Towards P300-Based Mind-Control: A Non-invasive Quickly Trained BCI for Remote Car Driving

Daniela De Venuto[(✉)], Valerio F. Annese, and Giovanni Mezzina

Department of Electrical and Information Engineering, Politecnico di Bari,
Via Orabona 4, 70125 Bari, Italy
{daniela.devenuto,valeriofrancesco.annese}@poliba.it

Abstract. This paper presents a P300-based Brain Computer Interface (BCI) for the control of a mechatronic actuator (i.e. wheelchairs, robots or even cars), driven by EEG signals for assistive technology. The overall architecture is made up by two subsystems: the Brain-to-Computer System (BCS) and the mechanical actuator (a proof of concept of the proposed BCI is shown using a prototype car). The BCS is devoted to signal acquisition (6 EEG channels from wireless headset), visual stimuli delivery for P300 evocation and signal processing. Due to the P300 inter-subject variability, a first stage of Machine Learning (ML) is required. The ML stage is based on a custom algorithm (t-RIDE) which allows a fast calibration phase (only ~190 s for the first learning). The BCI presents a functional approach for time-domain features extraction, which reduces the amount of data to be analyzed. The real-time function is based on a trained linear hyper-dimensional classifier, which combines high P300 detection accuracy with low computation times. The experimental results, achieved on a dataset of 5 subjects (age: 26 ± 3), show that: (i) the ML algorithm allows the P300 spatio-temporal characterization in 1.95 s using 38 target brain visual stimuli (for each direction of the car path); (ii) the classification reached an accuracy of $80.5 \pm 4.1\%$ on single-trial P300 detection in only 22 ms (worst case), allowing real-time driving. For its versatility, the BCI system here described can be also used on different mechatronic actuators.

Keywords: BCI · Machine Learning · Classification · EEG · ERP · P300

1 Introduction

Only a few years ago the idea of mind controlling a robot or a prosthesis seemed to belong only to science fiction films. Nowadays, the possibility of using brain signals to control external actuators is reality. A Brain-Computer Interface (BCI) is a system providing a direct communication channel between human brain and an external mechanical device, via computer, µPC or FPGA. The functioning principle is based on the detection of specific Brain Activity Pattern (BAP) concurrently to a particular stimuli-based task: the BAP detection related to a particular stimulus expresses the user's intention to perform the command/actuation to whom that stimulus is linked. At the current state of the art, there are four different BAPs widely used for BCI applications:

sensorimotor rhythms (SMR), amplitude modulation of slow cortical potentials (SCP), visual cortex potentials (VEPs) and Event-Related Potentials (ERPs) [1]. The ERP-based BCI systems are mainly based on the P300. Differently from SMR or SCP, a P300-based BCI does not require intensive user training, because the P300 component results from endogenous attention-based brain function. Despite the P300 inter-subject variability in terms of latency and amplitude [2], which makes necessary a phase of Machine Learning (ML), the component is detectable on every cognitively healthy human being. Although W. Grey et al. [3], which developed a mind-controlled cursor, implemented the first BCI system in 1964, the scientific research in this field is experiencing an exponential growth due to the possibility to improve the life quality of paralytic, tetraplegic and motor impaired people. Nowadays, P300-based BCI systems cover a wide range of applications such as locomotion (i.e. wheelchairs [4], robot or neuro-prosthesis [5, 6]), rehabilitation (i.e. the "Bionic Eye" [7]), communication (i.e. the P300 speller [5]), environmental control (i.e. the "Brain Gate" [8]) and entertaining (i.e. neuro-games [8]). In the field of mind-controlled car, Luzheng Bi et al. [9] developed a P300-based BCI for destination selection in vehicles. However, despite the high accuracy (93.6% ± 1.6%), the system does not allow real-time navigation since the selection time is about 12 s. Differently, D. Gohring et al. [10] reported the development of a semi-autonomous mind-controlled car. Although the prototype allows the free-drive mode, the proposed BCI is SMR-based, requiring a very intensive training stage.

In this paper a P300-based BCI for the control of a mechatronic actuator driven by electroencephalographic (EEG) signals for assistive technology is presented. A proof of concept of the proposed BCI is shown using an "ad-hoc" implemented Raspberry-based prototype car. The system has been tested on 5 subjects (age: 26 ± 3). The main innovations of the implemented BCI are: (i) the development of the first P300-based mind-controlled vehicle to be used in free-drive mode; (ii) the adoption of a custom algorithm, t-RIDE [2], for the ML allowing a complete P300 spatio-temporal characterization in only 1.95 s using 38 target brain visual stimuli (for each addressable command) resulting in a very fast training phase; (iii) a functional approach for the real-time classification based on features extraction (FE), combining fast interpretation of the user's intention (worst case: 19.65 ms ± 10.1 ms) and high accuracy in the P300 single-trial detection (80.51 ± 4.1% on 5 subjects). The structure of the paper is described in the following. Section 2 outlines basic knowledge on the P300 component. Section 3 presents the architecture of the implemented BCI, focusing on the (ML and classification algorithms). Section 3.4 describes the development of the prototype car system. Section 4 reports the experimental results performed on 5 subjects. Section 5 concludes the paper with final observations.

2   Evoked Related Potentials: The P300 Component

The P300 is a positive deflection in the human brain event related potentials (ERPs) evoked when a subject is actively and cognitively engaged in the discrimination of one target stimulus by not-target ones, generally denoted as "oddball paradigm" (Fig. 1) [2, 11]. The 'Stimulus' is a single external event (audio, visual, tactile, etc.) delivered

to the subject under test. The target stimulus is the event to be recognized among different ones (not-target). When a target is delivered, the subject under test performs a cognitive task i.e. to count in mind the number of target stimuli occurrences ("no-go task" i.e. response without any muscular movement).
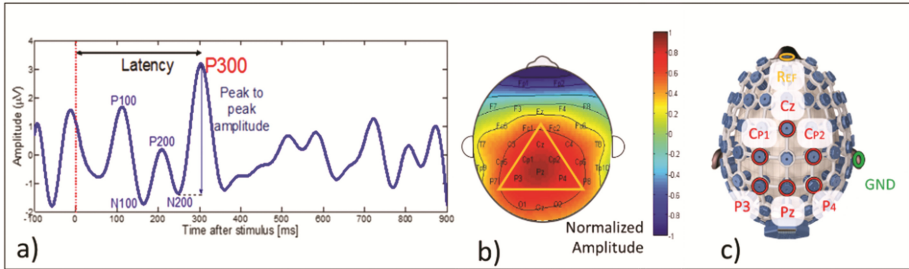

Fig. 1. (a) Time-domain P300 waveform; (b) cortical-area involved by the P300; (c) monitored electrodes. (Color figure online)

A trial is 1 s EEG signal starting from 100 ms before the stimulus delivery. A test is an assemble of stimuli (target and not-target ones) randomly delivered to the subject. In a single test, the probability of target occurrence has to be lower than the not-target one. The P300 characterization is based on its latency, amplitude and cortical-area involved (see Fig. 1). The P300 latency is heavily affected by trial-to-trial variability (P300 jitter) within a given experimental condition and ranges from 290 ms to 447.5 ms [12]. The P300 amplitude is the peak-to-peak amplitude between the previous deflection (N200) and the P300 maximum value (Fig. 1). The P300 amplitude can reach even 37.7 µV depending on the age and on the rarity of the target [12]. The cortical area involved by the P300 is not "a priori" known but, generally, it is the central parietal cortex [12]. The brain mapping of P300 is computed by a topography [2]. Differently form the classic oddball paradigm [2, 13, 14], for BCI applications more than two stimuli are delivered with uniform probability distribution. Each stimulus is linked to a particular actuation command. The subject select a single stimulus to be considered as the target and performs the cognitive task only on that particular stimulus according to his intentionality and to the command he wants to address. Only that particular stimulus will evocate the P300. The P300 detection on a particular stimulus allows understanding that the subject was focusing on that particular stimulus and that he wants to address the linked actuation command.

3 Overall Architecture

The overall architecture (Fig. 2) can be divided into two subsystems: the Brain Computer System (BCS) and the Prototype Car System (PCS) connected together through a Client-Server TCP/IP communication protocol. The BCS composed by the hardware acquisition system (wireless EEG headset station) and by the data processing algorithms aiming to interpret the user intentions. Due to the inter-subject variability, the system needs to

be tuned on the particular user through a first stage of Machine Learning (ML). A custom algorithm, the tuned-Residue Iteration Decomposition (t-RIDE) [2, 13, 14], which characterizes the P300 waveform and extracts all the parameters to be used for the real time classification stage, performs the ML. A linear hyper-dimensional classifier combining low computational efforts and high accuracy basing on a stage of Feature Extraction (FE) performs the real-time P300 detection. When the classifier recognizes and validates the user intention, the BCS communicates the actuation to be performed to the PCS. The PCS is made up by an acrylic prototype car, equipped by a camera, two DC motors, three servomotors (two for camera and one for steering), managed by a programmed μPC (Raspberry Pi 2 model B+). Afterwards, PCS runs a pre-compiled set of scripts that drive the steering, the camera servomotors and the DC motors. The PCS sends a real-time video to the BCS while receiving command for the control of DC motors and servomotors. A neurophysiological protocol allows the interaction between human brain and BCS by using an "ad-hoc" locally generated visual stimulation that evocates the P300 potential.
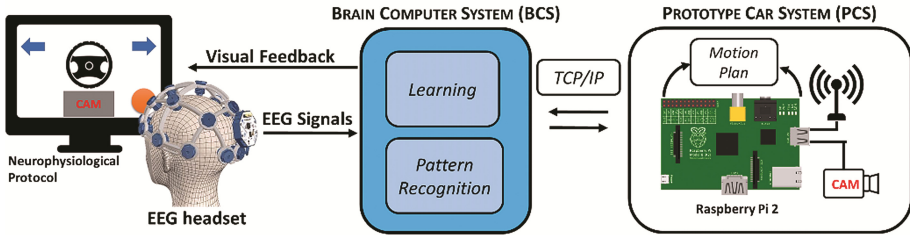

Fig. 2. Schematic overview of the developed BCI architecture.

3.1   Neurophysiological Protocol and Hardware Instrumentation

The adopted data acquisition hardware is a 32-channels wireless EEG headset with active electrodes (conditioning integrated circuit are embedded in the electrode performing amplification, filtering and digitalization) as the one used in [15–21]. According to the international 10–20 standard, the EEG recordings have been performed using six electrodes (C_z, CP_1, CP_2, P_3, P_z, P_4 – in red in Fig. 1c) referenced to AF_z electrode (in orange in Fig. 1c) and right ear lobe (A_2 – in green in Fig. 1c) is used as ground. The locations of the electrodes have been selected according to previous P300 studies [2, 13, 14]. EEG signals are recorded with sampling frequency of 500 Hz, 24-bit resolution, ±187.5 mV input range and filtered using a bandpass (Butterworth, 8th order 0.5–30 Hz) and power line notch filtered (Butterworth, 4th order 48–52 Hz - embedded into the front-end) [22–27]. The recording scheme is monopolar. The EEG signals are recorded while the user performs the neurophysiological protocol. The neurophysiological protocol is made up by 4 visual stimuli, individually and randomly flash on a display (25% occurrence probability) with an inter-stimuli time of 1 s. In the developed neurophysiological protocol there is no pre-defined target stimulus: the stimulus on which the user freely focuses his attention became the target (selective attention).

Each stimulus persists on the screen for 200 ms. Stimuli generation/delivered are driven by a Simulink model using a random numeric signal.

Each stimulus is related to a P300 latency and amplitude are related to the saliency of stimulus (in term of color, contrast, brightness, duration, etc.): the BCS associates to each direction different shapes and colors, in sharp contrast to each other. Figure 3 shows a sequence of driving environment snapshots provided by the BCS to the user. For instance, if the user wants to turn right, the target items is the one in Fig. 3c, while the other blinking stimuli are the not-target. EEG data collection, stimuli generation/ delivery and signal processing are performed by the BCS. In this implementation, the BCS has been assumed as a PC (Intel i5, RAM 8 GB, 64 bit).
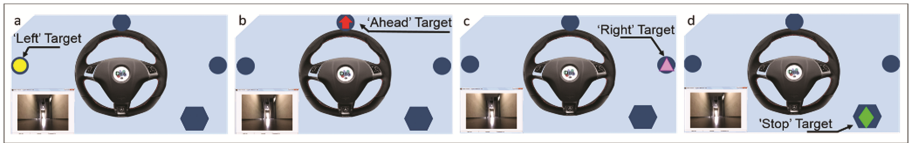

Fig. 3. The neurophysiological protocol is made up by 4 stimuli individually and randomly flashing on a display. Each stimulus is related to a particular command i.e. turn left (a), turn right (c), go ahead (b) and stop (d).

3.2   Machine Learning

The first step of the adopted pattern recognition strategy is to train the system via offline experiment with a learning stage. There are a number of methods reported in literature for P300 detection in single-trial and averaged-trials environments. The main challenge in this field is that the P300 is generally submerged by artifacts and background EEG, resulting in an extremely low Signal to Noise Ratio (SNR). The implemented ML is based on a novel algorithm based on the RIDE approach: the Tuned-Residue Iterative Decomposition (t-RIDE). A detailed description and a deep comparison between t-RIDE and other method at the state of the art (RIDE, ICA, PCA, Grand Average) are reported in [2]. The ML can be divided into four stages: pre-processing, P300 characterization, features extraction (FE) and weights/thresholds definition.

Pre-processing. This stage is intended to reduce the sources of noise, artifacts such as eye movements and head movements preserving the P300 and eliminating the critical issue affecting RIDE. Pre-processing is performed for each monitored channel. The acquired EEG signal is further low-pass filtered (Butterworth, 6th order, fstop = 15 Hz) and aligned to the stimulus signal. Subsequently, the EEG signal is decomposed in trial of 1 s, each epoch starts 100 ms before the rising edge of the stimulus (target and non-target). Trials are fitted in a 6th order polynomial. The resulting fitted curve is subtracted from the EEG signal, which is then centered (offset cancellation) and normalized. Finally all the trials are organized into a 3D matrix **DATA** $\in R^{S \cdot N \cdot M}$ where S is the number of samples into 1 trial (500 in our implementation), N is the number of monitored channels (6 in our work) and M is the number of delivered stimuli (target and not-target).

P300 Characterization. This stage is based on the custom algorithm t-RIDE [2, 13, 14]. The RIDE approach [2] is a multi-purpose method for ERP extraction and, for its generality, does not take advantages from the "a priori" information about P300. This leads to the necessity of a specialized staff setting static calculation parameters for RIDE (i.e. the computation window) as well as the impossibility to follow the latency modulation of the P300 component. Furthermore, RIDE includes neither pre-processing (which is application-dependent) nor spatial considerations. t-RIDE is a custom tuned version of RIDE for P300 extraction which includes three additional steps to RIDE: pre-processing, window optimization and spatial characterization [2, 13, 14]. Starting from a default window, t-RIDE automatically calculates the optimized one. Only one signal derived from the average of P_Z and C_Z is considered for window optimization. A first default window is defined as 250–400 ms after the stimulus and the default time-step increments are set to 4 ms and 8 ms for starting point and end point of the window, respectively. The procedure for window optimization is described in the following. The number of iterations for an optimal tuning phase is given by:

$$n_{IT} = \left\lceil \frac{t_{lim} - t_{e,win}}{t_{sh,r}} \right\rceil \tag{1}$$

where t_{lim} is the upper limit, $t_{e,win}$ is the selected window end time, $t_{sh,r}$ is the right-shift parameter. Therefore, n_{IT} different windows are considered, sweeping the entire time slot where the P300 is expected. By default configuration, the iteration cycle starts from a fixed window, then its start/end points are progressively right shifted by 4 ms and 8 ms respectively. Thus, the last window considered in the computation is 278 ms–456 ms. For each considered window, the RIDE approach extracts the P300 time-domain waveform. After n_{IT} iterations, the algorithm selects the optimized window i.e., the one with the highest P300 amplitude. As soon as the optimized window is defined, the **DATA** matrix is processed by t-RIDE. Time-domain results from each channel are subsequently interpolated in order to extract the spatial characterization (P300 topography). In this stage, the system stores the vectors $\mathbf{L} = (l_1,\ldots, l_N) \in R^N$ and $\mathbf{A} = (a_1,\ldots, a_N) \in R^N$ containing respectively the expected P300 latencies and amplitude for each channel (N = 6).

Feature Extraction (FE). The t-RIDE extracted P300 pulse undergoes a phase of FE to be used as 'golden reference' by the classifier. According to specialized medical staff P300 visual inspection guidelines, five features have been selected. They take into account the time-domain P300 shape information, exalting the differences between target and not-target typical trends. For the FE on the jth channel, the trial x(i) is windowed by a rectangular 200 ms window (number of samples ns = 100) centered on the expected latency l_j The extracted features are (see Fig. 4):
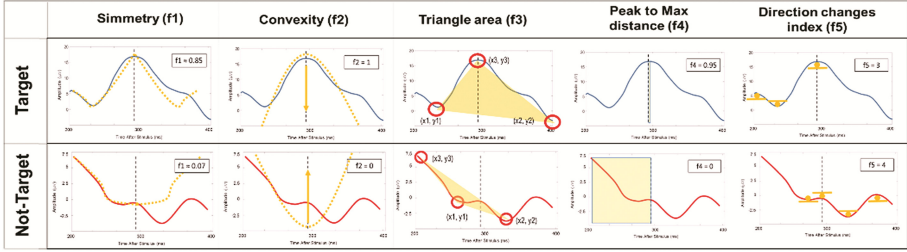
Fig. 4. Calculation example of the features set for target and not-target stimuli.

1. The **Simmetry** quantifies the symmetry degree of the signal with respect to the expected latency:

$$f_1 = 1 - \left| \frac{2}{ns-1} \sum_{i=1}^{ns} [x(i) - x(ns-i)] \right| \qquad (2)$$

2. The **Convexity** identifies the convexity degree of the considered data points with respect to the expected latency:

$$f_2 = 1 \leftrightarrow \sum_{i=1}^{\left(\frac{ns}{2}\right)-1} \frac{\partial x(i)}{\partial i} \geq \sum_{i=\left(\frac{ns}{2}\right)+1}^{ns} \frac{\partial x(i)}{\partial i}; otherwise \quad f_2 = 0 \qquad (3)$$

3. The **Triangle area (TA)** delivers the area of the triangle inscribed into the potentially P300 component deflection:

$$f_2 = 0.5 \cdot \begin{vmatrix} x1 & y1 & 1 \\ x2 & y2 & 1 \\ x3 & y3 & 1 \end{vmatrix} \qquad (4)$$

where (x1, y1) is the minimum value in the 100 ms before the P300 learned latency as well as (x2, y2) is the minimum value of the 100 ms on its right side. (x3, y3) are the coordinates of maximum value of the extracted data points.

4. The **Peak to Max distance (PMD)** quantifies how close is the maximum point of the single trial with respect to the expected one:

$$f_4 = \left\{ \frac{(ns+1)}{2} - \left| \frac{(ns+1)}{2} - index(\max(x)) \right| \right\} \frac{2}{(ns+1)} \qquad (5)$$

5. The **Direction changes index (DCI)** quantifies the number of considered waveform direction changes. It can be obtained by counting the slope sign changes, referring to signal derivative.

Weights/Thresholds Definition. The FE is performed offline, on the same acquired raw data and performed on single-trials (for both targets and not-targets) i.e. on the

DATA matrix. A statistical analysis elaborates the features distributions and extracts the 25th and 75th percentiles and median value for each feature. While the percentiles are used for thresholds definition, the median values allow the determination of a set of weights (one set for each channel) to be used by the classifier. They are assigned considering the subtraction between the median value of the j-th feature vector referred to the target responses and the not-target ones. The ML orders their values in descending order and assigns 0.3 to the best feature and 0.1 to the worst one. The other 3 features have weight that decreasing from 0.3 to 0.1 with 0.05 steps. These values allow obtaining a sum that provides a maximum of 1. Finally, the responsivity of each monitored channel is evaluated in order to define a further set of spatial weights.

Summarizing, at the end of the ML, the following subject-depending parameters have been learned by the BCS:

1. **UP** $\in R^{5\times6}$: its generic element $up_{i,j}$ contains the upper thresholds for i-th feature (the 75th percentile) referred to the j-channel.
2. **DN** $\in R^{5\times6}$: its generic element $dn_{i,j}$ contains the lower thresholds for i-th feature (the 25th percentile) referred to the j-channel.
3. **W** $\in R^{5\times6}$: its generic element $w_{i,j}$ is the i-th weight referred to the j-channel.
4. **S** $\in R^6$ contains the indications about the responsivity of the channels.
5. **L** $\in R^6$ contains the expected P300 latencies for each channel.
6. **A** $\in R^6$ contains the expected P300 amplitude for each channel.

3.3   Real-Time Hyper-Dimensional Classification

For each subject, the "golden" learned P300 can be represented by a single point in a n-dimensional space where the features are the bases (n = number of features). The real-time classifier performs a FE on incoming streaming EEG data and compares the results with the "golden" reference: the decision about the absence/presence of P300 is based on the n-dimensional distance between reference and incoming trial exploiting thresholds. In order to reduce the computational times, the classification is performed on a down-sampled (from 500 sps to 100 sps) and windowed (M samples centered on the expected latency, M = 20–200 ms) version of EEG trials. The first step is data validation (*1st classifier rule*): in order to avoid that artifacts affect the results, data are validated only if they are similar in amplitude to the values in **A**. As soon as a new stimulus occurs, the BCS performs the FE on the single-trial for each channel basing on **L**. This leads to the computation of $\mathbf{f} \in R^{5\times6}$ where its generic element $f_{i,j}$ expresses the value of the i-th feature on the j-th channel. The classifier adopts the following decisional rule (*2nd classifier rule*):

$$F_{i,j} = \begin{cases} 0 & \leftrightarrow f_{i,j} < dn_{i,j} \\ 0.5 & \leftrightarrow dn_{i,j} \leq f_{i,j} \leq up_{i,j} \\ 1 & \leftrightarrow f_{i,j} > up_{i,j} \end{cases} \qquad (6)$$

This procedure leads to the creation of the matrix $\mathbf{F} \in R^{5\times6}$. A weighted sum of $\mathbf{F}$ defines the presence of the P300 on the j-th channel, through the calculation of the vector $R \in R^6$, where the generic element is (*3rd classifier rule*):

$$r_j = w_{1,j} \cdot F_{1,j} + \dots + w_{5,j} \cdot F_{5,j} \text{ for } j = 1, 2, \dots, 6 \tag{7}$$

Afterwards, the classifier adopts the *4th decisional rule* to evaluate the presence/absence of P300 on the j-th channel:

$$y_j = \begin{cases} 0 \leftrightarrow r_j \leq y_t \\ 1 \leftrightarrow r_j > y_t \end{cases} \tag{8}$$

Where y_j is the generic element of $\mathbf{Y} \in R^6$ and y_t is a decision threshold set to 0.5. The decision rule with $y_t = 0.5$ means that on the j-th channel, at least 3 features have been detected. The classification ends with the spatial validation (*5th classifier rule*): the classifier validate the P300 presence only if the P300 is simultaneously detected on 5 out of 6 channels detect w.r.t. channels, which deliver a high detection rate (vector $\mathbf{S}$). When the classification is over, the BCS sends to the PCS a 2 bits code informing Raspberry about the actuation through TCP/IP wireless communication.

3.4   The Prototype Car System (PCS)

Two 3.7 V batteries (Panasonic 18650) deliver a nominal supply voltage of 7.4 V. Through the DC-DC converter (XL-1509), this voltage is converted into a 5 V stable power supply for the entire prototype car. Raspberry Pi 2 (Model B+), equipped by a Wi-Fi antenna, a USB camera and a SD card, is the control unit of the prototype car. Raspberry controls three main aspects of the navigation i.e. obstacle detection and avoidance, servomotor orientation (both for car driving both for camera positioning) and DC motor power control. The obstacle detection and avoidance is based on three ultrasonic proximity sensors (HC-SR04) which point in three different directions (straight and sideways). The 'Trig' pins of the ultrasonic sensors are driven by a single output Raspberry pin and triggered at 10 Hz with a pulse wave of 10 µs. The sensors response delivered on their 'Echo' pin are connected to three different Raspberry input pins. An in-loco running python script continuously monitors the presence/absence of obstacles: when an obstacle is detected ahead with a distance lower than 50 cm, the prototype car stops. Differently, when the prototype car detect a side obstacle that is not in its trajectory, Raspberry alert the BCS, which adapt the neurophysiological protocol, which does not propose to curve on that direction. The DC motor power control is managed by Raspberry Pi using an h-bridge (L298N): GPIO pins controls the enable pins of the L298N ('ena', 'enb') using pulse with modulation (PWM). The motor power control depends on the directives sent by the BCS. Servomotor orientation is controlled by PWM using the PWM module (PCA 9685). There are three servomotors: while the first one manages the prototype car direction (and is controlled by the BCS), the other ones control the orientation of the USB camera, which streams a real-time video to the BCS.

4   Experimental Results

The entire architecture has been tested on a dataset from 5 different subjects (age: 26 ± 3). The subjects performed at first the learning protocol and, subsequently, the real-time prototype car control. The P300 amplitude range was 3–8 µV with a mean value of 4.7 µV $\pm$ 0.61 µV; the P300 latency was included in the range 300–403 ms, with a mean value of 349.25 ms $\pm$ 35.52 ms. Table 1A shows the subject-by-subject topographies of the amplitudes (for both target and not-target stimuli), latencies and time-domain waveform achieved by the t-RIDE algorithm. From the latency topography it is shown that the P300 is detected from the lateral mid-line electrodes (200–250 ms on P3 and P4) and the central electrodes (Fz, Cz, Pz) 300–400 ms after stimulus. Additionally, from the amplitude topographies it is clear that despite the subject-by-subject P300 variability, this component his deeply suitable for binary discrimination. The complete results of the ML stage are presented in Table 1B where, for each subject and feature are express as medium value $\pm$ std. deviation computed on all the monitored channels. The online validation approach included two different tests: (i) single direction repetitive selections and (ii) pattern recognition. In the first approach, the user is asked to select repeatedly a single target. The reached classification accuracies computed in these conditions are (see Table 2A): sub1: 73.68 $\pm$ 5.3%; sub2: 83.71 $\pm$ 4.6%; sub3: 80 $\pm$ 3.1%; sub4: 81.30 $\pm$ 4.8%; sub5: 83.84 $\pm$ 5.8%. The best classification accuracy is 89% while the worst one is 68.38%.

Table 1. (A) P300 subject-by-subject P300 spatio-temporal characterization; (B) detailed subject-by-subject trained parameters.

Table B		S. 1	S. 2	S. 3	S. 4	S. 5
A (µV)		7.2±0.7	5.2±0.4	4.3±0.57	4,1±0.93	4±1.06
L (ms)		360±107	340±10	330±47	330±7	300±45
f1	UP	0.68±0.01	0.68±0.03	0.68±0.03	0.67±0.03	0.65±0.02
	DN	0.43±0.09	0.23±0.07	0.19±0.10	0.43±0.14	0.30±0.08
	W	0.15	0.15	0.15	0.15	0.15
f2	UP	0.5±0	0.5±0	0.5±0	0.5±0	0.5±0
	DN	0.5±0	0.5±0	0.5±0	0.5±0	0.5±0
	W	0.3	0.3	0.3	0.3	0.3
F3 (µV·ms)	UP	125.5±24.8	84.9±18.1	65.6±12.9	44.5±11.9	38.7±8.8
	DN	39.6±12.3	44.8±2.2	19.7±1.9	18.8±1.9	26.6±6.86
	W	0.25	0.2	0.25	0.25	0.2
f4 (ms)	UP	0.31±0.08	0.31±0.14	0.35±0.23	0.33±0.07	0.33±0.12
	DN	0.11±0.04	0.07±0.05	0.15±0.07	0.25±0.03	0.24±0.03
	W	0.10	0.10	0.10	0.10	0.10
f5	UP	7.8±0.41	7.7±0.82	8±0.63	7.17±0.41	6.17±0.75
	DN	1.67±0.81	2.2±0.41	2.33±0.81	4.00±1.26	4.50±1.38
	W	0.2	0.252	0.2	0.2	0.15

The analysis demonstrated a channel-to-channel accuracy modulation: Table 2A highlights the subject-by-subject highest accuracy.

Differently, the pattern recognition test consists in the selection of a known stream of directions. The reference pattern to be perform by the user was made up by 10 commands covering all the addressable directions. The users performed this test more than once. The performed accuracies computed in this test are: sub1: 67.9 $\pm$ 6.7%; sub2: 72.5 $\pm$ 7.1%; sub3: 67.5 $\pm$ 4.4%; sub4: 69.2 $\pm$ 8.5%; sub5: 70.6 $\pm$ 3.7%. The best pattern recognition is 8/10 of the pattern (80%). The worst response is detecting 5/10 of the

Table 2. (A) P300 subject-by-subject P300 spatio-temporal characterization; (B) detailed subject-by-subject trained parameters.

TABLE A	CLASSIFICATION ACCURACY			
	SINGLE SELECTION		PATTERN RECOGNITION	
SUB	ACCURACY (%)	BEST ELECTRODE	ACCURACY (%)	BEST ELECTRODE
1	73.7 ± 5.3	Pz (87%)	67.9 ± 6.7	Pz (74%)
2	83.7 ± 4.6	P4 (91%)	72.5 ± 7.1	P4 (79%)
3	80 ± 3.1	Cz (86%)	67.5 ± 4.4	Cz (71%)
4	81.3 ± 4.8	C3 (95%)	69.2 ± 8.5	C3 (77%)
5	83.84 ± 5.8	Cz (89%)	70.6 ± 3.7	Cz (74%)

TABLE B	LEARNING COMPUTED ON SUB.				
CLASSIFICATION COMPUTER ON SUB.	1	2	3	4	5
1	75.4	36.84	21.35	34.40	30.17
2	47.50	85.1	34.57	9.85	11.25
3	12.41	32.71	79.3	17.23	27.87
4	40.20	10.75	18.21	82.3	21.61
5	13.13	31.9	26.50	24.47	84.8

pattern (50%). The most accurate channel for each subject is the same as the one resulted from the previous test, although with lower accuracy.

In order to highlight the necessity of the ML, in table B the classification accuracy of each subject basing on the learned parameters of the other subjects is presented. It is worth to notice that the procedure of classification basing on non-optimized thresholds for that specific subject does not maximize the accuracy of the P300 detection: the preponderance of the diagonal elements with respect to the off-diagonal ones highlights the high degree of subjectivity of the classification. In fact, the classification performed using sets of erroneous learning (such as those of others) exhibit low accuracies. Thus, table B highlights the need for a phase of learning on the specific subject.

Since the application is in real-time, special attention should be devoted to timing in order to guarantee the correct functioning. The advantages of t-RIDE respect the state of the art in terms of computational speed have been already discussed in [2]: t-RIDE allows to drastically reduce the duration of the training since it needs only 38 target stimuli for a complete characterization of P300 (190 s). The shortening of the training phase allows reducing the effect of the "habituation" which spoils the P300. t-RIDE computational time was only 1.95 s (against ICA: 3.1 s on the same dataset) [2] and it does not require a minimum number of channels. The fixed communication latency from EEG headset and gateway is 14 ms. The classifier needs to buffer 1 s data after the stimulus in order to perform the computation. The worst-case computational time for each feature extraction on single channel and single trial was 0.653 ± 0.32 ms. The worst-case total time for the FE stage on 6 channels was 19.58 ± 9.7 ms. The successive definition of the matrix F was performed in 0.026 ± 0.011 ms on all the channels. The computational time (for 6 channels) for the spatial validation was 0.041 ± 0.008 ms. Given this computational details, the worst-case total time needed by the classifier to complete the classification for all the channels was 19.65 ± 10.1 ms. The FE stage is the most time consuming part of the process. The communication time between BCS and PCS takes about 3.35 ns (only 2 bits to be sent by Wi-Fi). As soon as Raspberry Pi receives the command, the actuation is performed in 3 ms (worst-case). The overall architecture completes a single actuation (from EEG raw data acquisition triggered by stimulus delivery to PCS actuation) in 1.03 s (worst-case).

5   Conclusion

The aim of the present work has been the study, the design, implementation and test of a brain-computer interface based on P300. The implemented neural interface allows remote control of a generic mechanical device such as a limb, a wheelchair or a robot. In order to validate the above BCI, in the present work the BCI has been applied to the remote control of a prototype vehicle, properly realized. The complete system architecture can be divided into two subsystems: the "Brain-to-Computer System" (BCS) and the "Prototype Car System" (PCS), communicating via TCP/ IP connection.

In order to adapt the system on the user, a first stage of Machine Learning (ML) is needed. The ML stage is based on the custom algorithm t-RIDE, which trains the following hyper-dimensional real-time classifier.

In order to satisfy real-time constraints, a linear thresholds classifier performs in real time the FE on raw data and detects the presence/absence of P300 basing on the learned references. The system has been validated on a dataset of 5 subjects driving the prototype car by their mind. The average classification accuracy on a single direction was $80.51 \pm 4.1\%$. The average classification accuracy in the detection of a 10-direction pattern was $69.6 \pm 1.9\%$. The classifier completes its process on all the channels in 19.65 ± 10.1 ms (worst-case).

References

1. Fernando, L., Alonso, N., Gomez-Gil, J.: Brain computer interfaces, a review. Sensors **12**(2), 1211–1264 (2012)
2. De Venuto, D., Annese, V.F., Mezzina, G.: Remote neuro-cognitive impairment sensing based on P300 spatio-temporal monitoring. IEEE Sensors J. **PP**(99) (2016). doi:10.1109/JSEN. 2016.2606553. Article no. 7562544
3. Graimann, B., Allison, B., Pfurtscheller, G.: Brain–Computer Interfaces: A Gentle Introduction. Brain-Computer Interfaces, pp. 1–27. Springer, Heidelberg (2009)
4. Grychtol, B., et al.: Human behavior integration improves classification rates in real-time BCI. Neural Syst. Rehabil. Eng. **8**(4), 362–368 (2010)
5. Ortner, R., et al.: An SSVEP BCI to control a hand orthosis for persons with tetraplegia. IEEE Trans. Neural Syst. Rehabil. Eng. **19**(1), 1–5 (2011)
6. Hochberg, L.R., et al.: Neuronal ensemble control of prosthetic devices by a human with tetraplegia. Nat. J. **442**, 164–171 (2006)
7. Stacey, A., Li, Y. Barnes, N.: A salient information processing system for bionic eye with application to obstacle avoidance. In: 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Boston, MA, pp. 5116–5119 (2011).doi: 10.1109/IEMBS.2011.6091267
8. Nijholt, A.: BCI for games: a 'state of the art' survey. In: International Conference on Entertainment Computing. Springer, Heidelberg (2008)
9. Bi, L., et al.: A head-up display-based P300 brain–computer interface for destination selection. IEEE Trans. Intell. Transp. Syst. **14**(4), 1996–2001 (2013)
10. Göhring, D., Latotzky, D., Wang, M., Rojas, R.: Semi-autonomous car control using brain computer interfaces. In: Lee, S., Cho, H., Yoon, K.J., Lee, J. (eds.) Intelligent Autonomous Systems 12. Advances in Intelligent Systems and Computing, vol. 194, pp. 393–408. Springer, Heidelberg (2013)

11. De Tommaso, M., Vecchio, E., Ricci, K., Montemurno, A., De Venuto, D., Annese, V.F.: Combined EEG/EMG evaluation during a novel dual task paradigm for gait analysis. In: Proceedings of 2015 6th IEEE International Workshop on Advances in Sensors and Interfaces, IWASI, pp. 181–186 (2015). doi:10.1109/IWASI.2015.7184949. Article no. 7184949

12. Dinteren, V., et al.: P300 development across the lifespan: a systematic review and meta-analysis. PLoS One **9**(2), e87347 (2014)

13. De Venuto, D., Annese, V.F., Mezzina, G., Ruta, M., Di Sciascio, E.: Brain-computer interface using P300: a gaming approach for neurocognitive impairment diagnosis. In: Proceedings of 2016 IEEE HLDVT, Santa Cruza, USA (2016). doi:10.1109/HLDVT.2016.7748261. ISBN 978-1-5090-4270-8

14. Annese, V.F., Mezzina, G., De Venuto, D.: Towards mobile health care: neurocognitive impairment monitoring by BCI-based game. In: Proceedings of IEEE SENSORS 2016, Orlando, USA (2016). doi:10.1109/ICSENS.2016.7808745. ISBN 978-1-4799-8287-5

15. Annese, V.F., De Venuto, D.: Gait analysis for fall prediction using EMG triggered movement related potentials. In: Proceedings of 2015 10th IEEE International Conference on Design and Technology of Integrated Systems in Nanoscale Era, DTIS (2015). doi:10.1109/DTIS.2015.7127386. Article no. 7127386

16. De Venuto, D., Annese, V.F., Ruta, M., Di Sciascio, E., Sangiovanni Vincentelli, A.L.: Designing a cyber-physical system for fall prevention by cortico-muscular coupling detection. IEEE Des. Test **33**(3), 66–76 (2016). doi:10.1109/MDAT.2015.2480707. Article no. 7273831

17. Annese, V.F., De Venuto, D.: FPGA based architecture for fall-risk assessment during gait monitoring by synchronous EEG/EMG. In: Proceedings of 2015 6th IEEE International Workshop on Advances in Sensors and Interfaces, IWASI 2015, pp. 116–121 (2015). doi:10.1109/IWASI.2015.7184953. Article no. 7184953

18. Annese, V.F., De Venuto, D.: Fall-risk assessment by combined movement related potentials and co-contraction index monitoring. In: Proceedings of IEEE Biomedical Circuits and Systems Conference: Engineering for Healthy Minds and Able Bodies, BioCAS (2015). doi:10.1109/BioCAS.2015.7348366. Article no. 7348366

19. Annese, V.F., Crepaldi, M., Demarchi, D., De Venuto, D.: A digital processor architecture for combined EEG/EMG falling risk prediction. In: Proceedings of the 2016 Design, Automation and Test in Europe Conference and Exhibition, DATE 2016, pp. 714–719 (2016). Article no. 7459401

20. Annese, V.F., De Venuto, D.: The truth machine of involuntary movement: FPGA based cortico-muscular analysis for fall prevention. In: 2015 IEEE International Symposium on Signal Processing and Information Technology, ISSPIT 2015, pp. 553–558 (2015). doi:10.1109/ISSPIT.2015.7394398. Article no. 7394398

21. De Venuto, D., Annese, V.F., Sangiovanni-Vincentelli, A.L.: The ultimate IoT application: a cyber-physical system for ambient assisted living. In: Proceedings of IEEE International Symposium on Circuits and Systems, July 2016, pp. 2042–2045 (2016). doi:10.1109/ISCAS.2016.7538979. Article no. 7538979

22. De Venuto, D., Carrara, S., Riccò, B.: Design of an integrated low-noise read-out system for DNA capacitive sensors. Microelectron. J. **40**(9), 1358–1365 (2009). doi:10.1016/j.mejo.2008.07.071

23. De Venuto, D., Castro, D.T., Ponomarev, Y., Stikvoort, E.: Low power 12-bit SAR ADC for autonomous wireless sensors network interface. In: 3rd International Workshop on Advances in Sensors and Interfaces, IWASI 2009, pp. 115–120 (2009). doi:10.1109/IWASI.2009.5184780. Article no. 5184780

24. De Venuto, D., Ohletz, M.J., Ricco, B.: Automatic repositioning technique for digital cell based window comparators and implementation within mixed-signal DfT schemes. In: Proceedings of International Symposium on Quality Electronic Design, ISQED, January 2003, pp. 431–437 (2003). doi:10.1109/ISQED.2003.1194771. Article no. 1194771
25. De Venuto, D., Ohletz, M.J., Riccò, B.: Digital window comparator DfT scheme for mixed-signal ICs. J. Electron. Test. Theory Appl. (JETTA) **18**(2), 121–128 (2005). doi:10.1023/A:1014937424827
26. De Venuto, D., Ohletz, M.J., Riccò, B.: Testing of analogue circuits via (standard) digital gates. In: Proceedings of International Symposium on Quality Electronic Design, ISQED, January 2002, pp. 112–119 (2002). doi:10.1109/ISQED.2002.996709. Article no. 996709
27. De Venuto, D., Vincentelli, A.S.: Dr. Frankenstein's dream made possible: implanted electronic devices. In: Proceedings of Design, Automation and Test in Europe, DATE, pp. 1531–1536 (2013). doi:10.7873/DATE.2013.311. Article no. 6513757