

Article

A Novel Dictionary-Driven Mental Spelling Application Based on Code-Modulated Visual Evoked Potentials

Felix Gemblér  and Ivan Volosyak * 

Faculty of Technology and Bionics, Rhine-Waal University of Applied Sciences, 47533 Kleve, Germany; felix.gembler@hochschule-rhein-waal.de

* Correspondence: ivan.volosyak@hochschule-rhein-waal.de

Received: 9 April 2019; Accepted: 26 April 2019; Published: 30 April 2019



Abstract: Brain–computer interfaces (BCIs) based on code-modulated visual evoked potentials (c-VEPs) typically utilize a synchronous approach to identify targets (i.e., after preset time periods the system produces command outputs). Hence, users have only a limited amount of time to fixate a desired target. This hinders the usage of more complex interfaces, as these require the BCI to distinguish between intentional and unintentional fixations. In this article, we investigate a dynamic sliding window mechanism as well as the implementation of software-based stimulus synchronization to enable the threshold-based target identification for the c-VEP paradigm. To further improve the usability of the system, an ensemble-based classification strategy was investigated. In addition, a software-based approach for stimulus on-set determination is proposed, which allows for an easier setup of the system, as it reduces additional hardware dependencies. The methods were tested with an eight-target spelling application utilizing an n -gram word prediction model. The performance of eighteen participants without disabilities was tested; all participants completed word- and sentence spelling tasks using the c-VEP BCI with a mean information transfer rate (ITR) of 75.7 and 57.8 bpm, respectively.

Keywords: brain–computer interface (BCI); electroencephalogram (EEG); visual evoked potentials (VEP); code-modulated visual evoked potentials (c-VEP)

1. Introduction

A brain–computer interface (BCI) records, analyzes and interprets brain activity of the user and can be used for communication with the external environment, without involving muscle activity [1]. BCIs can be utilized as communication device for severely impaired people; e.g., people suffering from spinal cord injuries, brain stem strokes, amyotrophic lateral sclerosis (ALS), or muscular dystrophies [2]. If used as a spelling device, character output speed and classification accuracy are the most important characteristics of the system.

Code-modulated visual evoked potentials (c-VEPs) have gathered increasing research interest in the field of Brain–Computer Interfaces (BCIs) [3–6]. In a c-VEP application, a set of flickering targets, each associated with a specific binary code pattern, that determines whether the stimulus is displayed or not displayed, is presented to the user. In parallel, the user’s brain signals are recorded, typically, via electroencephalography (EEG). For classification, the system makes use of target-specific EEG templates, which have been pre-recorded in a training session. When the BCI user gazes at one of the targets, the program compares the collected EEG data to the templates and produces an output command.

Usually, time lags of an m -sequence, a type of pseudo-random code sequence with desirable autocorrelation properties, are used for stimulus modulations [7]. In the field of BCIs, m -sequences with

a code length of 63 bits are most popular; this code length is suitable for multi-target implementations on 60 Hz monitors (allowing a stimulus duration of $63/60 = 1.05$ s).

In terms of implementation, synchronization between the amplifier and stimulus presentation is required as the lag between stimuli can be as low as the inverse of the monitor refresh rate. Hence, stimulus onset markers are typically sent to the EEG hardware. These timestamps can be acquired using a photo-resistor or photo-diode attached to the screen [4,8]. Another approach is to send the timestamps from the stimulation computer to the amplifier using the parallel port [3].

In this article, a purely software-based approach is proposed, allowing the detection of stimulus onset without the need for additional hardware.

Typical use cases of c-VEP BCIs are spelling applications for people with severe disabilities [9]. For these implementations, high classification accuracy and speed are desired. An issue with c-VEP BCIs is that, usually, a full cycle of the code-pattern is used to produce a command output. Moreover, it is desirable that the system is able to distinguish between intentional and unintentional target fixations. The length of the code pattern becomes a bottleneck with respect to the overall responsiveness of the system. Here, a more user friendly approach is presented, utilizing dynamic classification time windows based on classification thresholds, which we previously used in our SSVEP applications [10–12].

Regarding the signal classification, ensemble-based methods, which are usually used in machine learning, have recently boosted performance in steady-state visual evoked potential (SSVEP)-based BCI systems [13]. Here, such an approach is adopted for the c-VEP paradigm and compared to the conventional approach.

The character output speed can further be enhanced by implementing word prediction methods [14]. Here, an n -gram word prediction model was utilized [15,16], which offers suggestions on the word level. The system was tested on-line using an eight-target spelling interface.

In summary, the contributions of this research are threefold:

- Implementation of a novel software-based synchronization between stimulus presentation and EEG data acquisition,
- investigation of performance improvements in c-VEP detection utilizing an ensemble-based classification approach,
- presenting dynamic on-line classification utilizing sliding classification windows and n -gram word prediction.

The article evaluates the feasibility of the proposed methods based on a test with 18 healthy participants.

2. Materials and Methods

This section describes the methods and materials as well as the experimental design. The sliding window mechanism, as well as the utilized dictionary-driven have been presented before in our previous publication [15].

2.1. Participants

Eighteen able-bodied participants (eight female and ten male) with mean (SD) age of 23.3 (4.4) years, ranging from 19 to 31, were recruited from the Rhine-Waal University of Applied Sciences. Participants had normal or corrected-to-normal vision. They gave written informed consent in accordance with the Declaration of Helsinki before taking part in the experiment. This research was approved by the ethical committee of the medical faculty of the University Duisburg–Essen. Information needed for the analysis of the test was stored pseudonymously. Participants had the opportunity to withdraw at any time.

2.2. Hardware

The used computer (MSI GT 73VR with nVidia GTX1070 graphic card) operated on Microsoft Windows 10 Education running on an Intel processor (Intel Core i7, 2.70 GHz). A liquid crystal display screen (Asus ROG Swift PG258Q, 1920 × 1080 pixel, 240 Hz refresh rate) was used to display the user interface and present the stimuli.

All 16 channels of the utilized EEG amplifier (g.USBamp, Guger Technologies, Graz, Austria) were used; the electrodes were placed according to the international 10/5 system of electrode placement (see, e.g., [17] for more details): P_Z , P_3 , P_4 , P_5 , P_6 , PO_3 , PO_4 , PO_7 , PO_8 , POO_1 , POO_2 , O_1 , O_2 , O_Z , O_9 , and O_{10} . In general, good results may be achieved with a smaller number of the EEG channels, however, a higher number of EEG electrodes is beneficial to achieve higher accuracies and ITRs. In this study, the number of electrodes used was defined by the hardware (limited to 16). Further, the common reference electrode was placed at C_Z and the ground electrode at AF_Z (quite common locations of the ground and reference electrodes for BCI studies based on visual stimuli). Standard abrasive electrolytic electrode gel was applied between the electrodes and the scalp to bring impedances below 5 k Ω . The sampling frequency of the amplifier, F_s , was set to 600 Hz.

2.3. Stimulus Design

In the c-VEP system used in this study, eight boxes (230 × 230 pixel), each corresponding to one of $K = 8$ stimulus classes, arranged as 2 × 4 stimulus matrix (see Figure 3) were presented. The color of a target stimuli alternated between the color of the background, ‘black’ (represented by ‘0’) and ‘white’ (represented by ‘1’) in accord with a distinct flickering pattern. To this end, the well-established 63 bit m-sequences, non-periodic binary code patterns, which can be generated using linear feedback were applied.

The m-sequences c_i , $i = 1, \dots, K$ were assigned to the stimulus matrix employing a circular shift of 2 bits (c_1 had no shift, c_2 was shifted by 2 bits to the left, c_3 was shifted by 4 bits to the left, etc.). The initial code, c_1 is presented in Figure 1.

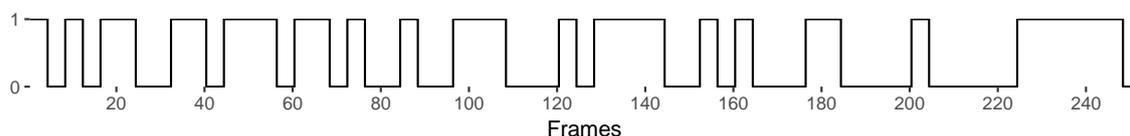


Figure 1. Stimulus pattern of the 63 bits m-sequence used in the experiment. Each ‘1’ in the m-sequence corresponded to four frames where the associated stimulus was shown and each ‘0’ to four frames where the stimulus was not shown. Thus the duration of a stimulus cycle was 1.05 s (also achievable with common 60 Hz monitors).

2.4. Synchronization

The synchronization between stimulus presentation and data acquisition is necessary, as the values for the sampling frequency of the amplifier F_s as well as for the monitor refresh rate r are not precise and small differences might accumulate.

Two timers were used to determine the stimulus onset delay d_s , which describes the time interval between the beginning of a signal acquisition block and stimulus onset. A time stamp t_1 was acquired directly after the command responsible for the initiation of the flickering in the thread dedicated to the stimulus presentation. A second time stamp, t_2 was acquired after receiving a block of EEG data in the thread dedicated for signal classification. The number of samples n_b of one amplifier block is set prior to the experiment. The duration of the collection of one EEG data block in seconds is $d_b = n_b / F_s$.

The time interval between drawing command and stimulus presentation can thus be calculated as $d_s = t_2 - d_b + t_1$.

The number of samples prior to stimulus onset, n_s , was determined as $n_s = \lceil d_s F_s \rceil$, where $\lceil \cdot \rceil$ denotes the nearest integer function; half integers were rounded to the nearest even integer.

Therefore, the difference between the calculated stimulus onset and the duration of the removed samples cannot surpass $1/2F_s$. This accuracy can not as easily be achieved with hardware based triggers; when using the digital input of the amplifier, the sample corresponding to stimulus onset is either rounded down or up, the difference can therefore be higher than $1/2F_s$. An illustration of the proposed software-based synchronization is provided in Figure 2.

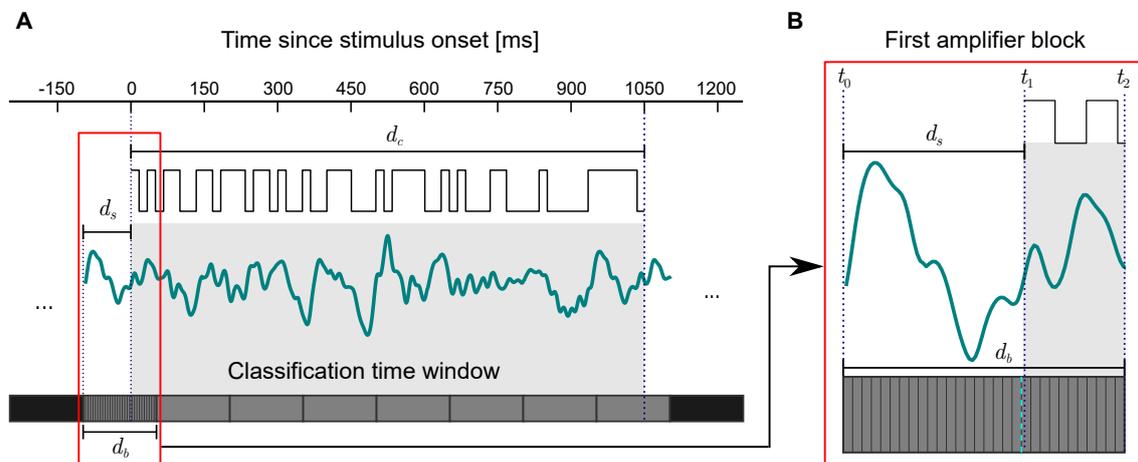


Figure 2. Software-based synchronization between signal acquisition and stimulus presentation. (A) Real-time data analysis interprets the acquired EEG-data with respect to a classification time window; displayed is the averaged EEG response. The classification is performed block-wise (i.e., after acquisition of a new amplifier block, every d_b ms). The EEG-data collected prior to stimulus onset need to be shuffled out. Collection of a minimum time window, e.g., the length of one stimulus cycle, d_c , can be used as additional condition to trigger an output command. (B) The first amplifier block is shown (at the bottom). Stimulus onset duration, d_s , was calculated after receiving the first block after the gaze-shifting period. It was determined using the block duration, d_b , as well as time stamps t_2 and t_1 which were set in the thread dedicated to the stimulus presentation, and classification respectively. The dashed blue line indicates the last sample that is shuffled out.

2.5. Experimental Procedure

First, each of the 18 participants went through a training phase, which was required to generate individual templates and spatial filters for on-line classification. Thereafter, an on-line copy spelling task was performed, which immediately followed the training phase.

In the training, data for each of the stimuli were collected. The data collection was grouped in six blocks, $n_b = 6$; in each block each of the $K = 8$ targets was fixated. Hence, $n_b \cdot K = 48$ trials were collected in total.

Each of these trials lasted for 3.15 s, i.e., the code patterns c_i (see Figure 1) repeated for 3 cycles. The box at which the user needed to fixate was highlighted by a green frame. At the beginning of each of the n_b recording blocks, the flickering was initiated by the user by pressing the space bar. After each trial, the next box the user needed to focus on was highlighted, and the flickering paused for one second. After every eight trials (one block) the user was allowed to rest.

The training phase was followed by a familiarization run where participants spelled the word BCI. The classification threshold was adjusted manually during this familiarization run to ensure adequate speed.

Three spelling tasks were performed: First, the word BRAIN was spelled (word task), thereafter the sentence THAT_IS_FUN (to get familiar with the integrated dictionary) and an additional sentence, different for each user (individual sentence task, see Table 1) were spelled with the BCI. Errors needed to be corrected using the integrated UNDO function. For the sentence spelling tasks, dictionary suggestions could be selected.

Table 1. Individual sentences for the on-line experiment.

Subject	Sentence
1	I_FORGOT_TO_DO_MY_HOMework
2	I_LIKE_TO_EAT_CHEESE
3	I_BOUGHT_EGGS_TODAY
4	I_COULD_NOT_HEAR_THAT_
5	I_DO_NOT_SPEAK_FINNISH
6	WHAT_DID_YOU_HAVE_IN_MIND
7	I_AM_NOT_YET_HUNGRY
8	HOW_LATE_IS_IT
9	I_COULD_EAT_PIZZA Everyday
10	THE_DIVING_SUIT_IS_TOO_SMALL
11	THE_SUN_IS_SLOWLY_RISING
12	IT_IS_GOING_TO_RAIN_TOMORROW
13	THE_DOG_BARKED_LOUDLY
14	THE_LIGHT_BULB_HAS_BURNED_OUT
15	HE_SANG_OUT_OF_TUNE
16	MY_BIKE_HAS_NOT_BEEN_STOLEN
17	THEY_OWN_A_BLACK_CAT
18	AND_THAT_IS_IT

2.6. Dictionary Supported Spelling Interface

An eight target spelling interface as presented in [15] was utilized. The graphical user interface (GUI) is illustrated in Figure 3. Selecting individual characters required two steps. The first row of GUI contained 28 characters (26 letters, underscore and full stop character) divided into four boxes (seven characters each). The second row offered three dictionary suggestions, as well as a correction option. By selecting the correction option, the last typed character or word was deleted. By selecting a letter group from the first row, the associated characters were presented individually (see Figure 3).

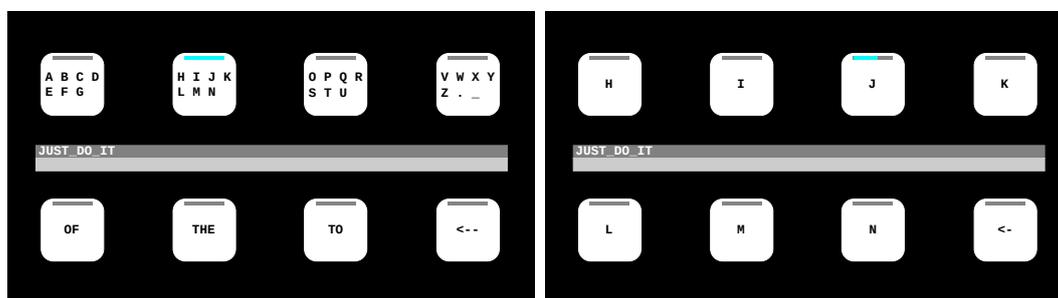


Figure 3. Interface of the eight-target speller used in the on-line experiment. In the first layer of the interface dictionary suggestions based on n -gram word prediction model were provided. By selecting a group of letters (e.g., H–N), a second layer containing individual letters was displayed.

The dictionary suggestions were updated after each performed selection, on the basis of an n -gram prediction model, which is used in computational linguistics.

This model considers a sequence of n items from a text database. An item x_i (here, a word) has the probability $P(x_i|x_{i-(n-1)}, \dots, x_{i-1})$. Here, a bi-gram ($n = 2$) was utilized, to predict word candidates based on the previously typed/selected word.

The text database was derived from the Leipzig Corpora Collection [16]. The corpora collection based on English news was derived from approximately 1 million sentences. It contained a word frequency list and a word bi-grams list (co-occurrences as next neighbors). The word suggestions were retrieved on-line from the database using structured query language (SQL). An example of the functioning of the dictionary-driven speller is provided in Table 2.

Every selection was accompanied by audio and visual feedback (the size of the selected box increased for a short time). Additionally, a progress bar displayed the current certainty level of the associated class label.

Table 2. Writing the sentence JUST_DO_IT with the eight-target speller. Selection of individual letters required two steps, the group containing the character (Layer 1), and the desired character (Layer 2). When selecting one of the three dictionary suggestions, the word as well as a space character where added to the current user sentence.

#	Selection	Layer	Command	Suggestion 1	Suggestion 2	Suggestion 3	User Sentence
1	H-N	1	2	OF	THE	TO	
2	J	2	3	-	-	-	
3	JUST	1	2	JULY	JUNE	JUST	J
4	A-G	1	1	A	AS	ONE	JUST_
5	D	2	4	-	-	-	JUST_D
6	DO	1	6	DAYS	DO	DOING	JUST_DO_
7	IT	1	5	IT	NOT	YOU	JUST_DO_IT_

2.7. Spatial Filtering and Template Generation

In this study, two approaches of spatial filtering, the conventional and the ensemble-based approach were investigated. In both approaches, Canonical-correlation analysis (CCA) [18], a statistical method which investigates the relationship between two sets of variables $\mathbf{X} \in \mathbb{R}^{p \times s}$ and $\mathbf{Y} \in \mathbb{R}^{q \times s}$, was utilized (see, e.g., [6]).

CCA determines weight vectors $\mathbf{w}_X \in \mathbb{R}^p$ and $\mathbf{w}_Y \in \mathbb{R}^q$ that maximize the correlation ρ between the linear combinations $\mathbf{x} = \mathbf{X}^T \mathbf{w}_X$ and $\mathbf{y} = \mathbf{Y}^T \mathbf{w}_Y$ by solving

$$\max_{\mathbf{w}_X, \mathbf{w}_Y} \rho(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{w}_X^T \mathbf{X} \mathbf{Y}^T \mathbf{w}_Y}{\sqrt{\mathbf{w}_X^T \mathbf{X} \mathbf{X}^T \mathbf{w}_X \mathbf{w}_Y^T \mathbf{Y} \mathbf{Y}^T \mathbf{w}_Y}}. \quad (1)$$

Each training trial was stored in a $m \times n_t$ matrix, where m is the number of electrode channels (here all 16 signal channels of the amplifier were utilized for computation, i.e., $m = 16$) and n_t is the number of sample points (here, $n_t = 1.05 \cdot F_s \cdot 3 = 1890$).

In the conventional approach, all training trials are shifted to a zero-class trials $\mathbf{Z}_i, i = 1, \dots, n_b K$ and then averaged yielding an averaged zero-class template $\bar{\mathbf{Z}}$.

The matrices

$$\hat{\mathbf{Z}} = [\mathbf{Z}_1 \mathbf{Z}_2 \dots \mathbf{Z}_{n_b K}] \quad \text{and} \quad \bar{\mathbf{Z}} = \underbrace{[\bar{\mathbf{Z}} \bar{\mathbf{Z}} \dots \bar{\mathbf{Z}}]}_{n_b K} \quad (2)$$

were inserted into (1), yielding a filter vector $\mathbf{w}^{(1)} = \mathbf{w}_{\bar{\mathbf{Z}}}$. Class specific templates $\mathbf{X}_i^{(1)}, i = 1, \dots, K$ were generated by circular shifting the zero-shifted average $\bar{\mathbf{Z}}$ in accordance with the bit-shift of the underlying code c_i .

For the ensemble-based approach, individual templates $\mathbf{X}_i^{(2)} \in \mathbb{R}^{m \times n_t}$ and filters $\mathbf{w}_i^{(2)}$ were determined for each stimulus ($i = 1, \dots, K$). Class specific trial averages $\bar{\mathbf{X}}_i$ were generated by averaging all trials corresponding to the i -th class, $\mathbf{T}_{ij}, j = 1, \dots, n_b$. The matrices,

$$\hat{\mathbf{T}}_i = [\mathbf{T}_{i1} \mathbf{T}_{i2} \dots \mathbf{T}_{in_b}] \quad \text{and} \quad \mathbf{X}_i^{(2)} = \underbrace{[\bar{\mathbf{X}}_i \bar{\mathbf{X}}_i \dots \bar{\mathbf{X}}_i]}_{n_b} \quad (3)$$

were constructed and inserted into (1), yielding $\mathbf{w}_i^{(2)} = \mathbf{w}_{\bar{\mathbf{X}}_i}, i = 1, \dots, K$.

For both methods, the on-line classification was performed after receiving new EEG data blocks, which were automatically added to a data buffer $\mathbf{Y} \in \mathbb{R}^{m \times n_y}$ with dynamically changing column dimension n_y .

The data buffer \mathbf{Y} was compared to reference signals $\mathbf{R}_i^{(j)} \in \mathbb{R}^{m \times n_y}, i = 1, \dots, K$ which were constructed as sub-matrix of the corresponding training template from rows $1, \dots, m$ and columns $n_s, \dots, n_y + n_s$ from $\mathbf{X}_i^{(j)}$ for the conventional ($j = 1$) and ensemble method ($j = 2$), respectively.

3. Results

All participants completed the on-line experiment. The two tested classification approaches were compared using off-line leave-one-out cross-validation. In this respect, all but one recording blocks were used for the training and one block was used as validation data. The cross-validation process was repeated n_b times, with each recording block used once as the validation data. The n_b results were then averaged. Figure 5 shows accuracies across all participants for classification time windows up to 1.05 s. The accuracies for the ensemble-based classification were significantly higher.

The on-line performance between word and sentence spelling tasks was evaluated utilizing the output command accuracy, the ITR, as well as the output characters per minute (OCM) which is a measure of typing speed. The OCM is calculated by dividing the total number of output characters by the time needed to type them [14]. The ITR in bpm [1] was calculated as

$$ITR = \frac{\log_2 K + p \log_2 p + (1 - p) \log_2 \left(\frac{1-p}{K-1} \right)}{t/60}, \quad (7)$$

where p represents the identification accuracy (the number of correctly classified commands divided by the total number of commands), and t represents the average time between consecutive selections, (in s). A calculation tool for the ITR can be found at <https://bci-lab.hochschule-rhein-waal.de/en/itr.html>.

Table 3 displays the results of the on-line spelling tasks. In terms of detection accuracy, all participants were able to complete the task with average accuracies above 80% for the word—as well as for the sentence task. For the spelling task BRAIN, a mean accuracy of 98.8% was reached; for the sentence spelling task, a mean accuracy of 95.9% was reached. Sixteen out of the eighteen participants completed the spelling task BRAIN without any errors, reaching an accuracy of 100%. For the sentence spelling tasks, still eight participants reached 100% classification accuracy.

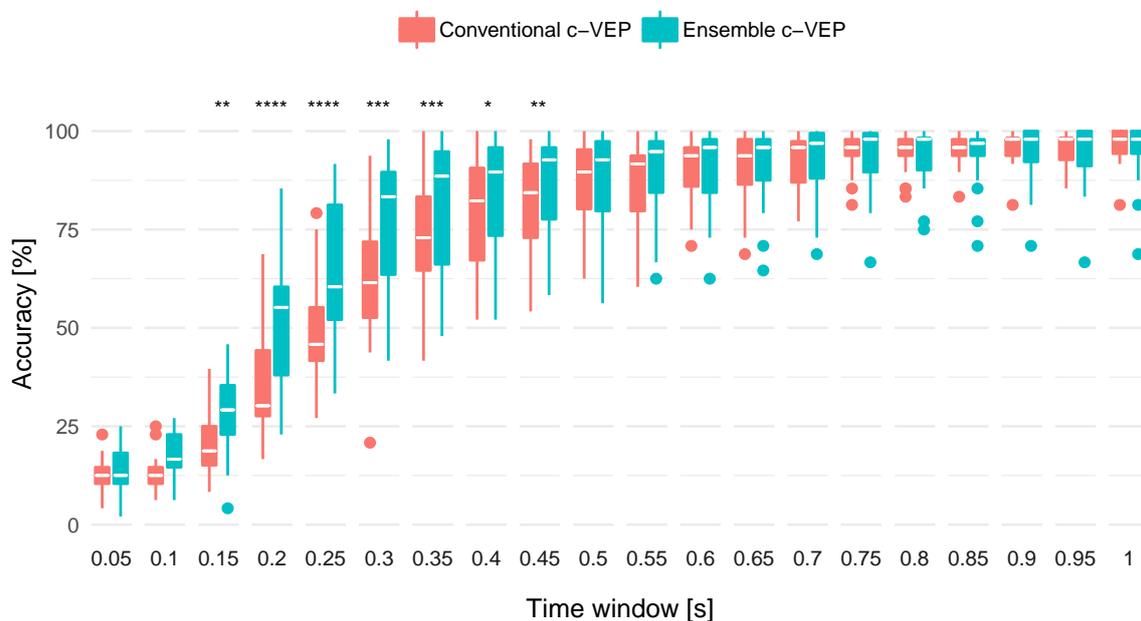


Figure 5. Classification accuracies achieved with the conventional c-VEP classification approach and the ensemble c-VEP classification approach. In the boxplot, outliers (data points outside 1.5 times the interquartile range) are located outside the “whiskers”. The asterisks mark statistical significance (* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$ and **** $p \leq 0.0001$).

Table 3. Provided are the results for the letter by letter spelling task BRAIN and the subject specific individual sentence task as listed in Table 1 (Sent.).

Subject	Accuracy [%]		ITR [bpm]		OCM [chars/min]	
	BRAIN	Sent.	BRAIN	Sent.	BRAIN	Sent.
1	100	97	84.7	60.1	14.1	19.3
2	100	96	65.9	45.5	12.2	15.1
3	100	95	79.3	57.5	13.2	20.1
4	100	100	74.1	58.9	12.3	24.0
5	100	100	54.9	61.6	9.1	19.7
6	100	97	71.6	58.3	11.9	18.8
7	100	100	60.8	57.0	10.1	17.3
8	100	100	79.8	80.0	13.3	19.0
9	100	100	97.0	48.1	16.2	15.4
10	100	100	125.4	95.8	20.9	22.6
11	100	97	79.0	53.7	13.2	15.0
12	86	96	43.2	49.4	7.7	19.1
13	100	100	86.1	76.0	14.4	21.4
14	100	86	85.1	46.8	14.2	18.5
15	92	90	50.2	49.0	8.1	13.8
16	100	91	91.1	62.6	15.2	23.9
17	100	100	57.2	46.1	9.5	11.4
18	100	82	77.1	34.5	12.8	16.6
SD	3.6	5.2	18.7	14.0	3.1	3.3
Mean	98.8	95.9	75.7	57.8	12.7	18.4

The average ITR for the spelling task BRAIN was 75.7 bpm. For the individual sentence spelling task, it was significantly lower, 57.8 bpm (paired two-sample t-test: $t = 4.6608$, $df = 17$, $p < 0.001$). Across individual participants, the minimal and maximal ITR were 43.2 bpm and 125.4 bpm for the spelling task BRAIN and 34.5 bpm and 95.8 bpm for the sentence spelling task, respectively.

However, in terms of OCM, significantly better results were achieved when the dictionary integration was used. The average OCM was 12.7 char/min for spelling BRAIN and 18.4 char/min for the individual sentence task ($t = -6.9089$, $df = 17$, $p < 0.00001$). Across individual participants, the minimal and maximal OCM were 7.7 char/min and 20.9 char/min for the spelling task BRAIN and 11.4 char/min and 20.4 char/min for the sentence spelling task, respectively.

4. Discussion

In this study, we presented a dictionary-driven c-VEP spelling application utilizing n -gram based dictionary suggestions. In this sense, implementation of flexible time windows were realized, which are rarely seen in c-VEP systems, where typically fixed time windows are used. Therefore, the presented BCI was able to accurately discriminate between intentional and unintentional fixations (i.e., if the user did not focus on a particular button, or just briefly attended it, e.g., when searching for the desired character, the threshold criterion was not met and no classification was performed).

Another advantage of the approach is the additional user feedback provided through progress bars. Typically, in c-VEP based BCIs, to our best knowledge, feedback is given on trial base only, i.e., after each trial (e.g., the selected letter is displayed, also called as discrete feedback). Here, continuous feedback was provided throughout the trial. This real-time information about the classification is also valuable to customize system parameters during familiarization. Similar methods have been incorporated into asynchronous SSVEP-based BCI systems and lead to increased user friendliness and system accuracy [19,20].

It should be noted, that due to the classification thresholds, the command selection time varies. Hence ITRs in achieved on-line experiments are typically much lower in comparison to results from an off-line analysis.

The selection options of the GUI changed after each selection; the dictionary suggestions were updated after each selection. Changing elements of the GUI could be handled easily due the dynamic time window approach.

It should be further noted, that for two step spelling interfaces, as the one presented here, letter by letter selection includes two selection time windows and two gaze shifting phases. It remains to be tested, if the dictionary support is as beneficial for multi target systems that require only one step to select a character.

Another addition to the state of the art, is the introduction of a novel trigger free stimulus onset determination approach. The high accuracies achieved in the study demonstrate that it is not necessary to send a trigger signal to the amplifier. The same principle can also be adopted to SSVEP systems that utilize hybrid frequency and phase coding, such as the system used by Nakanishi and colleagues [13].

Furthermore, in addition to the latency of the stimulus presentation, some time elapses between stimulus presentation of the eye and the occurrence of a VEP. Although not applied here, some researchers achieved improvements in BCI performance by excluding samples from the beginning of the data buffer to address the latency of the visual system, e.g., Wittevrongel et al. recommended to exclude the first 150 ms of the trials from the decoding for the c-VEP paradigm [5]. Similarly, Jia and colleagues [21] found SSVEP latencies of different stimulus frequencies to be around 130 ms.

As evident from the off-line classification, see Figure 5, the classifier produced accurate labels before a full stimulation cycle was completed. As expected, the accuracy increased when larger time windows were used. However, it can be seen, that for the ensemble-based approach, a time window as low as 0.35 s yielded accuracies around 90% for the majority of participants. In general, the ensemble-based approach, utilizing individual templates for each target demonstrated superior off-line performance.

This can also be observed in on-line spelling: In our previous study [15], we used the conventional approach for copy spelling tasks utilizing the same interface. Participants completed sentences with a mean ITR of 31.08 bpm. Here, the mean ITR was roughly twice as high (57.8 bpm).

A downside of the approach utilized is the prolonged training duration. Performance typically increases when longer training sessions are conducted. Here, we averaged the data over six trials for the ensemble approach. As eight targets were used, the same data yielded 48 trials with the conventional approach.

As investigated by Nagel and colleagues [4], target latency is dependent on the vertical position on the screen; the conventional approach can therefore benefit from a correction of these latencies. It should also be noted, that some c-VEP-BCIs have additional flickering objects around the selectable targets (principal of equivalent neighbors, see e.g., [6]). This strategy has not been applied here, and could lead to additional differences between outer and inner targets.

Furthermore, it must be noted, that higher ITRs can be achieved utilizing the c-VEP paradigm. Spüler et al. [6] achieved 144 bpm and an average of 21.3 error-free letters per minute in on-line spelling tasks; the authors utilized a 32 target c-VEP system with fixed classification time windows of 1.05 s. However, thanks to the dictionary integration, the average number of error-free characters achieved in the presented study (i.e., 18.4 characters/min) was quite similar, albeit using only eight targets.

The dynamic sliding window mechanism as well as the implementation of software-based stimulus synchronization utilized in this study add to a growing body of literature on c-VEP based BCIs. In a future study, we will adopt the methods described here to a multi-target interface. Typically, 32 targets are used to maximize ITR [6,7]. VEP-based BCIs are often compared with eye tracking interfaces, as both require control eye gaze. The responsiveness of the here presented system was promising; hence c-VEP paradigm could be hybridized e.g., with eye tracking technology as described in our previous publication [12].

Author Contributions: Conceptualization, F.G. and I.V.; Data curation, F.G. and I.V.; Formal analysis, F.G. and I.V.; Funding acquisition, I.V.; Investigation, F.G. and I.V.; Methodology, F.G. and I.V.; Project administration, I.V.; Software, F.G. and I.V.; Supervision, I.V.; Validation, F.G. and I.V.; Visualization, F.G. and I.V.; Writing—original draft, F.G. and I.V.; Writing—review & editing, F.G. and I.V.

Funding: This research was funded by the European Fund for Regional Development (EFRD—or EFRE in German) under grant numbers GE-1-1-047 and IT-1-2-001.

Acknowledgments: We thank all the participants of this research study and our student assistants. We especially thank Abdul Saboor and Piotr Stawicki for their contributions to the presented software as well as Aya Rezeika and Mihaly Benda for their valuable feedback.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wolpaw, J.R.; Birbaumer, N.; McFarland, D.J.; Pfurtscheller, G.; Vaughan, T.M. Brain–Computer Interfaces for Communication and Control. *Clin. Neurophysiol.* **2002**, *113*, 767–791. [[CrossRef](#)]
2. Kübler, A.; Furdea, A.; Halder, S.; Hammer, E.M.; Nijboer, F.; Kotchoubey, B. A Brain–Computer Interface Controlled Auditory Event-Related Potential (P300) Spelling System for Locked-In Patients. *Ann. N. Y. Acad. Sci.* **2009**, *1157*, 90–100. [[CrossRef](#)] [[PubMed](#)]
3. Wei, Q.; Feng, S.; Lu, Z. Stimulus Specificity of Brain–Computer Interfaces Based on Code Modulation Visual Evoked Potentials. *PLoS ONE* **2016**, *11*, e0156416. [[CrossRef](#)] [[PubMed](#)]
4. Nagel, S.; Dreher, W.; Rosenstiel, W.; Spüler, M. The Effect of Monitor Raster Latency on VEPs, ERPs and Brain–Computer Interface Performance. *J. Neurosci. Methods* **2018**, *295*, 45–50. [[CrossRef](#)] [[PubMed](#)]
5. Wittevrongel, B.; Van Wolputte, E.; Van Hulle, M.M. Code-Modulated Visual Evoked Potentials Using Fast Stimulus Presentation and Spatiotemporal Beamformer Decoding. *Sci. Rep.* **2017**, *7*. [[CrossRef](#)] [[PubMed](#)]
6. Spüler, M.; Rosenstiel, W.; Bogdan, M. Online Adaptation of a c-VEP Brain–Computer Interface(BCI) Based on Error-Related Potentials and Unsupervised Learning. *PLoS ONE* **2012**, *7*, e51077. [[CrossRef](#)] [[PubMed](#)]
7. Bin, G.; Gao, X.; Wang, Y.; Li, Y.; Hong, B.; Gao, S. A High-Speed BCI Based on Code Modulation VEP. *J. Neural Eng.* **2011**, *8*, 025015. [[CrossRef](#)] [[PubMed](#)]
8. Riechmann, H.; Finke, A.; Ritter, H. Using a cVEP-Based Brain–Computer Interface to Control a Virtual Agent. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2016**, *24*, 692–699. [[CrossRef](#)] [[PubMed](#)]
9. Rezeika, A.; Benda, M.; Stawicki, P.; Gembler, F.; Saboor, A.; Volosyak, I. Brain–Computer Interface Spellers: A Review. *Brain Sci.* **2018**, *8*. [[CrossRef](#)] [[PubMed](#)]
10. Stawicki, P.; Gembler, F.; Volosyak, I. Driving a Semiautonomous Mobile Robotic Car Controlled by an SSVEP-Based BCI. *Comput. Intell. Neurosci.* **2016**, *2016*, 1–14. [[CrossRef](#)] [[PubMed](#)]
11. Gembler, F.; Stawicki, P.; Volosyak, I. Autonomous Parameter Adjustment for SSVEP-Based BCIs with a Novel BCI Wizard. *Front. Neurosci.* **2015**, *9*. [[CrossRef](#)] [[PubMed](#)]
12. Stawicki, P.; Gembler, F.; Rezeika, A.; Volosyak, I. A Novel Hybrid Mental Spelling Application Based on Eye Tracking and SSVEP-Based BCI. *Brain Sci.* **2017**, *7*, 35. [[CrossRef](#)] [[PubMed](#)]
13. Nakanishi, M.; Wang, Y.; Chen, X.; Wang, Y.T.; Gao, X.; Jung, T.P. Enhancing Detection of SSVEPs for a High-Speed Brain Speller Using Task-Related Component Analysis. *IEEE Trans. Biomed. Eng.* **2018**, *65*, 104–112. [[CrossRef](#)] [[PubMed](#)]
14. Speier, W.; Arnold, C.; Pouratian, N. Integrating Language Models into Classifiers for BCI Communication: A Review. *J. Neural Eng.* **2016**, *13*, 031002. [[CrossRef](#)] [[PubMed](#)]
15. Gembler, F.; Stawicki, P.; Saboor, A.; Benda, M.; Grichnik, R.; Rezeika, A.; Volosyak, I. A Dictionary-driven Mental Typewriter Based on Code-Modulated Visual Evoked Potentials (cVEP). In Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan, 7–10 October 2018; pp. 619–624. [[CrossRef](#)]
16. Eckart, T.; Quasthoff, U. Statistical Corpus and Language Comparison on Comparable Corpora. In *Building and Using Comparable Corpora*; Sharoff, S., Rapp, R., Zweigenbaum, P., Fung, P., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 151–165. [[CrossRef](#)]
17. Jurcak, V.; Tsuzuki, D.; Dan, I. 10/20, 10/10, and 10/5 Systems Revisited: Their Validity as Relative Head-Surface-Based Positioning Systems. *NeuroImage* **2007**, *34*, 1600–1611. [[CrossRef](#)] [[PubMed](#)]
18. Hotelling, H. Relations between two sets of variates. *Biometrika* **1936**, *28*, 321–377. [[CrossRef](#)]

19. Benda, M.; Stawicki, P.; Gemblar, F.; Grichnik, R.; Rezeika, A.; Saboor, A.; Volosyak, I. Different Feedback Methods For An SSVEP-Based BCI. In Proceedings of the 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Honolulu, HI, USA, 18–21 July 2018; pp. 1939–1943. [[CrossRef](#)]
20. Volosyak, I. SSVEP-Based Bremen–BCI Interface—Boosting Information Transfer Rates. *J. Neural Eng.* **2011**, *8*, 036020. [[CrossRef](#)] [[PubMed](#)]
21. Jia, C.; Gao, X.; Hong, B.; Gao, S. Frequency and phase mixed coding in SSVEP-based brain–computer interface. *IEEE Trans. Biomed. Eng.* **2011**, *58*, 200–206. [[CrossRef](#)] [[PubMed](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).