimized. With respect to the user interface, this would result in having targets containing different numbers of symbols where the most probable ones are associated with the targets with the lowest cardinalities. However, the probability of each symbol being selected changes according to the state of the composition (i.e., the letters previously inserted), and rearranging all the symbols in their optimal targets could be disorienting for the final user (Figure 5(a)). There is indeed a trade-off between the optimal encoding scheme and the usability of the system.

The solution we adopted is to keep all symbols fixed in the same targets while disabling or enabling them according to their probability in the current composition context (Figure 5(b)). In this way, a target containing just one enabled symbol does not require further expansions. With this method, it is possible to combine the requirement of improving the communication rate through letter prediction while keeping the user interface simple and cognitively easy to manage. Further details about the algorithm used to enable/disable letters are given in Section 7.

6.3. Handling Errors

An important requirement in the design of the user interface is the capability of handling errors in an efficient way. We identified three types of errors that may occur while using the spelling application.

—*BCI errors*. These errors are caused by a wrong classification of the EEG signal and are the most likely to happen. The BCI module, indeed, provides classification accuracies that are often quite far from optimal. This is mainly because the EEG signal is characterized by a poor signal-to-noise ratio and is strongly influenced by the presence of artifacts. Moreover, a decreasing level of concentration during the usage of the application or feelings such as frustration and edginess may influence the performance of the brain-computer interface.

—*Errors induced by the user interface*. These errors occur when the user fails to interpret the current interface state of the speller. Errors of this type are less common than BCI errors but still present, especially if the user has not much experience with the spelling application. For instance, a user may believe to have successfully entered a symbol while, in reality, a final selection was still missing or may be unsure about the interface state that will follow an "undo" operation.

—*User errors*. These errors are due to a change of mind of the user about the text that was intended to be written. These errors are less common than the others since the user usually thinks carefully about what to write before starting the spelling session. The timings imposed by the BCI are indeed quite strict, and formulating or changing a phrase during composition may prove to be a hard task.

Errors belonging to the first two categories can be regarded as similar since they are immediately recognized by the user as soon as they occur. In order to correct these errors efficiently, there always should be a fast way of reverting the last selection performed, thereby restoring the interface state immediately before the error. Therefore, in our application, we ensure that an "undo" or an "exit menu" function is always reachable with at most two target selections (Section 6.4).

Errors due to a change of mind of the user are, instead, quite different. Indeed, since the selection of one letter requires on average two or three target selections, changing an entire word means reverting to an interface state that may be many steps behind. Obviously, it is not worth going through all the intermediate states step by step, and a better way of handling these errors would be to activate auxiliary functions. Therefore, in our application, we provide functions for deleting a character and for deleting an entire word directly (Table I). However, since this kind of error is assumed to be rather rare, the functions "delete char" or "delete word" are not available at each interface state, but still reachable through the corresponding menu of auxiliary functions (Figure 6).

6.4. Modeling the User Interface

The formalism with which we chose to model the user interface structure and behavior is pushdown automaton (PDA) [Hopcroft et al. 2007]. Pushdown automata are an extension of finite state machines (FSM) in which, along with input symbols, states, and transitions, a stack of symbols that can be used in defining the machine behavior is also provided. In the interface models designed for this work, a state corresponds to a set of choices proposed to the user, while the stack is used to keep in memory the

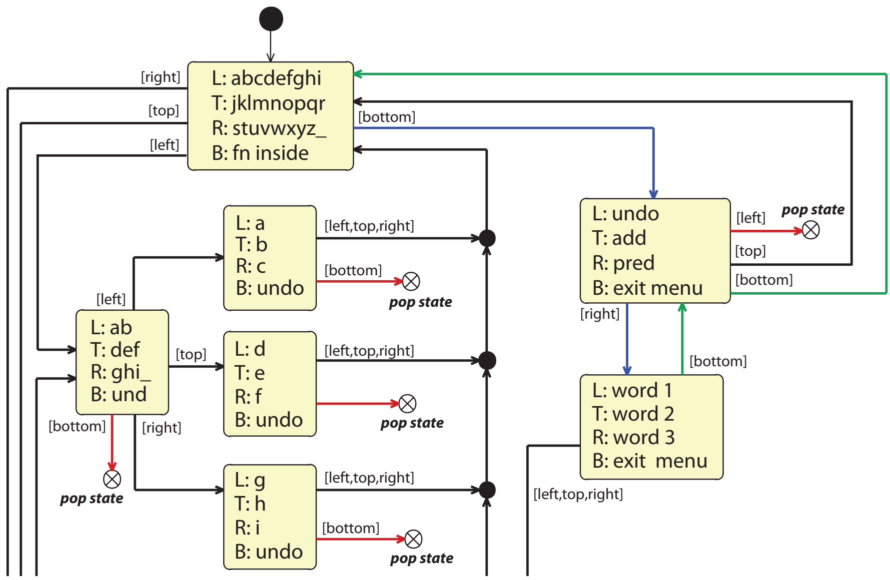


Fig. 6. Excerpt of a user interface model with four targets. Each state is labeled with the contents of its targets located at position left (L), top (T), right (R), and bottom (B). *Action*-transitions are depicted in black, *undo*-transitions in red, *menu*-transitions in blue and transitions of type *exit-menu* in green.

history of past states. Using a stack is indeed easy to model the “undo” operations that are required for handling errors.

A state is identified by a set of targets, each displaying a different choice to be selected. One target may be associated with a set of symbols to be expanded or with a function to be activated. Considering four available choices, each state has thus four targets and four transitions departing from it. Each transition determines which is the next state to display when the corresponding target is selected. Additionally, in order to ensure the consistency of a user interface model, some constraints have been defined on states and transitions. These constraints specify which states are admissible following a specific type of transition. Transitions may be indeed organized into four main categories.

—*Action transitions*. Transitions that, if taken, result in a modification of the internal state of the application. These are the most common transitions, and they are associated, for example, to the expansion of a set of symbols, the selection of a suggested word, the deletion, of a word, etc. When a transition of type *action* is taken, the source state is pushed to the top of the stack.
—*Undo transitions*. Transitions whose goal is to recover the interface state that was immediately before the last *action*-transition. Therefore, when a transition of this type is taken, the next state is popped from the top of the stack.
—*Menu transitions*. Transitions associated exclusively to menu expansions and that do not modify the internal state of the application. A transition of type *menu* is associated, for example, to the target “predictions”, which brings to an interface state displaying a set of suggested words. Transitions of this type do not read or modify the content of the stack.
—*Exit menu transitions*. Transitions used to restore the interface state from where the last *menu*-transition was taken. These transitions define explicitly their end point and do not read or modify the content of the stack.

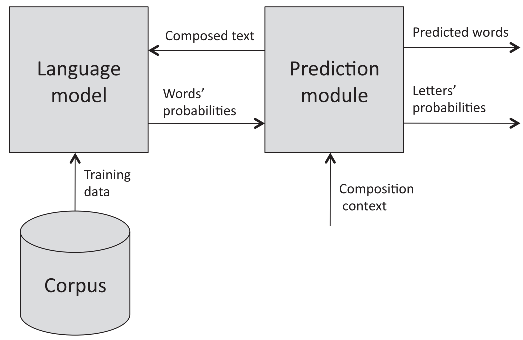


Fig. 7. Prediction module.

The consistency constraints characterizing a valid user interface model are the following.

—*Action*-transitions shall bring to states in which an *undo*-transition is directly available or is available in a directly reachable state.
—*Menu*-transitions shall bring to states in which a transition of type *exit-menu* is directly available.

These constraints ensure that when a transition is taken by mistake, it is possible to recover the error with one or at most two further selections. As discussed in Section 6.3, this decision is motivated by the fact that the control signal provided by the BCI is not reliable and the most likely errors are due to wrong classifications of the EEG signal.

We designed two versions of the user interface: one controlled by a three-state control signal (displaying three targets at a time) and one controlled by a four-state control signal (displaying four targets at a time). In this section, we briefly introduce the user interface model with four targets. Figure 6 shows an excerpt of this user interface model; the complete description is provided in the Appendix. The left, top, and right targets in the starting state contain nine letters each; if, say, the user selects the left target, the next state subdivides the "a b c d e f g h i" letters into three targets, as shown. Then, if the user selects the top target, the letters "d e f" are subdivided into three targets; a further selection, say the right target, inserts the letter in the text ("c" in this example), and the interface returns to the initial state. Whenever the user selects the target corresponding to the "undo" function, the interface state goes one step back and displays the previous targets. In this model, the targets "add to dictionary", and "predictions" are also available through the target "functions" in the starting state. Selecting the target "add to dictionary", the interface enables all symbols that may have been disabled because of letter prediction, while selecting the target "predictions", the interface displays three predicted words that are selectable through the corresponding targets.

7. THE PREDICTION MODULE

The goal of this module is to provide language predictions to the spelling application. The module is informed by the speller interface about the current composition context and queries a statistical language model to estimate both word and letter probabilities (Figure 7).

The language model is trained by computing statistics on a corpus of texts. As the training corpus, we selected a subset of the British National Corpus (BNC) [Leech 1992]. The BNC is a collection of more than 100 million words sampled from a large range of sources chosen to be representative of a wide cross-section of British English, both spoken and written. From the BNC, we extracted only spoken text, since we argued that spoken language could be considered closer to the type of communication we are dealing with (simple sentences and simple vocabulary). In particular, we considered only texts with a respondent age ranging from 15 to 44 years and with a spoken context that is either educational-informative or leisure. According to these criteria, we extracted 107 training texts with a total number of $\sim 3.3 \cdot 10^6$ word occurrences.

Word predictions and letter probabilities are computed by evaluating the current composition context. The composition context is constantly kept updated with messages received from the speller interface and provides information about (1) the words entered before the current word being composed, (2) the prefix of the current word, and (3) the set of letters that are selectable from the current state of the interface.

Both word predictions and letter probabilities are computed starting from a list of candidate words extracted from the vocabulary. This list contains all words that match up with the current prefix such that the first letter after the prefix is selectable from the speller interface. For example, if the current prefix is “w” and only the letters “a”, “e”, “h”, “i” are selectable from the interface, the word “well” is a possible candidate, but the word “wonderful” is not.

For each candidate word, the corresponding probability is estimated by means of a 3-gram—a statistical language model in which the probability of the current word depends on the two previous words. In particular, we adopted a Katz backoff 3-gram model [Katz 1987; Jurafsky and Martin 2000] and applied Good Turing smoothing [Gale 1995] for low counts. The predicted words provided as output are the N words with the highest probabilities, being $N = 3$ for the four-targets interface and $N = 2$ for the three-targets interface.

The computation of letter probabilities relies on word probabilities, too: with $l_0, l_1, \ldots, l_{n-1}$ being the prefix entered for the current word and w_{k-2}, w_{k-1} being the two preceding word tokens, the probability of the next letter being l_n is computed as the following.

$$P(l_n|l_0, l_1, \ldots, l_{n-1}, w_{k-2}, w_{k-1}) = \frac{\sum_{w_k \in W^n} P(w_k|w_{k-2}, w_{k-1})}{\sum_{w_k \in W^{n-1}} P(w_k|w_{k-2}, w_{k-1})}, \tag{1}$$

where W^n is the set of words with prefix $l_0, l_1, \ldots, l_{n-1}, l_n$. That is, we sum up the probabilities of all words having l_n after the entered prefix, and we normalize by the total probability of all candidate words. Letter probabilities are used in order to enable or disable letters in the speller interface (all letters with nonzero probabilities were enabled in our experiments).

8. RESULTS

This section describes all the experiments and tests performed for this work along with a discussion on the results obtained. Sections 8.1, 8.2, and 8.3 deal with tests performed, respectively, on the BCI module, the speller interface, and the overall system.

8.1. Experiments and Tests with the BCI

BCI experiments have been performed at the *AirLab*, the Artificial Intelligence & Robotics Laboratory at Politecnico di Milano, from September 2008 to June 2009. Three subjects without motor disorders participated in our BCI experiments. The participants were all students at Politecnico di Milano of ages 24 to 26, male, and right handed. None of them had previous experiences with any other BCI system. A left-handed subject was excluded from our experiments since the first offline analyses revealed that he would not be able to achieve BCI control. We refer to subjects by the first letter of their names—P, T, and F. The experiments performed with the BCI module

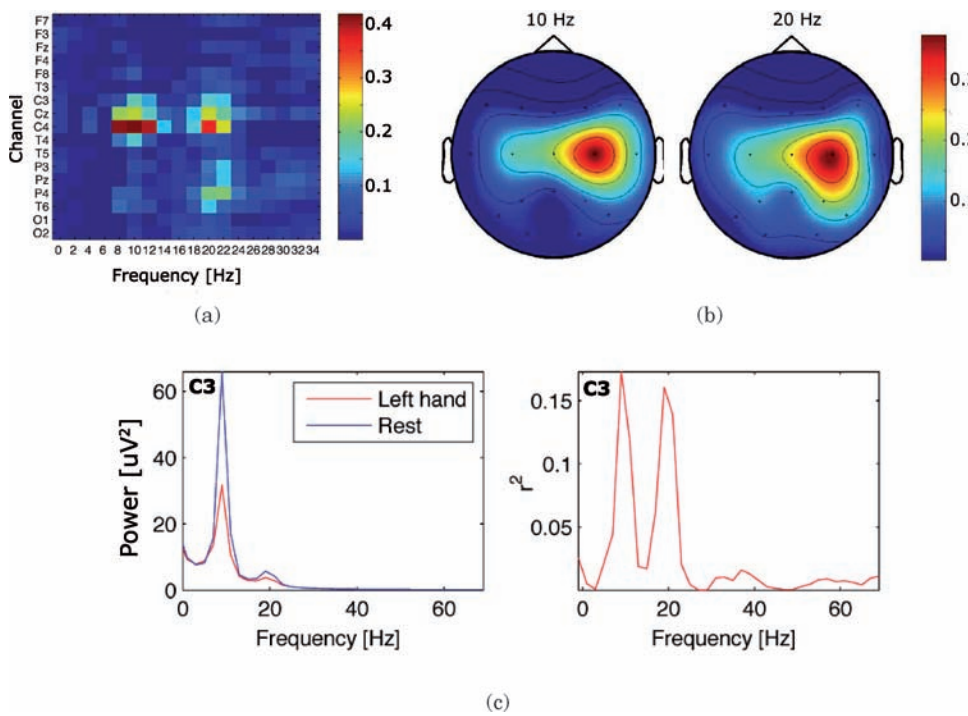


Fig. 8. Offline analyses plots for subject P and the condition "Left hand Vs Rest". (a) Coefficient r^2 as a function of frequency and channel (color coded). (b) Scalp topography distribution of r^2 at 10Hz (mu rhythm) and 20Hz (beta rhythm). (c) Signal power (left) and r^2 coefficient (right) for channel C3.

include offline analyses, offline classification, and online classification. In the following, such experiments are briefly discussed; an in-depth discussion can be found in Online Appendix B available in the ACM Digital Library.

8.1.1. Offline Analyses. We started our experiments with a set of acquisition sessions (*initial sessions*) whose goal was to perform offline analyses of EEG data and train the supervised algorithms implemented. Here, we focused on the BCI module alone; hence, we did not use the spelling interface introduced in Section 6. Instead, we adopted a minimal user interface in which different visual stimuli were subsequently presented in the center of a blank screen and each stimulus was associated with a motor-imagery task to be performed by the user (see Online Appendix B.1 available in the ACM Digital Library for further details).

With these analyses, we wanted to characterize the EEG patterns induced by different motor-imagery tasks and identify the most discriminative channels and frequency bands for each subject. Discrimination levels have been quantified using the r^2 coefficient, also known as the coefficient of determination [Wonnacott and Wonnacott 1985]. Values of r^2 close to 1 reveal very good discrimination between conditions, while r^2 values close to 0 are associated with conditions that are hardly distinguishable. For each channel, we computed the r^2 coefficient between the EEG power acquired during a motor-imagery task and the EEG power acquired during rest (no motor imagery).

Three categories of plots have been generated for each subject: plots displaying r^2 as a function of frequency and channel, plots displaying the topographical scalp distribution of r^2 for the mu/beta rhythms, and plots displaying how signal power and r^2 values are distributed over frequencies for selected channels. Figure 8 shows examples of these plots for subject P.

Table II. Offline Classification Performance (Avg and Std Dev)

# classes	Bit-rate	K	Overall	Left hand	Right hand	Both hands	Both feet
Subject P							
4	1.158	0.797	0.848	0.810	0.890	0.760	0.937
	0.193	0.063	0.047	0.050	0.050	0.049	0.047
3	1.244	0.920	0.946	0.948	0.947	—	0.945
	0.161	0.048	0.032	0.036	0.022	—	0.040
Subject T							
4	0.353	0.439	0.580	0.500	0.566	0.411	0.848
	0.034	0.023	0.017	0.028	0.030	0.039	0.026
3	0.485	0.602	0.734	0.708	0.663	—	0.834
	0.026	0.016	0.011	0.041	0.006	—	0.009
Subject F							
4	0.195	0.317	0.488	0.412	0.450	0.342	0.736
	0.072	0.061	0.046	0.066	0.067	0.034	0.059
3	0.295	0.457	0.638	0.556	0.588	—	0.772
	0.138	0.105	0.070	0.068	0.077	—	0.066

The results obtained with offline analyses have been used to design and parametrize the features used by the classifier. Even though we found similar EEG patterns across subjects, offline analyses revealed strong subject-to-subject variability. In particular, the EEG spectra of subject P showed more prominent sensory-motor rhythms compared to subjects T and F, and the motor imagery of both feet was associated with different EEG patterns for subject T compared to subjects P and F (see Online Appendix B.1 available in the ACM Digital Library for details). The highest r^2 values have been found in the frequency bands 8–12 Hz and 18–22 Hz for all subjects; therefore, these values have been taken as band limits for computing features. Moreover, mu and beta rhythms were mostly located at the centers of these bands, being, respectively, 10Hz for the mu rhythm and 20Hz for the beta rhythm.

8.1.2. Offline Classification. Data acquired during *initial sessions* also have been used to estimate classification performance offline. In evaluating offline classification performance, we adopted a five-fold cross-validation scheme in which, at each iteration, all data acquired in one day are used for testing, and all remaining data are used for training. Indeed, signals recorded in exactly the same settings are likely to be correlated and may lead to overfitting. Classification performance was evaluated computing the overall classification accuracy (percent correct), the class-specific classification accuracy, the Cohen's Kappa coefficient [Cohen 1960], and the theoretical information transfer rate (in bits per trial) [Pierce 1980].

For all subjects, offline classification performance was evaluated considering both a four-class and a three-class classification problem, these being the two options available for the spelling application designed. Dealing with the four-class problem, we considered one class for each motor-imagery task (i.e., left hand, right hand, both hands, and both feet). Dealing with the three-class problem, instead, the three classes yielding to the best classification performance have been selected. For all subjects, the best three classes were left hand, right hand, and both feet motor imagery. According to the results obtained, we evaluated the best number of classes to be considered for each subject. Table II summarizes the results obtained (see Online Appendix B.2 available in the ACM Digital Library for detailed results).

As expected from offline analyses, subject P is the one for which the best classification performance has been reached. Considering four classes, we obtained for subject P an overall accuracy of 0.848, a Kappa coefficient of 0.797, and an information transfer rate of 1.158 bits per trial. Additionally, it should be noted that class "both hands" is the one with the lowest accuracy, while class "both feet" is the one that is best

Table III. Online Classification Performance (Avg and Std Dev)

Bit-rate	K	Overall	Left hand	Right hand	Both hands	Both feet
			Subject P (4 classes)			
0.975	0.736	0.802	0.754	0.832	0.721	0.897
0.137	0.052	0.039	0.032	0.032	0.045	0.051
			Subject T (3 classes)			
0.462	0.586	0.724	0.663	0.643	0.860	—
0.070	0.044	0.029	0.025	0.056	0.049	—
			Subject F (3 classes)			
0.363	0.518	0.679	0.636	0.710	0.689	—
0.083	0.060	0.040	0.044	0.008	0.078	—

discriminated from the others. Considering just three classes and excluding the class "both hands", classification performance improves, reaching in this case an overall accuracy of 0.946. However, the information transfer rate does not increase considerably since the number of classes is lower. Therefore, looking at the BCI performance alone, these two configurations can be regarded as equivalent for subject P.

Subjects T and F, instead, present classification performances that are much lower compared to those of subject P. Indeed, with four classes, we obtained an overall classification accuracy 0.580 for subject T and 0.488 for subject F. Even in this case, the class "both hands" is the most problematic. Excluding this class, performance improves, reaching an overall classification accuracy of 0.734 for subject T. Comparing bit-rate values and Kappa coefficients in the two settings, there are no doubts that three classes are better than four for subject T. Similar considerations are also valid for subject F. However, the overall classification accuracy estimated offline remains rather low for this subject.

8.1.3. Online Classification. The BCI module was also tested online with the three subjects. The classifier was trained with the data acquired in the initial sessions considering only the best subset of classes for each subject according to the results obtained offline, that is, four classes for subject P and three classes for subjects T and F. The user interface and the acquisition protocol were the same as in the initial sessions with the only difference, being that the predicted class was provided at the end of each trial, and a visual feedback was also displayed during the task (two green vertical bars similar to the ones used in the spelling interface). The results obtained are summarized in Table III.

For subjects P and T, the average classification accuracy obtained online is slightly lower compared to the results obtained offline, while for subject F, performance improved. From these tests, it is not clear which is the actual effect of feedback on performance, but it seems that at least subject F got benefits from it.

8.2. Testing the Speller Interface

In order to evaluate the expected performance of the BCI spelling application, we implemented a simulator of the spelling interface. The reason for performing simulations is that BCI experiments are very time consuming and wearing for the participants. Moreover, since different versions of the speller interface were designed, we wanted to assess the best configuration for each subject before performing online experiments. The interface simulator takes as input a text to be composed, a spelling interface model, and a confusion matrix. The program simulates the composition of the given text, performing target selections on the specified interface model in a way that is similar to what a real user would do. Errors that may be introduced by the BCI are simulated, too: the probability of selecting each target, given the target planned by the simulator, is estimated from a confusion matrix provided as input. To our knowledge, this is the

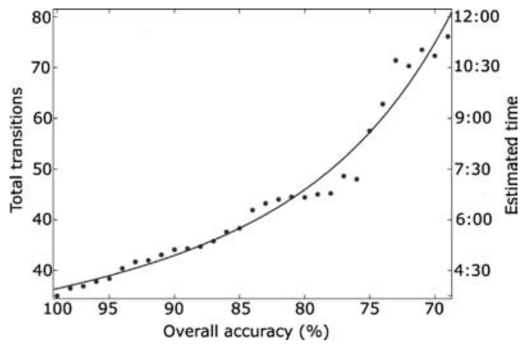


Fig. 9. Impact of classification accuracy on performance. Each sample (point marker) indicates the average number of transitions (and the corresponding estimated time) obtained, simulating the task ten times with different randomization seeds. The solid line is an exponential fit to the data.

first study in which a simulator program like ours is used for testing BCI-controlled user interfaces; thus, the approach proposed in this article shall be regarded as novel.

8.2.1. Impact of Classification Accuracy on Performance. In order to assess how classification accuracy impacts system performance, the composition of a simple phrase has been simulated with different accuracy levels. The sentence chosen for all simulations performed and used also for online tests is "what a wonderful day". The selected words are representative of the mean length of words in English, and four out of five are among the 200 most common English words [Leech et al. 2001]. Using a single sentence for testing allows for better comparison among different experimental runs; however, possible biases deriving from the specific sentence selected remain to be evaluated.

For these simulations, we synthetized confusion matrices generated with increasing levels of overall accuracy. Balanced matrices have been considered, that is, the off-diagonal elements of each row are computed evenly, distributing the error probability over all classes. We used the interface model with four targets and enabled language predictions. The speller performance has been measured by counting the total number of interface transitions required to complete the task.

Figure 9 shows the results of these simulations. We notice that the number of transitions required increases exponentially as the accuracy level decreases, with 25 being transitions the minimum number required with 100% accuracy and about 80 being the total transitions required with 70% accuracy. This exponential trend is explained by the fact that each error needs to be corrected with new transitions that in turn may be sources of other errors.

8.2.2. Impact of Language Predictions on Performance. In order to assess how language predictions impact system performance, the composition of the same phrase was simulated enabling and disabling the predictive capabilities. For these simulations, we used the interface model with four targets and the confusion matrix of subject P estimated after offline classification (Table XI(c) in Online Appendix B available in the ACM Digital Library). The prediction modes considered are the following.

—*Word and letter*. All predictive capabilities are available in the interface. Hence word suggestions are provided, and letters are activated according to their probability in the composition context.
—*Only word*. Only word prediction is available.
—*Only letter*. Word predictions are not available while unlikely letters are disabled in the interface.
—*None*. None of the predictive capabilities are available in the interface.

Table IV. Simulated Performance of the Spelling Application with Different Prediction Modes

Pred. mode	Est. time (min.)	Transitions	Errors	# Shortcuts
Word and letter	6:09	41.18 (7.84)	7.18 (3.32)	2.00 (0.00)
Only word	6:27	43.03 (7.70)	7.52 (3.55)	0.00 (0.00)
Only letter	10:38	70.92 (11.09)	13.06 (4.84)	12.00 (0.00)
None	12:56	86.26 (10.62)	15.70 (5.84)	0.00 (0.00)

Note: All values are averaged over 50 runs performed with different randomization seeds. Standard deviations are reported in brackets.

Table V. Simulated Performance of the Spelling Application with Different Versions of the Interface

# Targets	Est. time	Transitions	Errors	Shortcuts	Completed
			Subject P		
4	6:09	41.18 (7.84)	7.18 (3.32)	2.06 (0.20)	50/50
3	6:21	42.36 (4.25)	2.9 (1.95)	5.01 (0.20)	50/50
			Subject T		
4	22:03	147.15 (55.81)	53.1 (23.10)	2.45 (1.06)	40/50
3	13:12	88.78 (17.78)	22.56 (7.50)	5.48 (0.76)	50/50
			Subject F		
4	48:27	323.50 (39.80)	131.50 (60.67)	2.22 (0.43)	18/50
3	27:20	182.58 (48.51)	64.74 (21.11)	5.96 (1.26)	50/50

Note: All values are averaged over the number of completed runs, standard deviations are reported in brackets.

Table IV shows the results obtained with these simulations. For each case considered, the estimated time for the task, the total number of transitions, the number of errors, and the total number of shortcuts taken are reported. A shortcut is taken when one or more transitions are skipped because of the presence of disabled symbols in one target.

The results obtained reveal that word prediction significantly impacts the overall performance: the estimated time, with this feature enabled, requires about half of the time required without word prediction. Indeed, all the words in the phrase "what a wonderful day", except for the article "a", have been completed with a word suggestion after the composition of the first one or two letters. The impact of letter prediction on performance is instead related to the presence of word prediction. Indeed, the probability of having a shortcut increases with the number of letters entered for a given word. Therefore, if word prediction is disabled and all letters need to be entered one at a time, the impact of letter prediction is more evident. Table IV shows that when word prediction is enabled, only two shortcuts are taken for the considered task, while when disabling word prediction, twelve shortcuts are taken, yielding a significant performance improvement with respect to the base case.

8.2.3. Interface Simulations Targeted on Subjects. In order to assess whether the BCI performance obtained with the three subjects was sufficient for an effective use of the spelling application, simulations targeted on the subjects were also performed. For this purpose, the composition of the phrase "what a wonderful day" has been simulated using the confusion matrices computed offline for each subject and considering both the four-targets and the three-targets versions of the speller interface. Table V shows the results obtained (the column "completed" indicates how many simulations terminated successfully; in the other cases, a wrong selection of the "quit" target occurred).

Simulations on subject P revealed good results with both three and four classes. Even if the three-class solution was associated with a higher bit-rate for this subject, simulations show that the option with four classes yields to slightly better results with the spelling application. Dealing with subject T, instead, the solution with three classes is clearly the best among the three. This is not surprising since the bit-rate achieved with three classes was higher for this subject. In this configuration, the estimated error

Table VI. Simulated Performance of the Spelling Application Using the Best Interface Configuration for Each Subject and the Confusion Matrices Obtained in Their Latest Online BCI Session

Subject	Est. time	Transitions	Errors	Shortcuts	Completed
P	6:36	43.94(10.43)	8.74(4.35)	2.00 (0.00)	50/50
T	15:00	100.22(25.99)	26.90(10.82)	6.18 (1.87)	50/50
F	24:00	159.68(77.96)	55.16(34.82)	5.60 (1.08)	50/50

Note: All values are averaged over the number of completed runs, standard deviations are reported in brackets.

rate is about 25%, and the expected time for completing the task is around 13 minutes. Simulations on subject F reveal that this user might have some difficulties in using the spelling application effectively. Indeed, in the best configuration, the expected time for completing the task is around 27 minutes. However, the latest BCI sessions performed with feedback revealed a trend of performance improvement for this subject.

Finally, we simulated the task using the best interface configuration for each subject and employing the confusion matrices obtained online in the latest BCI session performed (with feedback). The results of these simulations are reported in Table VI. Comparing these values with the results obtained before, we note a slight performance degradation for subjects P (−6.7%) and T (−12.8%) but a performance improvement for subject F (+12.54%).

8.3. Online Testing of the Whole System

Finally, the whole BCI spelling application has been tested online. A test session consisted in performing seven repetitions of the same task, that is, the composition of the phrase “what a wonderful day”. Subject P performed three test sessions on different dates, while for subjects T and F a single test session was available. For each repetition, the following values have been registered: the total time required to complete the task, the total number of interface transitions performed, and the number of errors occurred. Subject P used the speller interface with four targets, while subjects T and F used the speller interface with three targets. The results obtained are reported in Table VII.

The average number of interface transitions required by subject P to complete the task is slightly higher compared to the simulations (performance degraded by −9.4%). This could be the effect of adopting the speller interface that is substantially different with respect to the ones used during initial sessions and feedback sessions. There is also a considerable variance in the performance registered in the three sessions, revealing that BCI control is strongly related to the psychophysical conditions met by the subject. In the best case, subject P succeeded in completing the task in 5 minutes and 24 seconds, while in the worst case it took 9 minutes and 27 seconds. The average time required for subject P is about seven minutes, corresponding to an average communication rate of 3 char/min.

Subjects T and F, on the other hand, obtained online performances that are considerably better compared to the simulations (performance improves by +27.1% for subject T and by +67.8% for subject F). Such results show that as the number of feedback sessions increases, good BCI control can be achieved by these subjects, too. Subject T took on average less than 11 minutes to complete the task, corresponding to a communication rate of about 2 char/min. Subject F, instead, required on average 7 minutes and 40 seconds to complete the task, corresponding to a communication rate of 2.7 char/min.

9. DISCUSSION

The objective of this work was to implement a BCI spelling application based on motor imagery for people with severe motor impairments, combining state-of-the-art BCI and NLP techniques to maximize the overall communication rate of the system.

Table VII. Online Performance of the Spelling Application for Subjects P, T, and F.

(a) Subject P

	Session 1			Session 2			Session 3		
Run	Time	Trans.	Errors	Time	Trans.	Errors	Time	Trans.	Errors
1	5:33	37	7	7:12	48	11	8:06	54	13
2	8:33	57	14	7:03	47	10	6:45	45	7
3	6:27	43	7	5:51	39	8	8:33	57	14
4	8:06	54	13	7:48	52	12	8:06	54	12
5	7:12	48	11	6:45	45	7	9:27	63	15
6	9:09	61	16	6:18	42	7	6:36	44	7
7	6:27	43	7	6:18	42	6	5:24	36	6
Mean	7:21	49	10.7	6:45	45	8.7	7:33	50.42	10.57
Std dev	1:18	8.6	3.7	0:40	4.4	2.2	1:22	9.18	3.8

(b) Subject T

Run	Time	Trans.	Errors
1	11:33	77	16
2	09:45	65	13
3	13:24	96	26
4	09:36	64	13
5	12:27	83	18
6	08:42	58	9
7	10:12	68	14
Mean	10:48	73.00	15.57
Std dev	01:42	13.16	5.38

(c) Subject F

Run	Time	Trans.	Errors
1	6:45	45	4
2	9:09	61	10
3	6:54	46	5
4	6:27	43	3
5	7:03	47	4
6	6:27	43	3
7	11:15	75	17
Mean	7:42	51.42	6.57
Std dev	1:48	12.19	5.20

Note: The task considered is the composition of the phrase "what a wonderful day".

The results obtained are satisfactory and comparable with the latest works on motor-imagery BCI spellers reported in the literature [Blankertz et al. 2007]. With subject P, we obtained high classification accuracies both offline and online, considering both the four-class and the three-class scenarios. This subject was able to use the spelling application to compose simple sentences, and the overall communication rate achieved was about 3 char/min. With subjects T and F, we initially obtained lower offline classification accuracies, but significant improvements were obtained as the number of feedback sessions increased. Subjects T and F achieved, respectively, an average communication rate of 2 and 2.7 char/min.

Significant performance variability has been observed across different acquisition sessions and different subjects. The problem of variability is one of the most challenging issues in BCI research, especially dealing with the paradigm of motor imagery. This is because the neurophysiological phenomena considered are self-induced, and the response of sensory-motor rhythms is strongly affected by the psychophysical conditions of the users. Previous studies [Neuper et al. 1999] showed that performing an adequate number of training sessions with feedback is essential to improving the performance of BCI systems based on motor imagery. An improvement trend has been observed in our tests, too (especially for subjects T and F), and we believe that with more feedback sessions, even better results could be achieved.

Regarding the BCI module, there is certainly room for further improvements. In order to cope with the problem of variability, the most promising approach is to use features in the spatial domain, specifically the method of Common Spatial Patterns (CSP) [Fukunaga 1990; Koles 1991; Blankertz et al. 2008]. This technique has been adopted with success in several recent works [Yan et al. 2008; Wang et al. 2004; Ramoser et al. 2000] and is one of the best performing techniques for motor imagery.

Alternative symbol arrangements in the speller interface could be investigated in order to speed up the selection process and to improve the overall communication rate. For example, symbols may be rearranged in order to minimize the number of selections while keeping the interface simple and intuitive. Dealing with the speller interface, new ways for exploiting language predictions should be investigated, for example, the possibility of introducing a bias in the classifier based on the probabilities of the symbols contained in each target.

Finally, the prediction module might be improved too. In order to increase the accuracy of language predictions, customized language models may be generated starting from subject-specific training texts. A simplified language syntax could be considered, too, for example, discarding articles, prepositions, or any other language token that is not essential for grasping the general meaning of the message to convey.

With this work, we developed an original BCI spelling application based on motor imagery and showed that by combining state-of-the-art algorithms for signal processing and pattern classification with novel EEG features and NLP techniques, good levels of communication rates could be achieved.

APPENDIX

In this appendix, we present the complete user interface model. The model is illustrated in Figure 10. As stated in Section 6.4, *action*-transitions are followed by a push of their source state on the stack, while *undo*-transitions recover the last state saved on top of the stack.

The left part of the diagram shows all interface states by means of which letters are selectable through recursive target expansions. The left, top, and right targets in the starting state contain nine letters each, becoming three letters at the second level, and just one letter at the third level.

While at levels two and three the bottom target is simply associated with an *undo*-transition, in the top-level state, the same target is associated with a *menu*-transition. The reason for this is that we want to provide a set of auxiliary functions once the selection of one letter is completed, but we want to keep the function "undo" as near as possible during group expansions. The functionalities provided after the completion of each letter are "add to dictionary" and "predictions". The first activates all symbols that may have been disabled because of letter prediction, while the second is actually a *menu*-transition pointing to another state in which three predicted words are selectable. The function "add to dictionary" is useless during group expansion since disabled letters are already visible at the top-level state and are not changed until letter selection is completed. The function "prediction", instead, may be useful also at lower levels since predictions are kept updated during group expansion. However, as providing fast access to error correction is quite important, we preferred to have an undo operation directly accessible and to delay the availability of predictions after letter completion.

Once one symbol has been selected, it is appended in text and an *action*-transition is taken. This transition always brings to the starting state except for the case in which a spacing character is selected. In this case, indeed, an entire word has been completed, and more functionalities are made available by the user interface. As shown on the diagram, the transition departing from the spacing character reaches a state in which the targets in the left, right, and top positions still contain the top-level groups of letters, while the bottom transition points to a different menu. This menu and its nested sub-menus, along with "add to dictionary" and "predictions" also provide the following functions: "delete world", "numeric", "speak", and "quit". Since it is unlikely that the user may want to use one of these functions while in the middle of a word, they are provided only after word completion. The function "numeric" returns a set of states in which numbers are selectable. During the composition of numbers, predictions are obviously not available. Finally, the menu accessible through the top-level state in the numeric environment provides the following functions: "undo", "delete character", and

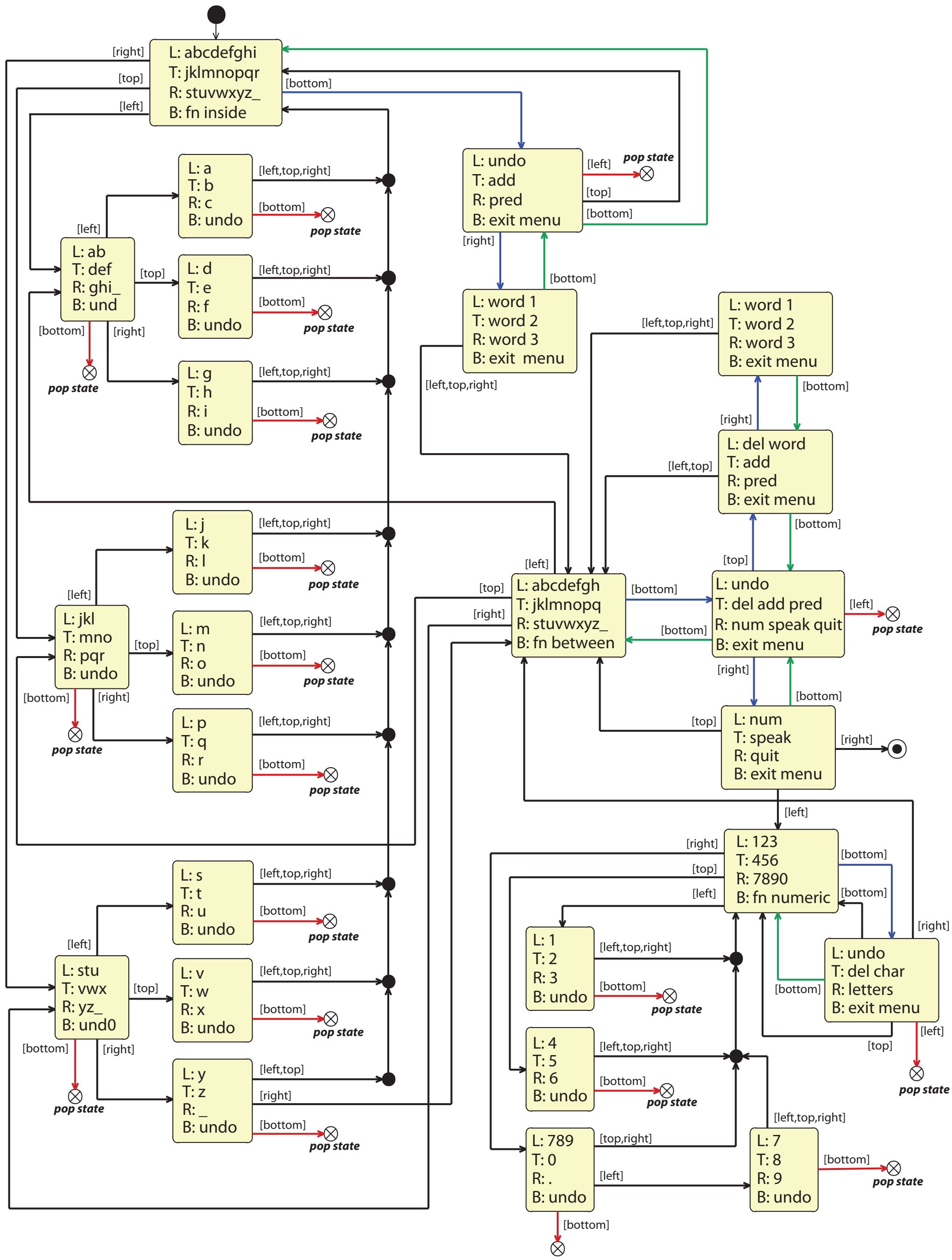


Fig. 10. User interface model with four targets. Each state is labeled with the contents of its targets located at position left (L), top (T), right (R), and bottom (B). *Action*-transitions are depicted in black, *undo*-transitions in red, *menu*-transitions in blue, and transitions of type *exit-menu* in green.

"letters". Selecting the target "letters", a spacing character is added in the text, and the state containing the top-level groups of letters is restored.

REFERENCES

BAUER, G., GERSTENBRAND, F., AND RUMPL, E. 1979. Varieties of the locked-in syndrome. *J. Neurol. 221,* 2, 77–91.

BIRBAUMER, N., GHANAYIM, N., HINTERBERGER, T., IVERSEN, I., KOTCHOUBEY, B., KÜBLER, A., PERELMOUTER, J., TAUB, E., AND FLOR, H. 1999. A spelling device for the paralysed. *Nature 398,* 6725, 297–298.

BISHOP, C. 2006. *Pattern Recognition and Machine Learning*. Springer, New York.

BLANKERTZ, B., DORNHEGE, G., KRAULEDAT, M., SCHRÖDER, M., WILLIAMSON, J., MURRAY-SMITH, R., AND MÜLLER, K. 2006. The berlin brain-computer interface presents the novel mental typewriter hex-o-spell. In *Proceedings of the 3rd International Brain-Computer Interface Workshop and Training Course*. 108–109.

BLANKERTZ, B., KRAULEDAT, M., DORNHEGE, G., WILLIAMSON, J., MURRAY-SMITH, R., AND MULLER, K. 2007. A note on brain actuated spelling with the Berlin Brain-Computer Interface. Lecture Notes in Computer Science, vol. 4555, Springer, Berlin, 759.

BLANKERTZ, B., TOMIOKA, R., LEMM, S., KAWANABE, M., AND MULLER, K. 2008. Optimizing spatial filters for robust EEG single-trial analysis. *IEEE Signal Proces. Mag., 25,* 1, 41–56.

BURG, J. 1975. *Maximum Entropy Spectral Analysis*. Stanford University. http://sep.www.stanford.edu/theses/sep06.

COHEN, J. 1960. A coefficient of agreement for nominal scales. *Educ. Psychol. Meas. 20,* 1, 37.

CURRAN, E. AND STOKES, M. 2003. Learning to control brain activity: A review of the production and control of EEG components for driving brain-computer interface (bci) systems. *Brain Cogn. 51,* 3, 326–336.

DAL SENO, B., MATTEUCCI, M., AND MAINARDI, L. T. 2010a. Online detection of P300 and error potentials in a bci speller. *Comput. Intell. Neurosc. 2010,* Article ID 307254.

DAL SENO, B., MATTEUCCI, M., AND MAINARDI, L. T. 2010b. The utility metric: A novel method to assess the overall performance of discrete brain-computer interfaces. *IEEE Trans. Neural Syst. Rehabil. Eng. 18,* 1, 20–28.

DONCHIN, E., SPENCER, K., AND WIJESINGHE, R. 2000. The mental prosthesis: Assessing the speed of a P300-based brain-computer interface. *IEEE Trans. Rehabil. Eng. 8,* 2, 174–179.

FARWELL, L. AND DONCHIN, E. 1988. Talking off the top of your head: Toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalogr. Clin. Neurophysi. 70,* 6, 510–523.

FUKUNAGA, K. 1990. *Introduction to Statistical Pattern Recognition*. Academic Press, Waltham, MA.

GALE, W. 1995. Good-Turing smoothing without tears. *J. Quant. Ling.*

GOLDBERG, D. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Boston, MA.

GRAIMANN, B., ALLISON, B., AND PFURTSCHELLER, G. 2011. *Brain-Computer Interfaces: Revolutionizing Human-Computer Interaction*. Springer Verlag, Berlin.

HOLLAND, J. 1992. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The MIT Press, Cambridge, MA.

HOPCROFT, J., MOTWANI, R., AND ULLMAN, J. 2007. *Introduction to Automata Theory, Languages, and Computation* 3rd Ed. Prentice Hall, Upper Saddle River, NJ.

JANSEN, B. H., BOURNE, J. R., AND WARD, J. W. 1981. Autoregressive estimation of short segment spectra for computerized EEG analysis. *IEEE Tran. Biomed. Eng. BME-28,* 9, 630–638.

JASPER, H. 1958. The 10/20 international electrode system. *EEG Clin. Neurophysio. 10*, 371–375.

JEANNEROD, M. 1995. Mental imagery in the motor context. *Neuropsychologia 33,* 11, 1419–1432.

JURAFSKY, D. AND MARTIN, J. H. 2000. *Speech and Language Processing*. Prentice Hall, Upper Saddle River, NJ.

KALCHER, J., FLOTZINGER, D., NEUPER, C., GÖLLY, S., AND PFURTSCHELLER, G. 1996. Graz brain-computer interface II: Towards communication between humans and computers based on online classification of three different EEG patterns. *Med. Biol. Eng. Comput. 34,* 5, 382–388.

KATZ, S. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Trans. Acous. Speech Signal Proces. 35,* 3, 400–401.

KOLES, Z. 1991. The quantitative extraction and topographic mapping of the abnormal components in the clinical EEG. *Electroencephalogr. Clin. Neurophysi. 79,* 6, 440–447.

KOZELKA, J. AND PEDLEY, T. 1990. Beta and mu rhythms. *J. Clin. Neurophysiol. 7,* 2, 191–208.

KRUSIENSKI, D., MCFARLAND, D., AND WOLPAW, J. 2006. An evaluation of autoregressive spectral estimation model order for brain-computer interface applications. In *Proceedings of the 28th Annual*

International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS'06). 1323–1326.

LEECH, G. 1992. 100 million words of English: The British National Corpus. *Lang. Res. 28,* 1, 1–13.

LEECH, G., RAYSON, P., AND WILSON, A. 2001. *Word Frequencies in Written and Spoken English: Based on the British National Corpus*. Longman, London.

LOTTE, F., CONGEDO, M., LÉCUYER, A., LAMARCHE, F., AND ARNALDI, B. 2007. A review of classification algorithms for EEG-based brain-computer interfaces. *J. Neural Eng.* 2, R1–R13.

MANNING, C. AND SCHÜTZE, H. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, MA.

MCFARLAND, D., KRUSIENSKI, D., SARNACKI, W., AND WOLPAW, J. 2008. Emulation of computer mouse control with a noninvasive brain–computer interface. *J. Neural eng. 5*, 101.

MCFARLAND, D., LEFKOWICZ, A., AND WOLPAW, J. 1997. Design and operation of an EEG-based brain-computer interface with digital signal processing technology. *Behav. Res. Methods 29,* 3, 337–345.

MCFARLAND, D., MCCANE, L., DAVID, S., AND WOLPAW, J. 1997. Spatial filter selection for EEG-based communication. *Electroencephalogr. Clin. Neurophysiol. 103,* 3, 386–394.

MCLACHLAN, G. 2004. *Discriminant Analysis and Statistical Pattern Recognition*. Wiley-IEEE, Press, Hoboken, NJ.

MIGLIORE, D., BLATT, R., MATTEUCCI, M., CERIANI, S., FONTANA, G., AND SENO, B. D. 2008. Brain control of a smart wheelchair. In *Proceedings of International Conference on Inteligent Autonomous Systems*. 221–228.

NEUPER, C., MÜLLER, G., KÜBLER, A., BIRBAUMER, N., AND PFURTSCHELLER, G. 2003. Clinical application of an EEG-based brain–computer interface: A case study in a patient with severe motor impairment. *Clin. Neurophysio. 114,* 3, 399–409.

NEUPER, C., SCHLÖGL, A., AND PFURTSCHELLER, G. 1999. Enhancement of left-right sensorimotor EEG differences during feedback-regulated motor imagery. *J. Clin. Neurophysiol. 16,* 4, 373.

OBERMAIER, B., MULLER, G., AND PFURTSCHELLER, G. 2003. "Virtual keyboard" controlled by spontaneous EEG activity. *IEEE Trans. Neural Sys. Rehabil. Eng. 11,* 4, 422–426.

PARDEY, J., ROBERTS, S., AND TARASSENKO, L. 1996. A review of parametric modelling techniques for EEG analysis. *Med. Eng. Physics 18,* 1, 2–11.

PFURTSCHELLER, G. AND LOPES DA SILVA, F. 1999. Event-related EEG/MEG synchronization and desynchronization: Basic principles. *Clin. Neurophysiol. 110,* 11, 1842–1857.

PFURTSCHELLER, G. AND NEUPER, C. 1997. Motor imagery activates primary sensorimotor area in humans. *Neurosci. Lett. 239,* 2–3, 65–68.

PFURTSCHELLER, G., NEUPER, C., MULLER, G., OBERMAIER, B., KRAUSZ, G., SCHLOGL, A., SCHERER, R., GRAIMANN, B., KEINRATH, C., SKLIRIS, D. WÖRTZ, M., SUPP, G., AND SCHRANK. C. 2003. Graz-BCI: state of the art and clinical applications. *IEEE Trans. Neural Syst. Rehabil. Eng. 11,* 2, 1–4.

PIERCE, J. 1980. *An Introduction to Information Theory*. Dover Publications, Mineola, NY.

PLUM, F. AND POSNER, J. 1966. *The Diagnosis of Stupor and Coma.* Oxford University Press, New York, NY.

POLICH, J. 1997. On the relationship between EEG and P300: Individual differences, aging, and ultradian rhythms. *Int. J. Psychophysiol. 26,* 1–3, 299–317.

PREGENZER, M. AND PFURTSCHELLER, G. 1999. Frequency component selection for an EEG-based brain to computer interface. *IEEE Trans. Rehabil. Eng. 7,* 4, 413–419.

RAMOSER, H., MÜLLER-GERKING, J., AND PFURTSCHELLER, G. 2000. Optimal spatial filtering of single trial EEG during imagined handmovement. *IEEE Trans. Rehabil. Eng. 8,* 4, 441–446.

RYAN, D., FRYE, G., TOWNSEND, G., BERRY, D., MESA-G, S., GATES, N., AND SELLERS, E. 2011. Predictive spelling with a P300-based brain–computer interface: Increasing the rate of communication. *Int. J. Hum.-Comput. Interac. 27,* 1, 69–84.

SCHALK, G., MCFARLAND, D., HINTERBERGER, T., BIRBAUMER, N., AND WOLPAW, J. 2004. BCI2000: A general-purpose brain-computer interface (BCI) system. *IEEE Trans. Biomed. Eng. 51,* 6, 1034–1043.

SCHERER, R., MÜLLER, G., NEUPER, C., GRAIMANN, B., AND PFURTSCHELLER, G. 2004. An asynchronously controlled EEG-based virtual keyboard: Improvement of the spelling rate. *IEEE Trans. Biomed. Eng. 51,* 6, 979–984.

SPYERS-ASHBY, J., BAIN, P., AND ROBERTS S. J. 1998. A comparison of fast Fourier transform (FFT) and autoregressive (AR) spectral estimation techniques for the analysis of tremor data. *J. Neurosc. Methods 83,* 1, 35–43.

WANG, C., GUAN, C., AND ZHANG, H. 2005. P300 brain-computer interface design for communication and control applications. In *Proceedings of the 27th IEEE-EMBS Annual International Conference of the Engineering in Medicine and Biology Society*. 5400–5403.

WANG, T., DENG, J., AND HE, B. 2004. Classifying EEG-based motor imagery tasks by means of time–frequency synthesized spatial patterns. *Clin. Neurophysiol. 115,* 12, 2744–2753.

WOLPAW, J., BIRBAUMER, N., MCFARLAND, D., PFURTSCHELLER, G., AND VAUGHAN, T. 2002. Brain-computer interfaces for communication and control. *Clin. Neurophysiol. 113,* 6, 767–791.

WOLPAW, J. AND MCFARLAND, D. 2004. Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. In *Proceedings of the National Academy of Sciences of the United States of America 101,* 51, 17849.

WOLPAW, J., MCFARLAND, D., NEAT, G., AND FORNERIS, C. 1991. An EEG-based brain-computer interface for cursor control. *Electroencephalog. Clin. Neurophysiol. 78,* 3, 252–259.

WOLPAW, J., MCFARLAND, D., AND VAUGHAN, T. 2000. Brain-computer interface research at the Wadsworth Center. *IEEE Trans. Rehabil. Eng. 8,* 2, 222–226.

WONNACOTT, T. AND WONNACOTT, R. 1985. *Introductory Statistics*. Wiley, New York, NY.

YAN, T., JINGTIAN, T., AND ANDONG, G. 2008. Multi-class EEG classification for brain computer interface based on CSP. In *Proceedings of the BMEI International Conference on BioMedical Engineering and Informatic.* 469–472.

Received November 2010; revised September 2011; accepted April 2012