P300-Based BCI Mouse With Genetically-Optimized Analogue Control

Luca Citi, Riccardo Poli, Caterina Cinel, and Francisco Sepulveda

Abstract—In this paper we propose a brain–computer interface (BCI) mouse based on P300 waves in electroencephalogram (EEG) signals. The system is analogue in that at no point a binary decision is made as to whether or not a P300 was actually produced in response to the stimuli. Instead, the 2-D motion of the pointer on the screen, using a novel BCI paradigm, is controlled by directly combining the amplitudes of the output produced by a filter in the presence of different stimuli. This filter and the features to be combined within it are optimised by an evolutionary algorithm.

Index Terms—Brain–computer interfaces (BCIs), genetic algorithms, mouse, P300.

I. INTRODUCTION

OVER the past few years, an increasing number of studies have evaluated the possibility of converting signals generated from the brain [especially electroencephalogram (EEG)] into control signals for applications in various disciplines, from virtual reality to hands-free control of augmentative communication technologies for individuals with disabilities. The resulting systems go under the name of brain–computer interfaces (BCIs).

The development of BCI is particularly important as it could provide, for example, new technology that does not require muscle control (e.g., [1]–[9]; see [10] for a comprehensive review). BCI studies have shown that nonmuscular communication and control is possible and might serve useful purposes for people who cannot use conventional technologies, for example, because they are "locked-in" or lack any useful muscle control [11]–[15]. Naturally, BCI systems can also be used by able-bodied users although, in these cases, they tend to complement other forms of control rather than being the only source of input.

The signals used in BCI studies to date include P300 waves [1] and other event related potentials (ERPs), μ or β rhythms [2], evoked potentials (EPs) [6], [16]–[20], event-related desynchronization/synchronization (ERD/ERS) [3], activation patterns induced by mental task strategies [21], slow cortical potentials [4] recorded from the scalp, cortical neuron activity recorded by implanted electrodes (see [5] for a review), neuromagnetic signals recorded through megnetoencelography (MEG) [7], BOLD responses recorded through functional magnetic resonance imaging (fMRI) [8], activity-related, and localized brain oxygenation recorded through near infrared systems (NIRS) [9].

ERPs are relatively well-defined shape-wise variations to the ongoing EEG elicited by a stimulus and temporally linked to it. ERPs include an exogenous response, due to the primary processing of the stimulus, as well as an endogenous response, which is a reflection of higher cognitive processing induced by the stimulus [22]. The P300 wave is a late appearing component of ERPs with a latency of about 300 ms which is elicited by rare and/or significant stimuli, i.e., P300 potentials are ERP components whose presence depends on whether or not a user attends to a rare, deviant or target stimulus. This makes it possible to use them in BCI systems to determine user intentions.

Given the point-and-click nature of most modern user interfaces, an important application of BCI is controlling 2-D pointer movements. Over the years, there have been some attempts to develop BCI systems for this purpose, the most successful of which, to date, being those based on the detection of μ or β rhythms [23], and those using invasive cortical interfaces (e.g., [24]). The former, however, require lengthy training periods before users can control them, while the latter are not very practical, requiring surgery, presenting risks of infections, etc. These problems could be overcome by noninvasive systems based on the use of P300s. To date, however, only limited successes with this approach have been reported in [26] where rather long interstimulus intervals led to the pointer moving at the rate of one movement every 10 s, and [25] where a speed of one cursor movement every 4 s was achieved but accuracy in detecting P300s was only about 50%.

In this paper, a P300-based system for the 2-D control of a cursor on a computer screen is presented. Four randomly-flashing squares are displayed on the screen to represent four directions of movement. Users devote their attention to the flashes of the square towards which the cursor should move. This produces endogenous EEG components following each stimulus, which the system analyses to infer the user's intentions and move the cursor. The system presents two unique features: it completely dispenses with the problem of detecting P300s (a notoriously difficult task) by logically behaving as an analogue device (as opposed to a binary classifier), and it uses a single trial approach where the mouse performs an action after every trial (once per second). This has been made possible by the use of evolutionary algorithms (EAs), which rapidly and effectively adapt the design of the system to each user and each session.

EAs are search and optimisation algorithms inspired by Darwinian evolution, which have been applied very successfully to a large number of difficult problems with human-competitive results. Here, we use a particular type of EA called the genetic algorithm (GA). In GAs, a population of random tentative solutions to a problem is created and evaluated to assess the degree to which they solve the problem at hand. New generations of individuals are created by recombining the characteristics of individuals in the previous generation, giving better performing parent solutions a higher chance of reproduction. Generation after generation, better and better solutions emerge. However, unlike in nature where thousands of years are necessary to evolve fit individuals, in our BCI mouse running the GA requires only minutes.

In our system, the analysis of the P300 components is preceded by a preprocessing phase in which the continuous wavelet transform (CWT) of each EEG channel is performed. CWT is computed at several tens of scales and times after the presentation of the stimuli. So, the ERP response to each stimulus is turned into a large array of features, which are available to the subsequent stages of P300 processing. To avoid the well-known problems associated to large feature sets, in the system we take a wrapper approach to feature selection where the selection of features and the training of the control system using them are performed jointly by the GA. We reported promising results with an evolutionary approach to P300 processing using data obtained with Donchin's speller paradigm [1] in [28]. Encouraged by these, we decided to extend the work and build the complete BCI mouse system with real-time processing and classification described in this paper. The system makes it possible for a person having undergone no previous training and within minutes of wearing the electrode cap, to control a 2-D pointer on a screen.

II. EXPERIMENTAL METHODS

A. Participants

After several preliminary experiments, the final design was tested with six participants: A aged 25, B aged 28, C aged 35, D aged 44, E aged 23, and F aged 40. Participants A–E were able bodied, while F has Dejerine-Sottas hypertrophic neuropathy.

B. Stimuli and Protocol

Four gray rectangles are constantly superimposed on whatever is shown on a computer screen. They are unobtrusive, being small and peripheral, as shown in Fig. 1. Each rectangle corresponds to a possible direction of movement for the mouse cursor. Two additional rectangles, one on the lower left corner of the screen, the other on the lower right corner of the screen, can also be present. When they are enabled, they perform the function of the left and right buttons of an ordinary mouse. When this function is not required, the two "button" rectangles are still logically present, but are not physically shown on the screen to avoid distracting the user.

At 180 ms intervals, this static display is altered by temporarily changing the colour of one of the rectangles from gray to bright red (for the rectangles controlling the direction of motion) or blue (for the rectangles representing mouse buttons).
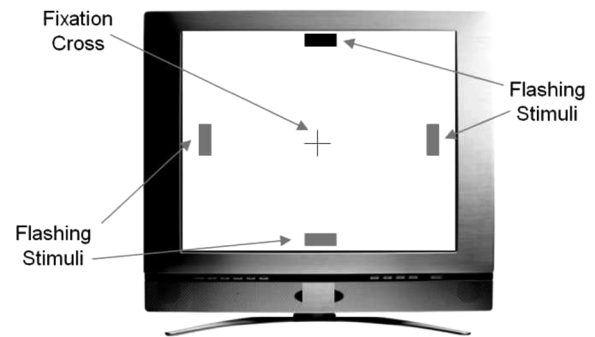

Fig. 1. Display used to control our BCI mouse. Four rectangles at the borders of the screen correspond to the possible directions of movement. They flash in random order. The picture shows the display when the stimulus "up" is presented (shown in light gray in this black and white reproduction, although it is bright red in reality). Users focus their attention on the flashes of the rectangle corresponding to the desired direction of motion.

The stimulus remains brightly colored for 100 ms, after which it becomes gray again. As a result, the rectangle appears to flash. Which particular rectangle is selected for flashing is determined randomly. However, after flashing, a rectangle is not allowed to flash again until all other rectangles have flashed once. When the buttons' function is disabled, as in all the experiments reported in this paper, the rectangles corresponding to the mouse buttons are not drawn and, so, their flashing has not effect except creating a 180-ms pause in the sequence of flashes. When the two "button" rectangles happen to be "flashed" in sequence, a 360-ms pause is produced instead, although this is infrequent. As we stressed in [29], repetition blindness and other perceptual errors often observed in rapid serial visual presentation experiments can reduce the accuracy of BCI systems. To limit the risks of this type of perceptual errors occurring in our BCI mouse, we ensured that the last rectangle to flash in each series of 6 was never allowed to be the first to flash in the following series.

Participants were comfortably seated on an armchair with their neck supported by a C-shaped inflatable travel pillow to reduce muscular artifacts. The eyes were approximately 1 m from the computer screen.

Participants were instructed that in order to move the mouse pointer in a particular direction they needed to focus their attention (but not necessarily point their gaze) on the corresponding rectangle on the screen. To facilitate this, they were asked to count how many times that particular rectangle flashed during a sequence of mouse movements in the same direction. Participants were asked to ignore whatever else was happening on the screen. Note that while it is slightly easier for subjects to perform the task when pointing their gaze on the rectangle of interest, the flashing rectangles are sufficiently conspicuous to be easily counted via peripheral vision.

A fixation cross was always present in the middle of the screen. We offered participants the option to point their gaze on the fixation cross. Three of our six participants reported to have made a conscious effort to constantly point their gaze on the fixation cross, while focusing their attention on the flashing rectangles. One subject reported to have pointed his gaze halfway between the rectangle of interest and the fixation cross.

The system has three modes of operation. The training mode is used for acquisition of training sets. Here, the experimenter selects one of the rectangles on the screen as a target, and participants are asked to focus their attention only on the target stimulus. In this mode, the screen has a light gray background with no stimuli other than the rectangles mentioned above. The tracing mode is similar, in that participants perform the same task with the same stimuli and homogeneous background. However, now signal processing, feature extraction and combination are performed in real-time and we can show participants the trajectory of the mouse pointer produced by their efforts. Feedback can be provided at each time step (i.e., once per second) or after a user-settable number of time steps. Finally, in normal use all sorts of windows and icons are present on the screen in addition to the rectangles necessary to control the BCI mouse. There is a variant of this mode of operation, which we call "scrolling mode," where instead of moving the mouse pointer on a fixed screen we scroll the screen. This ensures that the entities of interest for the user are always near the fixation cross. In scrolling mode, we used a zoom factor of 2 to ensure maximum readability.

Here, we only present results with the training and tracing modes. In tracing mode, we provided visual feedback to the user only every 30 time steps (lasting approximately 1 s each). In these conditions, the displays and acquisition conditions in the two modes are identical. This allowed us to do full offline analysis of performance, including cross-validation (Section IV). Each run of our experiments involved presenting a full series of four flashing rectangles for 30 times. The process was repeated for each of the four possible directions, multiple times for each direction. Every few runs, participants were given a few minutes to rest and stretch. For participant A, 12 runs were recorded (three for each of the four directions) while the others performed 16 runs (four for each direction).

C. EEG Apparatus

We used a 19 channel setup in a Mindset24 System to acquire and digitise EEG signals. Within the device the signal is first filtered with two order-3 analogue bandpass filters with 3 dB band between 1.5 and 34 Hz and then sampled at 256 samples per second. We used an electrode cap (Electrocap International) with electrodes already prearranged according to the 10–20 international standard. Signals are referenced to the earlobe with the lowest impedance. Efforts were made to obtain impedances below 7 kΩ in all experiments.

III. Preprocessing and Translation Methods

A. Preprocessing

We used 19 channels corresponding to the 10-20 international system to acquire EEG. Each channel is low-pass filtered using a finite impulse response (FIR) filter of order $N = 30$. The coefficients of the filter were obtained via the least mean squares method with the transition band between $f_{\text{pass}} = 34$ Hz and $f_{\text{stop}} = 47$ Hz. After low-pass filtering, the signal is decimated to 128 Hz by leaving every other sample out.

B. Feature Extraction

After filtering and decimation, we compute the features to be made available to the GA which will be responsible for feature selection and filter optimization. Our features are obtained by performing CWT on the 19 channels. The wavelet transform $C_{a,b}$ is defined as the inner product between a signal and a function, known as mother wavelet, appropriately scaled and translated. That is $C_{a,b} = \int_{\mathbb{R}} s(t)(1/\sqrt{a})\psi^*(t - b/a)\mathrm{d}t$, where $b \in \mathbb{R}$ represents the temporal shift of the wavelet, while $a \in \mathbb{R}^+$ is the scale. We used the *rbio3.3* wavelet family.

The features are extracted for each epoch of the signal. An *epoch* is a 1-s window starting when a stimulus is presented. In each epoch the system needs to process the EEG signals and appropriately emphasize and utilize a P300, if this is present in the epoch. Note, however, that stimuli are presented at a very fast rate, and, so, epochs overlap. This makes the task much harder since the late responses to a stimulus may interfere with P300s generated by another.

Pointer control is determined by a filter (more on this later) which is applied to each epoch. No averaging takes place. The filter uses the coefficients of the CWT of an epoch. We use 30 different scales between 2 and 40. Scales are not equally spaced. With a uniform scale distribution, the pseudo-frequencies corresponding to smaller scales are too far apart, while those corresponding to bigger scales are unnecessarily close. Thus, we used the polynomial $a = (1/45)s^2 + (2/3)s + 2$, which approximates an exponential for $s = 0, 1, \ldots, 29$.

Since P300s occur within a well-known time window, we compute CWT only for a range of relevant samples within each epoch. These correspond to the translations $b = t + 30$ for $t = 0, 1, \ldots, 39$. At our sampling frequency of 128 Hz (after decimation), these correspond to a temporal window between 235 and 540 ms after the beginning of each epoch.

To sum up: for each epoch we process 19 channels, for each channel we compute CWT at 30 different scales, and at each scale we compute 40 consecutive samples. This gives us a 3-D array $\mathbf{V}(c, s, t)$ of features, where c indexes the channel, s the scale, and t the time corresponding to a feature. In total we have $19 \times 30 \times 40 = 22\,800$ components. Such a large number of features requires some form of feature selection. To do this, we used a wrapper approach which involves searching for the best features and for the best parameters for our mouse controller (the semi-linear filter described in the next section) concurrently. This is more demanding than performing the two tasks independently since there is a much bigger search space to explore. However, it also has the advantage of potentially giving the best possible results, which is very important given the high noise and limited information content in EEG signals and the difficulty of the task. Given the size of the search space and its discontinuities, it appeared natural to use a robust search algorithm such as a GA for this task. In our system, the GA needs to choose the best features and filter parameters to control the 2-D motion of the pointer. As described in the next section at the hearth of our system is a semi-linear filter.

C. Filter

Our controller P performs a linear combination of a subset of elements of the feature matrix $\mathbf{V}$

$$P(\mathbf{V}) = a_0 + \sum_{j=1}^{N} a_j \cdot \mathbf{V}(c_j, s_j, t_j) \qquad (1)$$

where N is the number of terms in the filter, the coefficients c_j, s_j, t_j identify which component of $\mathbf{V}$ is used in the jth term, and finally the values a_j are coefficients weighing the relative effect of each term. Although a linear filter may appear a simplistic choice, it actually was well pondered. Extensive preliminary experiments with a more general class of nonlinear filters (polynomial filters), which we used very successfully in the treatment of other biomedical signals (e.g., in [27]), revealed that linear terms play a predominant role in P300 exploitation in our system, nonlinear terms appearing rarely in the evolved filters and, when present, contributing marginally to the output. These tests also revealed that $N = 3$ or $N = 4$ were good choices.

The value $P(\mathbf{V})$ is then passed through a "squashing" function ϕ to produce the filter output $O(\mathbf{V}) = \phi(P(\mathbf{V}))$ (squashing functions are used to limit the output of classifiers to some predefined interval). We could interpret a high value of $O(\mathbf{V})$ as an indication of the degree to which an epoch contains a target and vice versa. In our experiments, targets are the stimuli on which a participant is focusing his/her attention for the purpose of selecting a particular direction of movement for the mouse cursor.

D. Cursor Motion Control

Every time a stimulus flashes, an epoch starts. For each epoch the system records the position of the corresponding stimulus on the screen and acquires and processes a 1-s segment of EEG signal. Epochs acquired during the period of training are annotated also with the direction on which the participant was focusing.

In epochs corresponding to target stimuli, we expect to find a P300 wave, while in epochs where a nontarget stimulus flashed this should not be present. The job of the GA is to use the information contained in the training set to evolve a filter that can exploit the differences between these two cases and achieve correct pointer control.

Computer mice are analogue devices (the more you move the mouse, the more the pointer on the screen moves). So, it seemed inappropriate to turn analogue brain activity recorded in the EEG into binary form to later turn the signal in analogue form again (which is necessary to move the pointer on the screen). Given the limited information available in EEG, we felt that an analogue BCI approach would avoid any further losses introduced by the detection process and would offer the potential to use P300 amplitude information.

To obtain this, the motion of the pointer is directly determined by the squashed output of the filter. More precisely, the vertical motion is proportional to the difference between the output produced by the filter when processing an epoch where the up rectangle was flashed and the output produced by the filter when processing an epoch where the down rectangle was flashed. The horizontal motion of the pointer is similarly determined by the responses to the flashing of the right and left rectangles.

Therefore, the task of the GA is not just selecting features and designing filters to best discriminate between P300 and non-P300 responses, but also to do so in such a way that the responses to pairs of stimuli provide the fastest and most precise way of moving the pointer in the desired direction. This, in principle, allows the full exploitation of any analogue variations present in P300s.

In order to turn P300s into mouse pointer motion, we divide the stream of epochs into groups of four. Each group contains, in random order, epochs corresponding to the flashing of all four possible stimuli. As soon as a full group is acquired, the features and the output of the function $P(\mathbf{V})$ are computed for each of the four epochs. Outputs are then squashed via the function $\phi = \arctan$.

As a result of these operations, we obtain a tuple of output values (O_u, O_d, O_l, O_r), where subscripts refer to the up, down, left and right stimuli, respectively. These are used to compute the motion vector $v = (v_x, v_y)$ for the mouse cursor on the screen, where $v_x = O_r - O_l$ and $v_y = O_u - O_d$.

E. Evolving Filters

The GA has the task of identifying a high-quality set of parameters for the filter in (1). These include: $N+1$ real-valued coefficients a_j; N feature channels c_j, which are integers in the set $\{0, \ldots, 18\}$; N integer feature scales s_j in the set $\{0, \ldots, 29\}$; and N integer feature samples t_j in the set $\{0, \ldots, 39\}$. The operation of feature selection is performed by the GA by choosing N tuples (c_j, s_j, t_j), while the filter training is performed by optimizing the corresponding a_j's.

The representation used is simply a concatenation of the floating-point and integer parameters of the linear filter in (1). We encoded both real and integer parameters as floating-point numbers, taking care of rounding c_j's, s_j's and s_j's to the nearest integer before using them in (1).

As our search operators we used blend-crossover and headless chicken crossover. In blend crossover [30], the offspring $(a_1^s, \ldots, a_i^s, \ldots, a_n^s)$ is obtained, component by component, using the formula $a_i^s = a_i^1 + c_i(a_i^2 - a_i^1)$, where a^1 and a^2 are parent individuals and, for each i, a different value of c_i is drawn uniformly at random from the interval $[-\alpha, 1 + \alpha]$, α being a suitable nonnegative constant. In this work, we used $\alpha = 0.1$. Headless chicken crossover, instead, is performed by recombining an individual selected from the population with a randomly generated individual.

As a selection operator, we chose tournament selection with tournament size 3. To maximally speedup evolution we used a steady state GA, where each individual created is immediately inserted in the population, without waiting for a full new generation to be ready. Since the population size is constant, we used negative tournaments (where the worst individual in the tournament is selected) as a replacement strategy. We used populations of 50 000 individuals.

F. Fitness Function

While we were able to use a standard representation and standard genetic operators, the design of the fitness function involved much more work and required numerous successive refinements. The final fitness function is described below.

The natural objective function for a mouse is, of course, the extent to which the pointer was moved in the desired direction.

So, this is clearly a necessary component in our fitness function. However, this is not enough. For example, it is possible that the pointer moved a great deal in the desired direction, while at the same time drifting significantly in a direction orthogonal to the desired one. A fitness function based on the natural objective would reward this behavior, effectively leading to very sensitive but not very precise controllers. So, clearly the problem is multiobjective, that is we want to obtain both maximum motion in the desired direction and minimal motion in the orthogonal direction.

As is routinely done in these cases, to deal with this problem we adopted a penalty method to combine the multiple objectives into a single fitness measure. With this method the fitness function f is a linear combination of the objectives ω_i of the problem, i.e., $f = \sum_i \lambda_i \omega_i$, where λ_i are appropriate coefficients (by convention $\lambda_1 = 1$).

Naturally, the quality of a controller cannot be evaluated on a single trial. This would not only be unrealistic (ten or more cursor movements may be required to move the mouse to the desired spot in a large screen) but it would also be very unreliable given the noise in EEG signals, and complexity of the task at hand. So, the performance of the controller was evaluated on the basis of its behavior over groups of 30 repetitions of a command (up, down, left or right), which we will term runs. Furthermore, in order to ensure that the controller performed well in all directions, these runs were acquired for all possible directions (and, in fact, multiple times for each direction to limit risks of over-fitting). All the resulting trials formed the controller's training set.

In each of the examples in the training set, the controller produced a velocity vector. Let us call $v^{i,t} = (v_d^{i,t}, v_o^{i,t})$ the velocity vector produced at repetition i in the tth run. This vector is expressed in a reference system where the component $v_d^{i,t}$ represents the motion in the target direction, while $v_o^{i,t}$ represents the motion produced in the direction orthogonal to the desired direction.

In the mouse control problem, we used three different objectives. The first objective, ω_1, assesses the extension of the motion in the target direction, the other two, ω_2 and ω_3, evaluate the motion in orthogonal direction. In particular, ω_2 assesses the average extension of the motion in such a direction at the end of runs. It, therefore, ignores any errors that have been later cancelled by errors with opposite sign (i.e., in the opposite direction). Instead, ω_3 evaluates the average absolute error deriving from motion orthogonal to the target direction. So, it assesses the extent to which the trajectory towards the target is convoluted. These objectives are computed as follows:

$$\omega_1 = \nu \sum_{t=1}^{N_r} \sum_{i=1}^{30} v_d^{i,t}$$

$$\omega_2 = \nu \sum_{t=1}^{N_r} \sum_{i=1}^{30} v_o^{i,t}$$

$$\omega_3 = \nu \sum_{t=1}^{N_r} \sum_{i=1}^{30} |v_o^{i,t}|$$

where N_r is the number of runs and $\nu = 1/(30 \times N_r)$ is a normalization factor. Since we want to maximize ω_1, but minimize $|\omega_2|$ and ω_3, and ω_1 is our main objective, the coefficients λ_2

and λ_3 need to be negative with magnitude much smaller than λ_1. In preliminary experiments, we found that the following set of values worked well: $\lambda_1 = 1, \lambda_2 = -0.2$ and $\lambda_3 = -0.2$. So, we adopted these parameters in the work reported here. The resulting fitness function is $f = \omega_1 - 0.2 \times |\omega_2| - 0.2 \times \omega_3$.

IV. RESULTS

A. Evaluation of Performance

A four-fold cross-validation (three-fold for participant A) has been applied to train the system and test its performance and generalization ability. For each participant, a total of 16 (or 12 for participant A) runs has been split in groups of four each containing one run for each direction (1–4, 5–8, etc.). Therefore, there were four groups for each participant (three for participant A). Then two of these groups have been used as training set while a third one as validation set. For each of the 12 combinations (for participants B–F), a population has been evolved while for each of the three combinations for participant A four populations have been evolved. So, for each participant a total of 12 different filters have been evolved for 40 generations. Filters were allowed to use up to $N = 4$ terms.

The three different objectives (Section III-F) have been evaluated for the training-set as well as for the validation-set. The outcomes in the validation-set have in no way influenced the evolution phase nor its termination and, therefore, can be considered a reliable index of the generalization ability of the filter when applied to new data.

Table I shows the values of ω_1, ω_2, and ω_3 for filters trained with the n-fold cross validation method explained above. For each participant, the average across the 12 evolutions, together with the standard deviation in brackets, is reported. Standard deviations are computed across averages, i.e., they represent the standard error of the mean.

These results have been obtained by processing offline data acquired with the BCI mouse in training mode where stimuli are presented to a participant but no feedback is provided as to the extend and direction of motion of the pointer. This is to ensure that the cleanest possible dataset is gathered. As soon as the acquisition of an appropriate training set is completed, the system performs all the necessary preparatory steps, including filtering and feature detection, for the application of the GA, whose runs are typically successful within 30 generations. The data preparation and the training process require between 5–10 min. The system can then be used.

It is apparent from the analysis of the results in Table I that, despite a slight tendency to overfit, the EA was able to find satisfactory solutions and all participants could move the pointer in the target directions (as quantified by ω_1) with very limited lateral error at the end of a sequence of commands (quantified by ω_2). In fact, if we consider ω_1 as the amplitude of the signal and ω_2 as the standard error of the noise in our system, then $\omega_1/|\omega_2|$ gives us an idea of the signal-to-noise ratio (SNR) during long-distance movement of the pointer. Averaged over our six participants, $\omega_1/|\omega_2|$ is 28.1, which gives us a good SNR = 29.0 dB on the validation set. Note that the notion of SNR seems particularly appropriate for an analogue pointer control system. This is because in these systems one can easily

TABLE I

VALUES OF ω_1 (EXTENSION OF THE MOTION IN THE TARGET DIRECTION), ω_2 (EXTENSION OF THE MOTION IN A DIRECTION ORTHOGONAL TO THE TARGET AT THE END OF A SEQUENCE OF TRIALS) AND ω_3 (STEP-BY-STEP MOTION ORTHOGONAL TO THE TARGET DIRECTION) FOR FILTERS TRAINED WITH THE n-FOLD CROSS VALIDATION. VALUES ARE AVERAGES AND STANDARD DEVIATIONS (IN BRACKETS) ACROSS 12 RUNS OF THE GA

| Participant	Training set			Validation set		
	ω_1	ω_2	ω_3	ω_1	ω_2	ω_3
A	1.0943 (0.1349)	0.0262 (0.0236)	1.2868 (0.0968)	0.7126 (0.1161)	0.0415 (0.0728)	1.3516 (0.0880)
B	1.5610 (0.0836)	-0.0035 (0.0282)	1.0283 (0.1432)	1.2713 (0.1857)	-0.0563 (0.1145)	1.0794 (0.1877)
C	1.3335 (0.1457)	0.0208 (0.0626)	1.1798 (0.1307)	1.0831 (0.2377)	0.0508 (0.0634)	1.2161 (0.1966)
D	1.0893 (0.0489)	0.0158 (0.0434)	1.3240 (0.1677)	0.5253 (0.2325)	0.0112 (0.1684)	1.3676 (0.1268)
E	1.4921 (0.0572)	0.0215 (0.0333)	1.1918 (0.1114)	1.1101 (0.1628)	0.0267 (0.1032)	1.2477 (0.1695)
F	1.0335 (0.0922)	-0.0068 (0.0328)	1.2790 (0.0827)	0.5101 (0.1299)	0.0269 (0.1449)	1.3617 (0.1290)
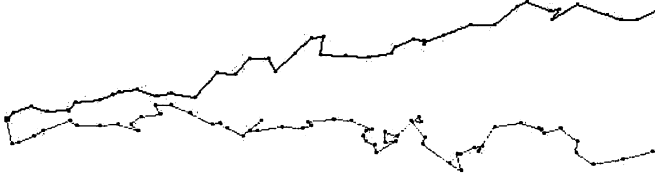
Fig. 2. Sample long-term behavior of the BCI mouse with a fragment of data from the training set (top trace) and the validation set (bottom trace). Every dot in the plots corresponds to a motion vector issued by the system.

achieve any desired speed of motion by multiplying the pointer velocity vectors by an appropriate constant. This multiplication effectively acts as a perfect, (almost) noise-free, amplifier for the original control signals. Like in any amplifier, the SNR of the original signal will still have an effect on the output.

This good performance is achieved thanks to the self-correcting nature of the task. That is, a mouse in an integral controller, where the final position of the pointer is the result of summing many contributions (velocity vectors) over a period of time. For this reason, if the noise (the motion in a direction orthogonal to the target) has a zero mean, the noise contributions will tend to cancel. This is what happens in our BCI mouse, as one can easily see by comparing ω_3, which gives us an idea of the noise on each single velocity vector, with ω_2 which, instead, gives us an indication of the residual noise at the end of a sequence of commands.

Fig. 2 illustrates the BCI mouse behavior at the end of a typical training run (for participant A using runs 1–4 and 9–12 as training set and 5–8 as validation set). The top plot represents the positions the mouse pointer took over a sequence of stimuli in the training set. The bottom plot shows the behavior of the mouse when using the validation set. As can be seen, in both cases the motion of the pointer is reasonably straight and mostly in the target direction. Note how the steps performed by our mouse are not all of the same size and how deviations orthogonal to the desired direction of motion tend to cancel out. This is a result of our analogue approach.

Although the small values of ω_2 in Table I suggest good long-distance movement precision, marked deviations from a straight line trajectory may occur when short-distance cursor movements are required. This is because in single trials, or over a small number of trials, the self-correcting effects of the task are either absent or less pronounced, respectively. Indeed, motion in orthogonal direction (fitness value ω_3) was for all subjects, except for B, larger than the movement in the target direction. In Fig. 3, we show an example of short-term behavior for each

of the subjects, for whom we plot the trajectory of the mouse pointer when one of the filters trained by the GA is applied to a typical groups of 15 consecutive trials in the validation set. The corresponding values of the objective functions ω_1, ω_2, and ω_3 are also indicated.

As one can see, even if the value of ω_3 is comparable with ω_1, motion in the desired direction is generally achieved even in the short distance (except for participant F, who appears to have had a lapse in concentration during some of the trials). However, lateral deviations from a desired trajectory, such as those observable in Figs. 2 and 3 could be easily compensated by a participant by simply focusing attention on stimuli orthogonal to the desired direction of motion for some flashes. This was not allowed, however, during the acquisition of the data shown in Figures 2 and 3 nor in any other phase of training and validation.

B. Strengths of the Analogue Approach

These results strongly suggest that our analogue approach is viable. One might wonder, however, exactly how the system operates. To explain this, we will use a representative example extracted from the validation set for Subject A. In this example the system had evolved the filter $P(\mathbf{V}) = -2.1328 + 0.1909 \cdot \mathbf{V}(6, 23, 17) + 0.1741 \cdot \mathbf{V}(8, 20, 22) + 0.4210 \cdot \mathbf{V}(9, 18, 8) - 0.5247 \cdot \mathbf{V}(13, 18, 9)$, which corresponds to

$$
\begin{aligned}
P(\mathbf{V}) = &-2.1328 \\
&+ 0.1909 \cdot \mathrm{CWT}(F8(t + 367\mathrm{ms}), 29.1) \\
&+ 0.1741 \cdot \mathrm{CWT}(C3(t + 406\mathrm{ms}), 24.2) \\
&+ 0.4210 \cdot \mathrm{CWT}(Cz(t + 297\mathrm{ms}), 21.2) \\
&- 0.5247 \cdot \mathrm{CWT}(P3(t + 305\mathrm{ms}), 21.2) \quad (2)
\end{aligned}
$$

if one maps electrode numbers to channel names and performs the scalings and time-shifts described in Section III-B.

To better understand how this operates, we consider an approximation. Let us ignore the one-sample shift between channels Cz and P3 and let us neglect the terms involving channels F8 and C3. The remaining terms are the result of convolving the corresponding signal with the mother wavelet at scale $a = 21.2$. Because CWT is linear, we could calculate the sum of these terms by first computing a linear combination of the raw signals from channels Cz and P3, and then convolving the result with such a wavelet. To give an idea of what this might look like for the example discussed above, we provide it in Fig. 4 for the epochs corresponding to the flashing of our four stimuli. Interestingly, all signals plotted appear to be markedly less noisy than the original channels. In addition, the P300 component present
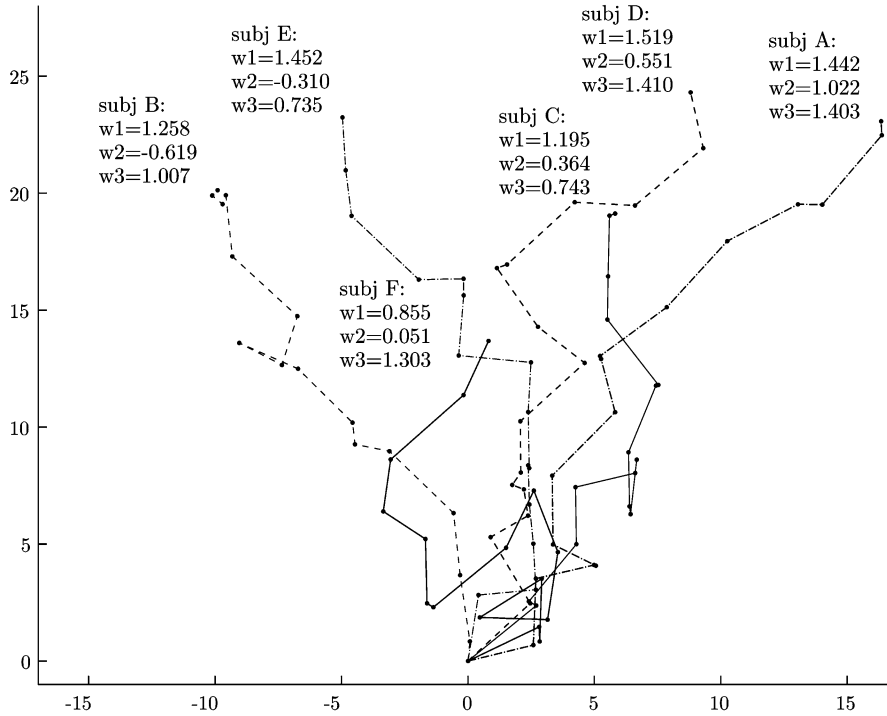
Fig. 3. Plot of the trajectory of the mouse pointer when filters trained by the GA are applied to 15 typical consecutive trials in the validation set. For each trajectory, we also show the values of $\omega_1, \omega_2,$ and ω_3.
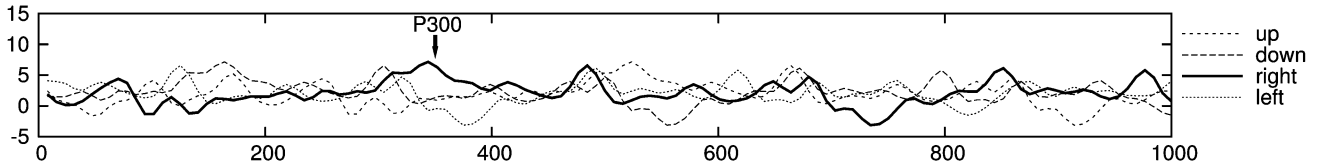

Fig. 4. Linear combination $0.4210 \cdot Cz - 0.5247 \cdot P3$ after the flashing of the up, right (target), left, and down rectangles.

when the target stimulus was flashed (the right rectangle) appears to be better delineated.

As shown in Fig. 5, after convolution with the wavelet, only the signal corresponding to the flashing of the right rectangle presents an ample positive wave in the P300 area. All other signals present a negative wave in the same region. In addition, the up and down signals remain almost identical well until after the P300 region. After squashing and pairwise subtraction, the signals in this approximate model are almost identical to those provided by the full (2), particularly in the region around the origin (where the signal is sampled to move the mouse pointer).

This analysis indicates that effectively the crucial component in the filter evolved by the GA is the weighted difference $0.4210 \times Cz - 0.5247 \times P3$, which appear to provide better information on the user's intentions that any of the channels separately. We believe that the other channels are used to minimize exogenous errors. Particularly, channel F8 appears to be used to inhibit pointer motion when participants performed ocular movements or blinked.

C. Statistical Analysis of Channels and Wavelets Used

Because the system adjusts to different users by using a stochastic search technique—a GA—in different runs one obtains different features sets and filter coefficients. So, it is interesting to look at which channels and which CWT times and scales are commonly used in solutions found by the GA. Fig. 6 shows histograms of these quantities for subject A. The histograms plot the fraction of times a given parameter value (channel, scale, time shift) was selected in one term of our filter in the cross-validation scheme. Note that since our filters had four terms, a frequency of 25% would mean than on average each evolved filter used a particular channel, scale or time shift. This is the case, for example, for channel P3 and for a time-shift of 300 ms. These distributions confirm, for example, that the filter in (2) is in fact typical. The most frequently used CWT parameters point directly to the wavelet shown in Fig. 8, which, as we discuss in the next section, matches almost perfectly the template of P300 reported in [31] when distorted by our device.

V. DISCUSSION

In this paper, we have presented a BCI system for the 2-D control of a cursor on a computer screen which requires no prior user training. Some of the techniques used in the present work are well-established technology. However, unlike previous approaches, we offer several novel features, which allow our system to deliver significantly better performance than other 2-D pointer control systems based on P300s.
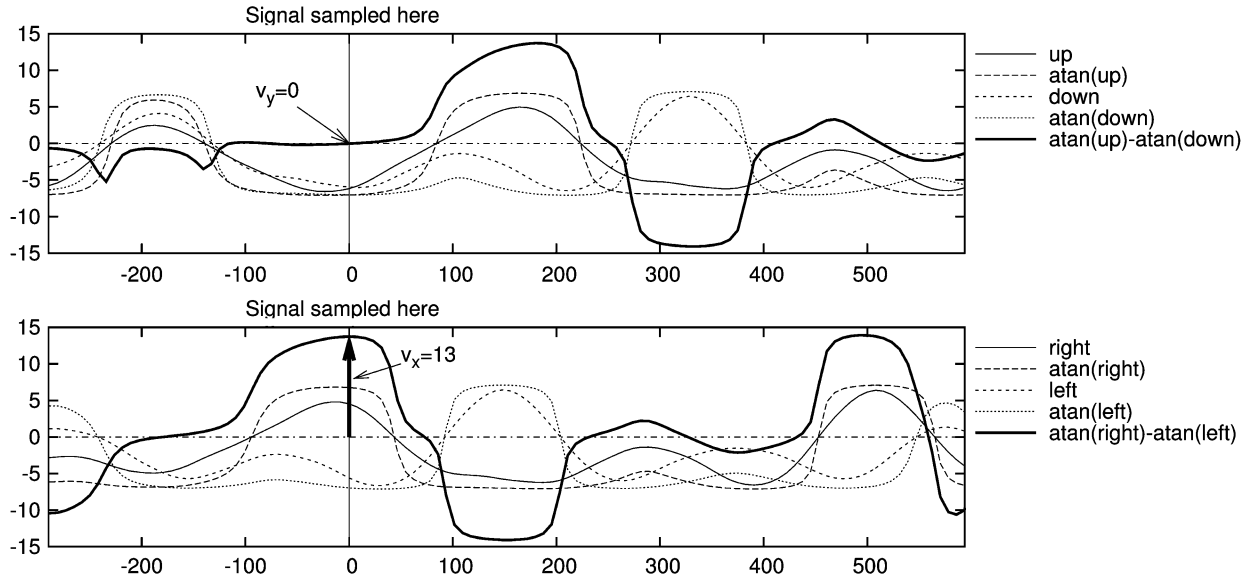
Fig. 5. Result of convolving the signals in Fig. 4 with the wavelet with scale $a = 21.2$ (up, down, left, and right plots). The signals are then squashed (atan(up), atan(down), atan(left), and atan(right) plots) and subtracted pairwise to obtain an approximation of the actual motion of the pointer.
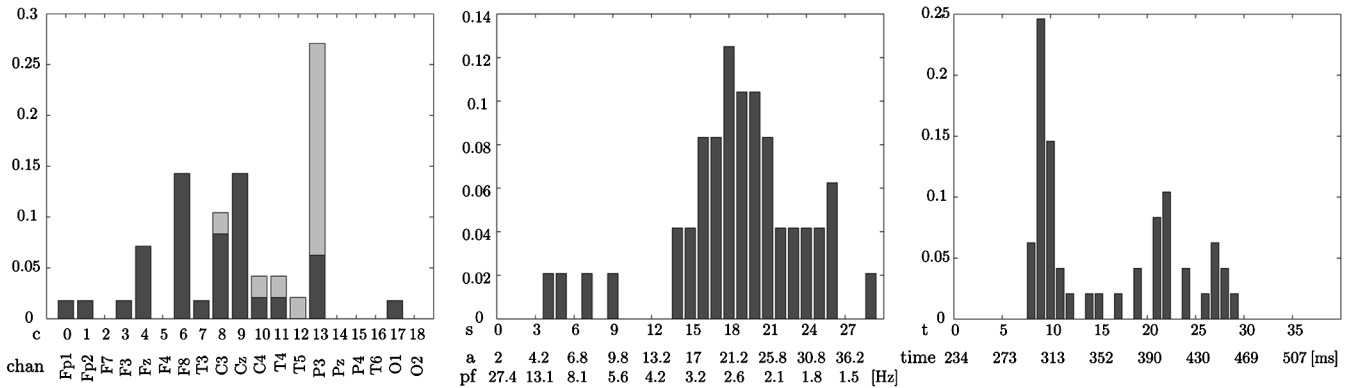

Fig. 6. Channels (left), scales (center), and time shifts (right) selected by the GA for subject A. Light gray indicates negative coefficients for a channel. For the scales diagram, we give three abscissas: the parameter "s," the scale "a," and the pseudofrequency "pf."

First, most P300-based BCI systems require a binary classification of the response to each of a set of stimuli as target or nontarget. Instead, in our system the vertical component of the movement is determined by the difference between a filter's output resulting from the stimuli up and down, while the horizontal component is similarly determined by the responses to the stimuli right and left. That is, logically our BCI mouse is an analogue (as opposed to digital) device (although physically all its signal processing is digital).

Second, the reliable detection of P300s in a single trial is still very difficult. In most systems an action is performed only when the system is reasonably certain as to the intentions of the user. However, this requires averaging over multiple presentations of the stimuli, avoiding high stimulus-presentation rates or both, which in turn reduces the bit rate of the system. In this work, instead, we take a single trial approach and we use a high presentation rate. That is, our BCI mouse performs an action at the end of every cycle of stimulus presentation and this happens approximately once per second. This is possible also thanks to the self-correcting nature of the task.

Finally, we use these novel characteristics in conjunction with state-of-the-art EAs, which effectively adapt the design of the system to each user and each session, thereby maximizing performance. This evolutionary adaptation is particularly powerful because it also performs the all-important phase of feature selection (jointly with the phase of parameter tuning) which is critical for a system with tens of thousands of features such as ours.

The main purpose of this study was to explore the advantages and limits of analogue control in a P300-based BCI. This required to use standardized conditions across multiple users. In these tests, the performance of our BCI mouse were very encouraging. Control in tracing mode (validation) was accurate and all participants were able to use the system within minutes of wearing the electrode cap. In the rest of this section, we address the issue of whether or not our system is based on P300s and we indicate problems that still require addressing.

A. Is Our Mouse P300-Based?

It is clear that we have created an "odd ball" paradigm in our display, a paradigm that facilitates the elicitation of the P300 by
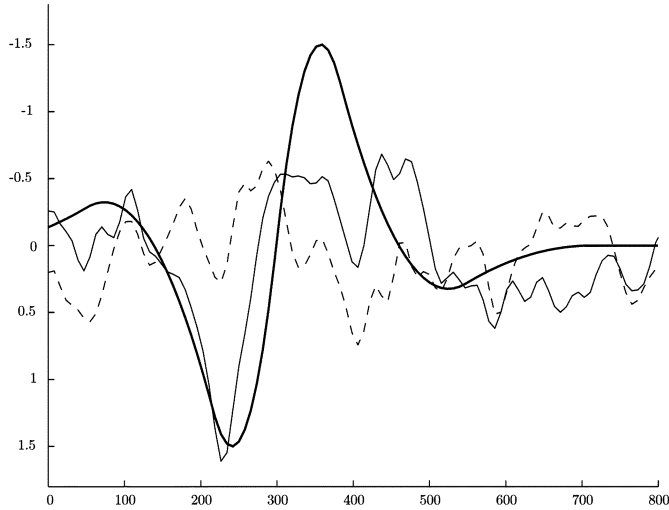
Fig. 7. Average of all the ERP related to flashes that caused a cursor movement in their direction (thin solid line) and average of the opposite direction (dashed line). Recordings are related to subject A, channel Cz, and the averages comprise 338 epochs. Mother wavelet "rbio3.3" (gain −12, scale 22.5, centred at approximately 300 ms) is also shown (thick solid line). This family was chosen because of its similarity with the signal to match. Note the inversion of the vertical axis (usual in ERP studies).
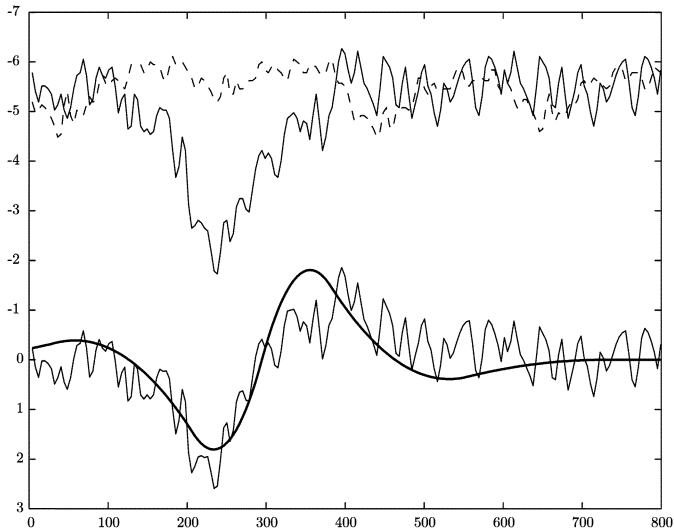

Fig. 8. Effects of the hardware filter in our EEG acquisition device. In the upper part of the figure we show an example of a waveform that could be used as a template for an ERP containing a P300 component (solid line, adapted with permission from [31] and shifter vertically for clarity) and a corresponding signal where there is no P300 (dashed line). Solid thin line in the lower part of the figure is the result of processing the signal containing the P300 with a software filter similar to the hardware filter of the EEG device (lowpass 34 Hz, highpass 1.5 Hz at −3 dB; ≈0.9 Hz at −6 dB). Finally, the thick line represents the mother wavelet "rbio3.3" as in Fig. 7.

deviant events. However, one might wonder whether our system does indeed have a P300 as the deciding factor for the output of the system. This question is particularly relevant since the recording bandwidth of many EEG acquisition devices (e.g., our Mindset24) misses some high energy spectral components of P300 [32]. With our bandwidth between 1.5–34Hz, in principle it would be possible that the "features" that drive our BCI mouse had little to do with the P300. The most natural way to check for this is perhaps to sort all the flashes into groups that did, and

did not, cause a cursor movement in their direction. If pointer control was achieved through the use of P300s, then averaging the data within the groups should yield traditional ERPs where the P300 will be elicited by those targets in whose direction the filter sent the cursor. Indeed this is exactly what happens in our experiments. For example, Fig. 7 compares the average of all the ERP epochs for participant A related to flashes that caused a cursor movement in their direction against the average of the ERP epochs in the opposite direction for channel Cz. The plots for the "odd ball" stimuli are similar to those reported in other P300 studies such as the one obtained by Sellers [31], and reported in Fig. 8 for comparison.

Having established that P300s are indeed elicited by our stimuli, the question then is: how can the system exploit P300s if our acquisition hardware cuts off frequencies below 1.5 Hz (see Section II-C)? To answer this question we need to understand how a typical P300 is altered by a band-pass acquisition filter. Clearly, the shape of the P300 will be significantly deformed. For example, the output must have zero-mean, while the original P300 typically does not. So, we should expect the acquired signal will show ample oscillations around zero. Indeed this is what happens. In Fig. 8, we show the result of applying a software band-pass filter with the same characteristics as the hardware filter in the Mindset24 to the typical P300 reported in [31]. As the figure shows, although the shape of the P300 is considerably distorted, it is matched very well by the shape of the rbio3.3 family of wavelets used in our system. In particular, the similarity with the wavelet with scale $a = 22.5$ and centred at approximately 300 ms is striking (note how these values almost exactly coincide with the peaks of the distributions in Fig. 6). It is then not surprising that, despite the distortions in the P300s, the system could reliably extract relevant control information from these.

As we mentioned in Section II-B, in our experiments we encouraged participants to point their gaze on the fixation cross in the middle of the screen. However, no mechanism to either enforce this or to quantify the degree to which this was achieved was put in place. So, like for many other P300 based systems presented in the literature, one might wonder to what extent the performance of the system depends on gaze shifts and whether the system would work without gaze control. In short, is our BCI system independent (in the sense of not being dependent on peripheral muscles and nerves and relying primarily on central nervous system activity)? We cannot exclude that some components generated by visual pursuit were indeed used in some of our evolved filters. However, the previous two observations (in conjunction with the effective predominant use of channels where P300s are best observed and of scales for the wavelets that match P300 shapes after band-pass deformation) strongly suggest that, indeed, the system is an independent BCI. This is further corroborated by the fact that the system worked well also for the three participants who made a conscious and constant effort to point their gaze to the fixation cross during the experiments.

Further evidence that this is an independent interface comes from the fact that relevant EEG epochs did not show eye movement artefacts, even in frontal channels, which would have been evident had visual pursuit taken place during a trial.

B. Open Problems

The system is accurate in moving the mouse pointer in the standard four directions, particularly over long distances. However, when using computer mice, it is common to move the pointer in diagonal directions. This was achieved in our tests with participant D (not reported), by switching attention from one target to another periodically, so as to obtain a sort of zigzagging movement. Nevertheless, it would be desirable to have a more direct and effective way to move diagonally. An obvious way to achieve diagonal motion would be to add four additional flashing rectangles at the corners of the screen. But, there might be another, more natural solution. Our mouse is analogue and so, in principle, it would not be impossible to achieve oblique movement if a user was able to concentrate his or her attention on two squares (e.g., up and right, right and down, etc.) at the same time, thereby producing P300 components both for horizontal motion and vertical motion. More research on this is needed.

After training and validation, the BCI mouse can be used in tracing mode with immediate feedback. In these conditions the motion of the pointer, which always starts from the middle of the screen, can distract the user, who will occasionally perform visual pursuit to check whether or not the desired motion is achieved. These visual pursuit movements immediately generate myoelectric artifacts in the signals being acquired which result in some, otherwise valid commands, to be misinterpreted. Also, once distracted, users may find it difficult to immediately refocus their attention on the stimuli. In informal testing we have noticed that these effects resulted in a worsening of the ability to control the pointer. The situation may be even worse when the screen contains numerous elements of interest for the user or when the system is used to with a user interface including icons, windows, etc. Future research will need to explore methods to mitigate these effects.

The concentration required to operate our BCI mouse is considerable. We believe its cognitive load is no different from the effort required to operate other BCI systems, but in future research we will need to evaluate the usability of the BCI mouse with purposely designed experiments.

VI. CONCLUSION

We have proposed a BCI mouse based on P300 waves. The system is analogue: at no point a binary decision is made as to whether or not a P300 was actually produced in response to a stimulus. Instead, the motion of the pointer on the screen is controlled by directly combining the amplitudes of the output produced by a filter in the presence of different stimuli.

Beyond providing carefully designed stimuli, a rich set of features and a very flexible combination mechanism through which we thought a solution to the problem of controlling a pointer via EEG could be found, we actually did not do any other design. The biggest part of the design in this system (i.e., the feature selection and the selection of the parameters of the controller) was entirely left to a genetic algorithm. The performance of our system has been very encouraging. All participants have been able to use the system quickly. The GA was effective and efficient at finding good designs for the system. Indeed, it succeed in every run, suggesting that we had chosen the infrastructure

for the system and the feature set reasonably well. In validation, the trajectories of the pointer have achieved high accuracy. The system issues control commands at a much faster rate than other P300-based computer mice previously reported.

These encouraging results indicate that there is a lot more information about user intentions in EEG signals, and that, perhaps, traditional design techniques may be a limiting factor.

REFERENCES

[1] L. A. Farwell and E. Donchin, "Talking off the top of your head: Toward a mental prosthesis utilizing event-related brain potentials," *Electroenceph. Clin. Neurophysiol.*, vol. 70, pp. 510–523, 1988.

[2] J. R. Wolpaw and D. J. McFarland, "An eeg-based brain-computer interface for cursor control," *Electroenceph. Clin. Neurophysiol.*, vol. 78, pp. 252–259, 1991.

[3] G. Pfurtscheller, D. Flotzinger, and J. Kalcher, "Brain-computer interface—A new communication device for handicapped people," *J. Microcomput. Applicat.*, pp. 293–299, 1993.

[4] N. Birbaumer, N. Ghanayim, T. Hinterberger, I. Iversen, B. Kotchoubey, A. Kübler, J. Perelmouter, E. Taub, and H. Flor, "A spelling device for the paralysed," *Appl Opt*, vol. 398, no. 6725, pp. 297–298, Mar. 25, 1999.

[5] A. B. Schwartz, "Cortical neural prosthetics," *Annu. Rev. Neurosci.*, vol. 27, pp. 487–507, Jul. 2004.

[6] S. Kelly, E. Lalor, R. Reilly, and J. Foxe, "Visual spatial attention tracking using high-density SSVEP data for independent brain-computer communication," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 13, no. 2, pp. 172–178, Jun. 2005.

[7] A. Georgopoulos, F. Langheim, A. Leuthold, and A. Merkle, "Magnetoencephalographic signals predict movement trajectory in space," *Exp. Brain Res.*, pp. 1–4, Jul. 2005.

[8] N. Weiskopf, K. Mathiak, S. W. Bock, F. Scharnowski, R. Veit, W. Grodd, R. Goebel, and N. Birbaumer, "Principles of a brain-computer interface (BCI) based on real-time functional magnetic resonance imaging (fmri)," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 966–970, Jun. 2004.

[9] S. Coyle, T. Ward, C. Markham, and G. McDarby, "On the suitability of near-infrared (NIR) systems for next-generation brain-computer interfaces," *Physiol. Meas.*, vol. 25, pp. 815–822, Aug. 2004.

[10] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain-computer interfaces for communication and control," *Clin. Neurophysiol.*, vol. 113, no. 6, pp. 767–791, Jun. 2002.

[11] N. Birbaumer, T. Hinterberger, A. Kubler, and N. Neumann, "The thought-translation device (TTD): Neurobehavioral mechanisms and clinical outcome," *IEEE Trans. Neural Syst. Rehab. Eng.*, vol. 11, no. 2, pp. 120–123, Jun. 2003.

[12] J. R. Wolpaw and D. J. McFarland, "Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans," *Proc. Nat. Acad. Sci. USA*, vol. 101, no. 51, pp. 17 849–17 854, Dec. 2004.

[13] A. Kübler, F. Nijboer, J. Mellinger, T. M. Vaughan, H. Pawelzik, G. Schalk, D. J. McFarland, N. Birbaumer, and J. R. Wolpaw, "Patients with ALS can use sensorimotor rhythms to operate a brain-computer interface," *Neurology*, vol. 64, no. 10, pp. 1775–1777, 2005.

[14] G. R. Müller-Putz, R. Scherer, G. Pfurtscheller, and R. Rupp, "EEG-based neuroprosthesis control: A step towards clinical practice," *Neurosci. Lett.*, vol. 382, pp. 169–174, 2005.

[15] E. W. Sellers and E. Donchin, "A P300-based brain-computer interface: Initial tests by ALS patients," *Clin. Neurophysiol.*, vol. 117, pp. 538–548, 2006.

[16] E. E. Sutter, "The brain response interface: Communication through visually induced electrical brain responses," *J. Microcomp. App.*, vol. 15, pp. 31–45, 1992.

[17] M. Middendorf, G. McMillan, G. Calhoun, and K. S. Jones, "Brain-computer interfaces based on steady-state visual evoked response," *IEEE Trans. Rehabil. Eng.*, vol. 8, no. 2, pp. 211–213, Jun. 2000.

[18] J. Hill, T. Lal, M. Schröder, T. Hinterberger, N. Birbaumer, and B. Schölkopf, "Selective attention to auditory stimuli: A brain-computer interface paradigm," in *Proc. 7th Tübingen Perception Conf.*, 2004, p. 102.

[19] Y. Wang, R. Wang, X. Gao, B. Hong, and S. Gai, "A practical vep-based brain-computer interface," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 14, no. 2, pp. 234–239, 2006.

[20] G. R. Müller, R. Scherer, C. Neuper, and G. Pfurtscheller, "Steady-state somatosensori evoked potentials: Suitable brain signals for brain-computer interfaces?," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 14, no. 1, pp. 30–37, 2006.

[21] E. A. Curran and M. J. Stokes, "Learning to control brain activity: A review of the production and control of EEG components for driving brain-computer interface (BCI) systems," *Brain Cogn.*, vol. 51, pp. 326–336, 2003.

[22] E. Donchin and M. G. H. Coles, "Is the P300 component a manifestation of context updating?," *Behav. Brain Sci.*, vol. 11, pp. 355–372, 1988.

[23] E. Donchin and M. G. H. Coles, "Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans," *Proc. Natl. Acad. Sci.*, vol. 101, no. 51, pp. 17 849–17 854, 2004.

[24] J. P. Donoghue, "Connecting cortex to machines: Recent advances in brain interfaces," *Nat. Neurosci.*, vol. 5, pp. 1085–1088, Nov. 2002.

[25] J. B. Polikoff, H. T. Bunnell, and W. J. B. Jr., "Toward a P300-based computer interface," in *Proc. Rehab. Eng. Assistive Technol. Soc. North America (RESNA'95)*. Arlington, Va: RENSA Press, 1995, pp. 178–180.

[26] F. Beverina, G. Palmas, S. Silvoni, F. Piccione, and S. Giove, "User adaptive bcis: SSVEP and P300 based interfaces," *Psych. J.*, vol. 1, no. 4, pp. 331–354, 2003.

[27] R. Poli, S. Cagnoni, and G. Valli, "Genetic design of optimum linear and non-linear QRS detectors," *IEEE Trans. Biomed. Eng.*, vol. 42, no. 11, pp. 1137–1141, Nov. 1995.

[28] L. Citi, R. Poli, and F. Sepulveda, "An evolutionary approach to feature selection and classification in P300-based BCI," *Biomedizinische Technik*, vol. 49, pp. 41–42, 2004, Proceedings of 2nd International BCI workshop and Training Course.

[29] C. Cinel, R. Poli, and L. Citi, "Possible sources of perceptual errors in P300-based speller paradigm," *Biomedizinische Technik*, vol. 49, pp. 39–40, 2004, Proceedings of 2nd International BCI workshop and Training Course.

[30] L. J. Eshelman and J. D. Schaffer, "Real-coded genetic algorithms and interval schemata," in *Foundations of Genetic Algorithms 2*, L. D. Whitley, Ed. San Mateo, CA: Morgan Kaufmann, 1993.

[31] E. W. Sellers, "A P300-based brain-computer interface: Testing an alternative method of communication," Ph.D. dissertation, Dept. Psychology, College of Arts and Sciences, Univ. South Florida, , 2004.

[32] C. Duncan-Johnson and E. Donchin, "The time constant in P300 recording," *Psychophysiol*, vol. 16, pp. 53–55, 1979.

Riccardo Poli He is a professor in the Department of Computing and Electronic Systems, the University of Essex, Colchester, U.K. His main research interests include genetic programming, particle swarm optimization, the theory of evolutionary algorithms, and biomedical engineering. He has published over 230 refereed papers on evolutionary algorithms, biomedical engineering, neural networks, and image/signal processing. He has coauthored the book "Foundations of Genetic Programming" (Springer, 2002). He has been Chair of numerous international conferences. He is an Associate Editor of the *Evolutionary Computation* journal, the *Genetic Programming and Evolvable Machines* journal and the *International Journal of Computational Intelligence Research*, an advisory board member for the *Journal of Artificial Evolution and Applications* and a member of the editorial board of *Swarm Intelligence*. He has been one of the funding members of the Brain–Computer Interfaces Group at the University of Essex.

He is a Senior Fellow of The International Society for Genetic and Evolutionary Computation (now ACM SIGEVO) and a recipient of the Evo* award for outstanding contributions to the field of evolutionary computation.

Caterina Cinel received the Laurea in general and experimental psychology from the University of Padua, Padua, Italy, in 1996, and the M.Sc. degree in cognitive science and the Ph.D. degree in psychology from the University of Birmingham, Birmingham, U.K., in 1998 and 2002, respectively.

Since 2002, she has been a Senior Research Officer in the Department of Psychology of the University of Essex, Colchester, U.K. She has been one of the members of the Brain-Computer Interfaces Group at the University of Essex since 2003. Her main research interests include attention, multisensory integration and memory, connectionist modelling, and brain–computer interfaces (particularly their psychological aspects).

Luca Citi received the B.Sc. and M.Sc. degrees (*Summa Cum Laude*) in electronic engineering from the University of Florence, Florence, Italy. He is currently working toward the Ph.D. degree in biorobotics at the IMT Institute for Advanced Studies, Lucca, Italy.

His research focuses on noninvasive, independent, brain–machine interfaces for the control of robotic devices, signal processing analysis of neural signals recorded by means of invasive peripheral longitudinal intrafascicular interfaces (LIFEs) for the control of a prosthetic hand, evolving components detector in event related potentials by means of particle swarm optimization and evolutionary methods and application of the former to a P300-based brain–computer interface.

Mr. Citi received the "Giuseppe Francini" prize from the Italian National Bioengineering Group for "an original and innovative graduation thesis in bioengineering."

Francisco Sepulveda (M'99) received the B.Sc. degree in nuclear engineering from the University of California, Santa Barbara (with "The outstanding graduating senior" award), in 1988, the M.Sc. degree in bioengineering from Clemson University, Clemson, SC, in 1990, and the Ph.D. degree (*Summa Cum Laude*) in biomedical engineering from Unicamp, Brazil (with a fellowship at the University of Strathclyde, Glasgow, U.K.), in 1996.

He coordinates the Brain–Computer Interfaces Group at the University of Essex, Colchester, U.K. His interests are in the fields of biomedical signals and systems, machine intelligence, neuroscience, and affective computing. Before moving to Essex, U.K., in 2002, he held academic posts in neural engineering at Aalborg University, Denmark, and at Unicamp, Brazil.