Proceedings of the
10th Graz Brain-Computer Interface Conference 2026
10.3217/978-3-99161-093-9-007

Efficient Decoding of the Code-Modulated Motion Visual Evoked Potential (c-MVEP) for BCIs

H.A. Scheppink[1,2], S. Bialonski[2,3,4], M. Tangermann[5], J. Thielen[5], I. Volosyak[1,2]

[1]Faculty of Technology and Bionics, Rhine-Waal University of Applied Sciences, Kleve, Germany
[2]Department of Informatics and Data Science, Graduate School for Applied Research in North Rhine-Westphalia, Bochum, Germany
[3]Department of Medical Engineering and Technomathematics
[4]Institute for Data-Driven Technologies
FH Aachen University of Applied Sciences, Jülich, Germany
[5]Radboud University, Donders Institute for Brain, Cognition and Behaviour, Nijmegen, The Netherlands

E-mail: ivan.volosyak@hochschule-rhein-waal.de

ABSTRACT: The code-modulated motion visual evoked potential (c-MVEP) paradigm visually stimulates targets using motion in pseudo-random intervals. Due to its code-modulated nature, c-MVEP shares aspects of the code-modulated visual evoked potential (c-VEP). This study investigated whether the c-MVEP can be decoded using similar classification approaches as used for c-VEP. Several canonical correlation analysis (CCA)-based methods were evaluated, with increasingly constrained assumptions about the neural data, to reduce model complexity. These included ensemble and global spatial filters, as well as template estimation by averaging across trials, cycles, classes, or stimulus events. Consistent with findings in c-VEP, we found that the event-based approaches, with events defined by the rising or falling edge of the stimulus sequence, achieved a higher decoding accuracy with fewer training data for calibration than the commonly used trial-based approach. This study shows the potential for decoding the c-MVEP with high accuracy, without requiring extensive training.

INTRODUCTION

A brain-computer interface (BCI) uses brain signals to enable users to control a BCI application without requiring muscular activity [1]. The fastest and highest performance non-invasive BCIs for communication use steady-state visual evoked potential (SSVEP) or code-modulated visual evoked potential (c-VEP), usually based on high-contrast flickering, which is visually and mentally fatiguing, and causes eye-strain during longer use [2, 3].
To address this, the steady-state motion visual evoked potential (SSMVEP) was proposed [2, 4], using frequency-based movement of the stimuli, such as rotation or radial zooming. Chai et al. showed users may prefer SSMVEP over the SSVEP flickering [5].
However, the decoding accuracy and information transfer rate (ITR) of SSMVEP are not on par with SSVEP and c-VEP [6]. Furthermore, while c-VEP is a robust and reliable paradigm for all users [7], SSMVEP was less reliable, as not all users can reach sufficient performance [6]. These findings motivated us to integrate the comfort of SSMVEP with the robustness of c-VEP, leading us to propose the code-modulated motion visual evoked potential (c-MVEP) [8]. Specifically, the novel c-MVEP paradigm uses pseudo-random stimulation sequences with optimal autocorrelation properties from c-VEP, as well as the comfortable smooth motion characteristics of SSMVEP.
In this study, we investigate whether the c-MVEP can be decoded similarly to the c-VEP. We evaluate (1) an ensemble canonical correlation (CCA) approach with class-specific spatial filters [9] and (2) a CCA restricted to one global spatial filter. For both, (a) templates are estimated by trial averaging. For (2), we additionally compared averaging across (b) stimulus sequences and (c) classes, exploiting the property of shifted m-sequences [3].
To further reduce calibration requirements, we use the (3) linear superposition model and reconvolution-CCA approach proposed by Thielen et al. [10], which represents sequence responses as the sum of event-related responses. This allows averaging over stimulus events, such as single flashes. Since each sequence consists of many events, reducing model complexity by event-level averaging decreases the number of parameters, while increasing the effective number of training samples per recording. As a result, the same recording delivers more training samples, which allows for a shorter calibration.
This study investigates the applicability of the assumptions made in c-VEP for c-MVEP. Specifically, we work towards an efficient decoding approach, with only a few free parameters and a small amount of training data to achieve high decoding accuracies.

MATERIALS AND METHODS

Data: We used a subset of previously recorded data

10.3217/978-3-99161-093-9-007

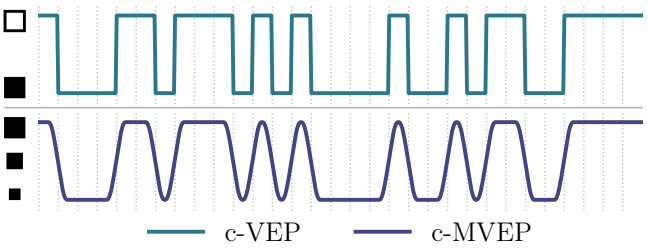

Figure 1: **Stimulation waveforms.** The stimulation waveforms for c-VEP (top) and c-MVEP (bottom). Both use the same 31-bit m-sequence, with c-MVEP using a smoothing of the edges for continuous stimulus size modulation (illustrated by three zoom levels), whereas for c-VEP it translates to binary black-white flicker (illustrated by two flicker states). The dotted vertical lines represent each bit.

from 19 participants (14 males, 5 females, 23.7 ± 3.9 years, range 19-35) with normal or corrected-to-normal vision, and four had previous experience with a BCI system [8]. The study was designed and conducted in accordance with the Helsinki Declaration and was approved by the ethical committee of the medical faculty of Duisburg-Essen University (24-11957-BO).

The EEG data was acquired at a sampling frequency F_s of 600 Hz with a g.USBamp amplifier (g.tec, Schiedlberg, Austria) using sixteen gel-based passive scalp electrodes placed according to the international 10-5 system (P_7, P_3, P_z, P_4, P_8, PO_7, PO_3, PO_z, PO_4, PO_8, O_1, O_z, O_2, O_9, I_z, and O_{10}), with reference at C_z and ground at AF_z. Impedances were kept below 5 kΩ. A 50 Hz notch filter and 1–40 Hz fourth-order Butterworth band-pass filter were applied to the raw data. No outlier removal was done, all epochs and channels were kept for analysis.

The experiment was conducted on a Dell Precision desktop (Windows 11 Education, Intel Core i9 3.70 GHz, NVIDIA RTX 3070). Visual stimuli were presented on an Acer Nitro XV252Q F LCD screen (1920 × 1080 pixels) with a refresh rate of 360 Hz.

The original experiment included four conditions: SSVEP, SSMVEP, c-VEP, and c-MVEP. Our analyses use the c-VEP and c-MVEP data only. Furthermore, we use the original calibration data for an offline analysis.

During this calibration session, four black square stimuli were shown on a gray background, each with a visual angle of $4.34° \times 4.34°$, with $7.26°$ distance between the stimuli, at a viewing distance of 60 cm. The participants fixated on the cued target until the next cue appeared. The cue was presented for 1.5 s, followed by a 1.0 s return to the original stimulus color prior to the onset of flickering or motion. The stimulation in a trial lasted for a total of three stimulus cycles, $3 \times 1.55 = 4.65$ s. Each condition included six four-trial blocks, with each target cued once per block (left to right), leading to 24 trials per condition.

Stimulus Generation: For c-VEP, we used $c_1 = 100011011101010000010011011001111$ as the 31-bit m-sequence for the first target. The second, third, and fourth target c_2, c_3, and c_4 used a 2-bit, 4-bit, and 7-bit right-circular shifted version of c_1, respectively. These m-sequences were displayed at a 20 Hz presentation rate to flash the stimuli between black ('0') and white ('1'), creating a stimulus cycle that is $31/20 = 1.55$ s long.

For c-MVEP, the same m-sequences were used. However, a binary sequence for motion would cause abrupt size changes without apparent motion. Thus, the sequences were made continuous by smoothing the transitions between zeros and ones, while preserving the original timing and duration properties of the m-sequence, similar to the approach used in a code-modulated auditory evoked potential (c-AEP) study [11]. This resulted in pseudo-random sequences with unchanged event onset and duration but smoother transitions, thereby improving the perception of motion.

Smoothing of the edges was implemented using a raised sine function on the sequence upsampled to the refresh rate of the screen, where each bit corresponded to 18 frames. For each rising edge ($0 \to 1$) and falling edge ($1 \to 0$), a transition window equal to one bit duration (50 ms) was placed around each transition. Inside a window, the sharp edge was replaced by a smooth sinusoidal function, while outside the sequence remained unchanged (see Fig. 1). The autocorrelation of the smoothed sequences closely resembled that of the original m-sequence, with a value of 1 at zero lag, and approximately -1/N at all other lags, except at lag 1, where it was 0.12. These smoothed m-sequences were used to modulate a radial zooming motion, mapping '1' to the full stimulus size and '0' to half its size, again with a stimulus cycle of 1.55 s.

Classification: Several template-matching classifiers have previously been proposed to predict the attended target from the recorded EEG data, specifically versions of the average-based reference CCA (denoted as aCCA) [3] and reconvolution CCA (denoted as rCCA) [12]. Both classify an unseen single-trial $\mathbf{X} \in \mathbb{R}^{C \times T}$ of C channels and T samples by selecting the template response $\mathbf{t}_i \in \mathbb{R}^T$ that maximizes the Pearson's correlation with the spatially filtered data using spatial filter $\mathbf{w} \in \mathbb{R}^C$:

$$\hat{y} = \arg\max_i \rho\left(\mathbf{w}^\top \mathbf{X}, \mathbf{t}_i\right). \tag{1}$$

The methods differ in how they estimate the spatial filter $\mathbf{w}$ and template responses $\mathbf{t}_i$ for each of the N classes:

aCCA: CCA learns two spatial filters $\mathbf{w}$ and $\mathbf{v}$:

$$\arg\max_{\mathbf{w},\mathbf{v}} \rho\left(\mathbf{w}^\top \mathbf{S}, \mathbf{v}^\top \mathbf{T}\right), \tag{2}$$

with the concatenated multichannel calibration data $\mathbf{S}$, and the corresponding stacked templates $\mathbf{T}$.

We evaluated four established modes of aCCA: First, we tested an ensemble approach (**aCCA-Ens**), in which individual spatial filters are determined for each stimulus class, that is, the CCA is trained for each class i, leading to spatial filters $\mathbf{w}_i$ and $\mathbf{v}_i$.

Let $\mathbf{X}_i^{(k)} \in \mathbb{R}^{C \times T}$ be the k-th trial of class i, with $i = 1, \ldots, N$ and $k = 1, \ldots, K_i$ trials of class i. Then, for each class i, we define the concatenated multichannel calibration data $\mathbf{S}_i = [\mathbf{X}_i^{(1)}, \ldots, \mathbf{X}_i^{(K_i)}]$ and the corresponding stacked templates $\mathbf{T}_i = [\overline{\mathbf{X}}_i, \ldots, \overline{\mathbf{X}}_i]$, where the

10.3217/978-3-99161-093-9-007

class-average across trials is $\overline{\mathbf{X}}_i = \frac{1}{K_i}\sum_{k=1}^{K_i}\mathbf{X}_i^{(k)}$. Then, aCCA-Ens solves Eq. 2 for each class separately, leading to $(\mathbf{w}_i, \mathbf{v}_i) = \arg\max_{\mathbf{w},\mathbf{v}} \rho\left(\mathbf{w}^\top\mathbf{S}_i, \mathbf{v}^\top\mathbf{T}_i\right)$, yielding one spatial filter per class $\mathbf{w}_i$. Classification is realized by $\hat{y} = \arg\max_i \rho\left(\mathbf{w}_i^\top\mathbf{X}, \mathbf{t}_i\right)$, with template response $\mathbf{t}_i = \mathbf{w}_i^\top\overline{\mathbf{X}}_i$.

Secondly, we test aCCA with one universal spatial filter, assuming that the classes only differ temporally, but share the same spatial source. This leads to the conventional aCCA that learns one spatial filter and estimates templates by averaging full trials (**aCCA-Trl**). Specifically, it solves Eq. 2 once, with $\mathbf{S} = [\mathbf{X}_1^{(1)}, \ldots \mathbf{X}_N^{(K_N)}]$, and $\mathbf{T} = [\overline{\mathbf{X}}_1, \ldots, \overline{\mathbf{X}}_N]$. The obtained universal spatial filter $\mathbf{w}$ is used for classification as in Eq. 1, with $\mathbf{t}_i = \mathbf{w}^\top\overline{\mathbf{X}}_i$.

Thirdly, we test an aCCA that estimates templates by averaging stimulus cycles, rather than trials, where each trial consisted of three cycles. We assume that the response across cycles is the same, such that averaging can be done over the cycles (**aCCA-Cyc**). Let each trial k consist of D cycles, and denote the d-th cycle of trial k of class i as $\mathbf{X}_i^{(k,d)} \in \mathbb{R}^{C\times T_c}$ with T_c samples in a cycle. Instead of averaging over full trials, we average over all cycles of a class $\overline{\mathbf{X}}_i = \frac{1}{K_i D}\sum_{k=1}^{K_i}\sum_{d=1}^{D}\mathbf{X}_i^{(k,d)}$. We solve Eq. 2 once, with $\mathbf{S} = [\mathbf{X}_1^{(1,1)}, \ldots, \mathbf{X}_N^{(K_N,D)}]$ and $\mathbf{T} = [\overline{\mathbf{X}}_1, \ldots, \overline{\mathbf{X}}_N]$. Classification is done as Eq. 1, with $\mathbf{t}_i = \mathbf{w}^\top\overline{\mathbf{X}}_i$.

Fourthly, we test an aCCA that learns a universal template across classes exploiting their lag differences (**aCCA-Lag**). We assume a shared template, because each class has the same underlying m-sequence, but only differs in its temporal shift τ_i. Thus, for calibration, we can align all cycles to zero lag via $\tilde{\mathbf{X}}_i^{(k,d)} = \mathscr{S}_{-\tau_i}(\mathbf{X}_i^{(k,d)})$, where $\mathscr{S}_{\tau_i}$ denotes a circular temporal shift by $-\tau_i$ samples, and then estimate a single canonical template as $\overline{\mathbf{X}}_0 = \frac{1}{\sum_{i=1}^{N} K_i D}\sum_{i=1}^{N}\sum_{k=1}^{K_i}\sum_{d=1}^{D}\tilde{\mathbf{X}}_i^{(k,d)}$. The class-specific templates are generated by circular shifts $\overline{\mathbf{X}}_i = \mathscr{S}_{\tau_i}(\overline{\mathbf{X}}_0)$. We construct the concatenated multichannel calibration data $\mathbf{S} = [\mathbf{X}_1^{(1,1)}, \ldots, \mathbf{X}_N^{(K_N,D)}]$ and the corresponding stacked templates $\mathbf{T} = \left[\overline{\mathbf{X}}_1^{(K_1 D)}, \ldots, \overline{\mathbf{X}}_N^{(K_N D)}\right]$, where $\overline{\mathbf{X}}_i^{(K_i D)}$ denotes $K_i D$ repetitions of $\overline{\mathbf{X}}_i$. This is again used to solve Eq. 2 to obtain $\mathbf{w}$. For classification we use template response $\mathbf{t}_i = \mathbf{w}^\top\overline{\mathbf{X}}_i$ in Eq. 1.

rCCA: The previous approaches averaged at the trial or cycle level. However, we can potentially assume that the response to a single event (e.g., a flash) within such a cycle is the same, no matter where and when it is presented. That is, given the linear superposition hypothesis, we assume that the response to a sequence of events is the linear summation of the responses to individual events. This is formulated in reconvolution CCA (rCCA) [12].

For rCCA, however, different event definitions can be used, such as the duration of the on/off states, the on and/or off states themselves, the rising and/or falling edges, or the stimulus sequence. In turn, rCCA learns the temporal response function to each of the defined events.

Table 1: **Number of parameters and minimum training data.** Listed are the number of parameters and the minimum required training data of each of the classifiers. Here, the number of classes $N = 4$, the number of channels $C = 16$, the sampling frequency $F_s = 600$, the number of event-related time points $L = 0.3 \cdot F_s$, and the number of events $E = 2$ for rCCA-Edge and $E = 1$ for rCCA-ID.

Classifier	Mode	N. Parameters	Min. Data
aCCA-Ens	Trial	$N \cdot C + N \cdot 4.65 \cdot F_s$ (11176)	N trials $4 \cdot 4.65 \cdot F_s$ (11160)
aCCA-Trl	Trial	$C + N \cdot 4.65 \cdot F_s$ (11176)	N trials $4 \cdot 4.65 \cdot F_s$ (11160)
aCCA-Cyc	Cycle	$C + N \cdot 1.55 \cdot F_s$ (3736)	N cycles $4 \cdot 1.55 \cdot F_s$ (3720)
aCCA-Lag	Cycle Lag	$C + 1.55 \cdot F_s$ (946)	1 cycle $1.55 \cdot F_s$ (930)
rCCA-Edge	Event Lag	$C + E \cdot L$ (376)	E events $L \cdot E$ (360)
rCCA-ID	Event Lag	$C + E \cdot L$ (196)	E events $L \cdot E$ (180)

We evaluated two event definitions for rCCA: firstly, we used the rising and falling edges (**rCCA-Edge**) to determine events. This has been shown to generalize across binary sequences [7, 10]. This event definition consists of an event for when the stimulus rises from 0 to 1 (c-VEP) or smoothly grows in size (c-MVEP), and when it falls from 1 to 0 (c-VEP) or shrinks in size (c-MVEP). The rCCA then learns separate responses for the two events. Secondly, because the c-MVEP is using a continuous motion path, we evaluated the direct convolution of the stimulus sequence and one temporal response function, scaled based on the amplitude of the sequence (**rCCA-ID**).

The rCCA can be applied on single cycles, so we again have $\mathbf{X}_i^{(k,d)} \in \mathbb{R}^{C\times T_c}$. A Toeplitz structure matrix $\mathbf{M}_i \in \mathbb{R}^{M\times T_c}$ is constructed, where $M = E \cdot L$, with E denoting the number of events, and L the length of the modeled response per event, here 300 ms so $L = 0.3 * fs$. rCCA-Edge has $E = 2$, corresponding to the rising and falling edge, and rCCA-ID has $E = 1$, corresponding to the stimulation input. The structure matrix $\mathbf{M}_i$ models the onset, duration, and overlap of responses to each event in the sequence of the i-th class. Then, we solve Eq. 3 with concatenated calibration data $\mathbf{S} = [\mathbf{X}_1^{(1,1)}, \ldots, \mathbf{X}_N^{(K_N,D)}]$ and stacked structure matrices $\mathbf{M} = [\mathbf{M}_1, \ldots, \mathbf{M}_N]$, where $\mathbf{M}_i$ is repeated $K_i D$ times.

The general rCCA can be formulated as follows:

$$\arg\max_{\mathbf{w},\mathbf{r}} \rho\left(\mathbf{w}^\top\mathbf{S}, \mathbf{r}^\top\mathbf{M}\right) \tag{3}$$

where both the spatial filter $\mathbf{w} \in \mathbb{R}^C$ for C channels and event-related response(s) $\mathbf{r} \in \mathbb{R}^M$ for M event-related time points are optimized. For classification, template response $\mathbf{t}_i = \mathbf{r}^\top\mathbf{M}_i$ is used in Eq. 1. Code for rCCA is available at https://github.com/thijor/pyntbci.

Analysis: The data were analyzed using a leave-one-block-out cross-validation, using six folds. In each of the folds, learning curves were estimated, where the amount of training data, i.e., train-time, started at a minimum and increased to using all available training data. All the

10.3217/978-3-99161-093-9-007

methods have different requirements regarding the minimum training data needed. Hence, the starting point and step size differ between methods (see Tab. 1).

The training data contained five trials of 4.65 s for each of the N classes, where each trial consisted of three cycles of 1.55 s. For aCCA-Ens and aCCA-Trl, the learning steps ranged from $N \times 4.65$ s (at least a full trial of 4.65 s per class) to $5 \times N \times 4.65$ s in steps of $N \times 4.65$ s, since they require at least one trial per class. The aCCA-Cyc used learning steps ranging from $N \times 1.55$ s (at least one full cycle of 1.55 s per class) to $5 \times 3 \times N \times 1.55$ s in steps of $N \times 1.55$ s, because it requires at least one cycle per class. The aCCA-Lag, rCCA-Edge and rCCA-ID used learning steps from 1.55 s to $5 \times 3 \times N \times 1.55$ s in steps of 1.55 s, because they can generalize across classes. Note that rCCA-Edge and rCCA-ID could use even less training data, as long as some data is present for each event.

To ensure identical training data across classifiers, cycles were added in a trial-like manner. Specifically, the cycle of another trial was added before adding the next cycle within a trial. This was done until we had a trial of all classes, that is $N \times 1.55$, after which we continued by adding the second cycle of each of the first N trials. This ensured that, for any given amount of calibration data, the classifiers were evaluated using identical data. For each fold, the testing data contained N full-trials, one for each class, used to estimate the performance of the classifier (see bci-lab.hochschule-rhein-waal.de/en/acc.html).

Statistical significance was tested using pairwise Wilcoxon tests, with Bonferroni correction.

RESULTS

We systematically compared the classification accuracy of the six classifiers across different amounts of training data while testing on a full trial (see Fig. 2 and Tab. 2). Note that the methods require different minimum training data, resulting in different learning curves onsets: 18.6 s (aCCA-Ens, aCCA-Trl), 6.2 s (aCCA-Cyc), 1.55 s (aCCA-Lag, rCCA-Edge, rCCA-ID), although the rCCA methods could start even earlier, see Tab. 1. For the selected time-points, p-values were Bonferroni-adjusted for the number of pairwise comparisons (3 at 1.55 s; 6 at 6.2 s; 15 at 18.6 and 93.0 s).

c-MVEP: We first evaluate at 1.55 s, which is the minimum data for aCCA-Lag and both reconvolution-based methods. The traditional methods (aCCA-Trl, and aCCA-Ens) and the cycle-based aCCA are not included here due to insufficient data. rCCA-Edge (46.7 %) and rCCA-ID (43.6 %) did not differ significantly ($p > .879$). However, both reconvolution-based methods outperformed aCCA-Lag (30.5 %; $p < .04$).

Secondly, at 6.2 s, also sufficient for aCCA-Cyc with one cycle per class, reconvolution based on the rising and falling edges (76.5 %) outperformed the direct convolution with the stimulus sequence (62.1 %; $p = .037$). Both reconvolution-based approaches and aCCA-Lag (74.3 %) outperformed aCCA-Cyc (31.8 %; $p < .002$). Any other comparison was not significant ($p > .05$).

Table 2: **Offline decoding versus learning time.** Mean decoding accuracy (in %) and standard error for each classifier at four learning times, where minimum data requirements were met. Grey rows indicate c-VEP, white rows c-MVEP.

Classifier	1.55 s	6.2 s	18.6 s	93.0 s
aCCA-Ens	-	-	41.9 (3.1)	98.9 (0.8)
	-	-	36.8 (2.1)	81.1 (5.2)
aCCA-Trl	-	-	41.7 (3.4)	99.8 (0.2)
	-	-	41.0 (2.8)	89.5 (3.6)
aCCA-Cyc	-	42.3 (4.9)	99.8 (0.2)	100.0 (0.0)
	-	31.8 (2.5)	83.1 (4.7)	96.9 (1.3)
aCCA-Lag	36.6 (4.1)	92.3 (3.6)	99.8 (0.2)	100.0 (0.0)
	30.5 (3.2)	74.3 (5.8)	93.9 (2.4)	98.2 (0.7)
rCCA-Edge	72.4 (6.4)	99.1 (0.6)	100.0 (0.0)	100.0 (0.0)
	46.7 (5.3)	76.5 (5.0)	89.9 (3.2)	96.9 (1.2)
rCCA-ID	56.6 (6.9)	93.6 (2.4)	99.6 (0.3)	99.6 (0.4)
	43.6 (4.9)	62.1 (6.2)	77.4 (4.4)	90.1 (3.0)

Thirdly, we compared at 18.6 s, where the calibration data contains a full trial for each class, hence all classifiers can be considered. The aCCA-Lag outperformed all other classifiers ($p < .05$). Reconvolution using the edge event (89.9 %) outperformed the direct convolution (77.4 %; $p = 0.005$) and both traditional methods (36.8 % and 41.0 %; $p < .005$), but had no significant difference with the cycle-based aCCA (83.1 %; $p = 0.098$). The cycle-based aCCA and the direct convolution outperformed both traditional methods ($p < .002$). The cycle-based aCCA and the direct convolution did not differ significantly, nor did the traditional methods ($p > .05$).

Fourthly, we compared the methods at 93.0 s, the maximum training duration. Using universal templates across classes (98.2 %) results in significantly higher decoding accuracies than direct convolution with the stimulus sequence (90.1 %; $p = .037$). Additionally, using the ensemble approach (81.1 %) yields significantly lower performances than the trial (89.5 %), cycle (96.9 %), and lag-based (98.2 %) approaches and reconvolution with the edge event (96.9 %; $p < .03$). None of the other comparisons were significant ($p > .05$).

c-VEP: Overall, the methods show a similar pattern for c-VEP as for c-MVEP, but with higher accuracy for c-VEP (Tab. 2 and Fig. 2). We compared the c-VEP methods using the same learning time points.

First, at 1.55 s, rCCA-Edge (72.4 %) significantly outperformed rCCA-ID ($p = .031$). Both reconvolution-based approaches achieved significantly higher decoding accuracies than the lag-based aCCA (36.6 %; $p < .02$).

Secondly, with one cycle per class, estimating templates by averaging over cycles (42.3 %) resulted in significantly lower accuracies than estimating a universal template across classes (92.3 %) or using the reconvolution approaches (99.1 % and 93.6 %; $p < .001$). Any other comparison did not show a significant difference ($p > .05$).

Thirdly, with one trial per class, both traditional approaches (41.9 % and 41.7 %) reached significantly lower accuracies than all other methods (mean accuracies between 99.6 % and 100.0 %; $p < .001$). Any other com-

10.3217/978-3-99161-093-9-007

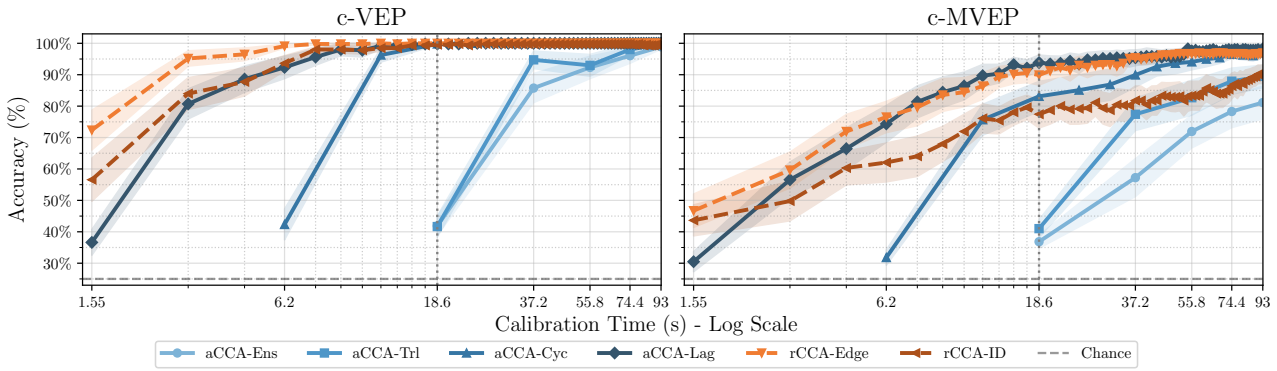

Figure 2: **Offline Learning Curves.** Grand average decoding accuracy and standard error for each classifier across learning time. Left: c-VEP, right: c-MVEP. Each classifier starts from its minimum data requirement to the maximum amount of training data available, in its individual step sizes. Vertical dotted lines represent a step size of a cycle (1.55 s). The vertical solid line at 6.2 s indicates the data included one cycle per class. Solid lines after 18.6 s indicate a step size of a full trial per class.

parison was not statistically significant ($p > .05$).
Fourthly, we evaluated when all training data was used for calibration (93.0 s). Classification accuracies for all classifiers were between 98.9 % and 100.0 %, and showed no significant difference ($p > .05$).

c-VEP versus c-MVEP: Finally, we compared the accuracies of the best-performing classifiers for c-MVEP and c-VEP at two learning time points. At 18.6 s, with one full trial for each class, c-MVEP reached the highest decoding accuracies when learning universal templates across classes, with a mean accuracy of 93.86 %. For c-VEP, rCCA-Edge performed best, reaching 100 %. A pairwise Wilcoxon showed that c-VEP significantly outperformed c-MVEP ($p = .009$). When using all training data for calibration, at 93.0 s, learning universal templates across classes for c-MVEP resulted in a mean decoding accuracy of 98.25 %. For c-VEP, rCCA-Edge again reached 100 %, which was significantly higher than c-MVEP (pairwise Wilcoxon; $p = .01$).

DISCUSSION

We investigated decoding approaches for the proposed code-modulated motion visual evoked potential (c-MVEP), using smoothed m-sequences for motion-based BCI stimulation. Given its code-modulated nature, we hypothesized that c-VEP decoding assumptions transfer to c-MVEP, and indeed observed improved decoding accuracy when applying complexity-reducing assumptions.
First, the traditional decoding approaches aCCA-Ens and aCCA-Trl, both needing to estimate a substantial amount of parameters, achieve moderate performance for c-MVEP, with 81.1 % and 89.5 %, respectively, on the maximum available training data (93.0 s). Additionally, as the learning curves continued to increase, suggesting that even more training data was required, the findings confirm the inefficiency of these traditional methods.
Second, assuming identical responses across cycles, the aCCA-Cyc at 18.6 s and 93.0 s reached 83.1 % and 96.9 % for c-MVEP. With only a trial per class, the averaging over cycles provides effectively more data, while at the same time needing to estimate less parameters than the traditional methods at full training data, suggesting that cycle averaging can substantially improve results.
Third, aCCA-Lag learned a universal template across classes by exploiting m-sequence lag differences. This reduced model complexity and achieved 74.3 % accuracy for c-MVEP with 6.2 s of calibration, increasing to 93.9 % with 18.6 s, suggesting that fewer parameters and effectively more training samples, improve accuracy and learning speed.
Fourth, we applied reconvolution, which assumes identical event responses, and as such further lowers model complexity. rCCA-Edge achieved high decoding accuracies with minimal training data: c-MVEP reached 76.5 % with only 6.2 s, improved to 89.9 % at 18.6 s, and to 96.9 % at maximum data. For c-VEP, one stimulus cycle of 1.55 s was sufficient to reach 72.4 %, 6.2 s to reach 99.1 %, after which it converged to 100 %.
Finally, we modeled responses as the linear convolution of the stimulus and one impulse response function, the model with the smallest complexity. However, this may have led to underfitting: For c-MVEP, rCCA-ID reached 43.6 % with one cycle, 62.1 % with one cycle per class, 77.4 % with one trial per class and, finally, 90.1 % at maximum available data (93.0 s). For c-VEP, this was 56.6 % with one cycle, 93.6 % with one cycle per class, and converged to 99.6 % at one trial per class.
For c-VEP, the results are similar to those observed in prior work [9, 10], with all methods reaching close to 100 % with sufficient training data.
In summary, reconvolution-based approaches appear well-suited for c-MVEP decoding and allow model calibration with minimal training data. Consistent with prior work [7, 10], they show superior performance for c-VEP.
Interestingly, c-MVEP achieved significantly lower accuracies than c-VEP. We believe that at least two factors require further investigation. First, the radial zooming amplitude was limited to 50 % of the stimulus size. Second, c-VEP used full-contrast flicker, while c-MVEP had a lower target-to-background contrast. Both may have caused weaker c-MVEP responses. Future studies should

10.3217/978-3-99161-093-9-007

investigate a fairer comparison in stimulus contrast.

Despite c-MVEP achieving lower performance than c-VEP, we believe it offers a viable alternative when high-contrast flickering is undesirable, impractical, or uncomfortable. It may be particularly suitable for low-light settings, sensitive users, or extended BCI use [4]. Therefore, despite the lower performance, c-MVEP broadens the range of code-modulated protocols.

Interestingly, for both c-MVEP and c-VEP, the rCCA-ID model yielded lower accuracies than rCCA-Edge. These findings suggest that rCCA-ID may be overly constrained, estimating too few parameters and thereby underfitting the response. Specifically, the assumption of full linearity with one temporal response function scaled by stimulus amplitude seems violated. For both contrast and motion reversals, it seems there are sub-additive properties that should be modeled carefully. While the rCCA-Edge that encodes stimulus contrast performed well, future studies should investigate the event model further. This highlights the trade-off between model simplicity and sufficient representational capacity.

For c-MVEP, we showed that it can be accurately decoded after calibration on only one cycle per class. Increasing this to one trial per class, the aCCA-Lag achieved a decoding accuracy of 93.9 %. This is a promising outlook for future c-MVEP BCI applications, particularly when combined with continuous cumulative learning to improve the classifier during online use. Specifically, the user could use the system with minimal calibration, while the decoder progressively refines and optimizes the learned parameters during online operation.

While this study shows clear results for constraining the decoding model, the evaluations were based on a single offline dataset. Replication on other data and an online study should be performed to further validate these findings. Furthermore, future work should address a larger number of classes beyond the used 4 targets in this work.

CONCLUSION

In this study, we introduced a motion stimulus into the c-VEP domain, leading to the code-modulated motion visual evoked potential (c-MVEP). We evaluated different classifiers, each having a smaller parameter space and requiring less training data. With minimal training data, a model learning a universal template across classes and a reconvolution-based model using stimulus events, achieved high decoding accuracies after just 6.2 s of calibration data. This demonstrates that the c-MVEP can be effectively decoded without requiring extensive amounts of training data for calibration.

ACKNOWLEDGMENTS

This work was supported by the European Union's research and innovation program under the Marie Skłodowska-Curie grant agreement No 101118964, the "The Friends of the University Rhine-Waal - Campus Cleve" association, and the Dutch Research Council (NWO) (project number 024.005.022).

AUTHOR CONTRIBUTIONS

HS: Conceptualization, Methodology, Formal Analysis, Writing - original draft - review & editing; JT: Conceptualization, Methodology, Writing - review & editing, Supervision; MT: Methodology, Writing - review & editing, Supervision; SB: Writing - review & editing; IV: Writing - review & editing, Funding acquisition

REFERENCES

[1] Wolpaw J, Wolpaw EW. Brain–Computer Interfaces Principles and Practice. Oxford University Press (2012).
[2] Xie J, Xu G, Wang J, Li M, Han C, Jia Y. Effects of Mental Load and Fatigue on Steady-State Evoked Potential Based Brain Computer Interface Tasks: A Comparison of Periodic Flickering and Motion-Reversal based visual attention. PLOS ONE. 2016;11(9):e0163426.
[3] Martínez-Cagigal V, Thielen J, Santamaría-Vázquez E, Pérez-Velasco S, Desain P, Hornero R. Brain–Computer Interfaces Based on Code-Modulated Visual Evoked Potentials (c-VEP): A Literature Review. Journal of Neural Engineering. 2021;18(6):061002.
[4] Xie J, Xu G, Wang J, Zhang F, Zhang Y. Steady-State Motion Visual Evoked Potentials Produced by Oscillating Newton's Rings: Implications for Brain-Computer Interfaces. PLoS ONE. 2012;7(6):e39707.
[5] Chai X, Zhang Z, Guan K, Liu G, Niu H. A Radial Zoom Motion-Based Paradigm for Steady State Motion Visual Evoked Potentials. Frontiers in Human Neuroscience. 2019;13.
[6] Volosyak I, Rezeika A, Benda M, Gembler F, Stawicki P. Towards solving of the Illiteracy phenomenon for VEP-based brain-computer interfaces. Biomedical Physics & Engineering Express. 2020;6(3):035034.
[7] Thielen J. Addressing BCI inefficiency in c-VEP-based BCIs: A comprehensive study of neurophysiological predictors, binary stimulus sequences, and user comfort. Biomedical Physics & Engineering Express. 2025;11(4):045017.
[8] Scheppink H, Herpers R, Thielen J, Volosyak I. Beyond Flickering: Introducing Code-Modulated Motion Visual Evoked Potentials for Brain-Computer Interfacing. arXiv preprint arXiv:2605.15801.
[9] Gembler F, Volosyak I. A Novel Dictionary-Driven Mental Spelling Application Based on Code-Modulated Visual Evoked Potentials. Computers. 2019;8(2):33.
[10] Thielen J, Marsman P, Farquhar J, Desain P. From full calibration to zero training for a code-modulated visual evoked potentials brain computer interface. Journal of Neural Engineering. 2021.
[11] Scheppink H, Ahmadi S, Desain P, Tangermann M, Thielen J. Towards auditory attention decoding with noise-tagging: A pilot study. In: Proceedings of the 9th Graz Brain-Computer Interface Conference 2024. 2024, 337–342.
[12] Thielen J, Broek P van den, Farquhar J, Desain P. Broad-Band Visually Evoked Potentials: Re(con)volution in Brain-Computer Interfacing. PLOS ONE. 2015;10(7):e0133797.