A PERFORMANCE-BASED APPROACH TO DESIGNING THE STIMULUS PRESENTATION PARADIGM FOR THE P300-BASED BCI BY EXPLOITING CODING THEORY

B. O. Mainsah, L. M. Collins, G. Reeves, C. S. Throckmorton

Electrical and Computer Engineering, Duke University, Durham, NC USA

ABSTRACT

The P300-based brain-computer interface (BCI) speller relies on eliciting and detecting specific brain responses to target stimulus events, termed event-related potentials (ERPs). In a visual speller, ERPs are elicited when the user's desired character, i.e. the "target," is flashed on a computer screen. The P300 speller is currently limited by its relatively slow typing speed due to the need for repetitive data measurements that are necessary to achieve reasonable signal-to-noise ratios. In addition, refractory effects limit the ability to elicit ERPs with every target stimulus event presentation. In this paper, we present a new method to design the stimulus presentation paradigm for the P300 speller by exploiting an information-theoretic approach to maximize the information content that is presented to the user while also mitigating refractory effects. We present results with real-time BCI use which demonstrate significant performance improvements with our performance-based paradigm compared to the conventional stimulus presentation paradigm.

Index Terms— Brain-computer interface, P300 speller, Stimulus paradigm, Coding theory, Combinatorial problem.

1. INTRODUCTION AND RELATION TO PRIOR WORK

The P300-based brain-computer interface (BCI) [1, 2], relies predominantly on event-related potentials (ERPs) as control signals to enable a user to make selections from an array of character choices. These ERPs are elicited as a function of a user's uncertainty regarding stimulus events in either an acoustic, tactile or a visual *oddball* recognition task [3]: the random occurrence of a rare oddball or *target* stimulus within a sequence of more frequently occurring or *non-target* stimuli. Ideally, the target stimulus elicits a distinct ERP response that includes a large positive peak, termed the P300 signal.

In a visual BCI speller, a user selects a character by focusing on that character while groups of characters are randomly illuminated on a screen, such as shown in figure 1. In this scenario, the illumination of the desired character, which is presented in a few of the flash groups, corresponds to a target stimulus event. Given a grid layout, a simple method of grouping characters is by the rows and columns of the grid and presenting them in a random order. This is known as the row-column paradigm (RCP) [1], which is used predominantly in the literature [4].

The P300 speller operates by analyzing electroencephalography (EEG) responses to the stimulus events in order to discern a user's intended character. Due to the low signal-to-noise ratio (SNR) of elicited ERPs embedded in noisy EEG signals, data are collected from multiple presentations of a potential target character to increase the SNR for improved selection accuracy. These repetitive data measurements contribute to the slow spelling speeds of ERP-based BCIs. Typically, a group of characters is presented in a single stimulus

This research was funded by the NIH under grant number R33 DC010470

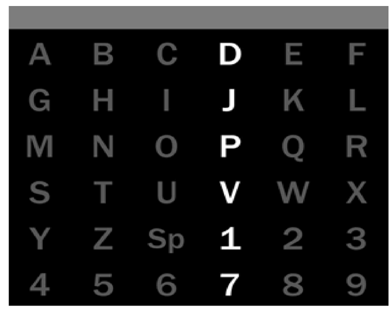

Fig. 1. BCI speller interface with a 6×6 grid layout. In this example, the fourth column is illuminated or "flashed."

event to increase the character presentation rate. However, this increases the likelihood of selection errors due to the added correlation in the cumulative EEG responses of characters that are often grouped together. For example, in the RCP, erroneous character selections are usually in the same row or column as the target character [5]. Another limitation is the negative impact on performance due to *refractory effects* as the relative strength of ERP responses is affected by the timing between target stimulus events, or the target-to-target interval (TTI). Due to the randomized order of presentation of the row and column flash groups in the RCP, there is the possibility of two consecutive target character presentations. In general, classification performance improves with increasing TTI [6, 7].

Re-designing the stimulus presentation paradigm has the potential to minimize selection errors caused by grouping characters for presentation or refractory effects. Some approaches have focused on cosmetic changes to the user interface to either increase focus or elicit other ERPs that can enhance performance, e.g. arranging flash groups into spatially distinct clusters [8], or using distinct elements during stimulus presentation [9, 10]. Other approaches impose a minimum TTI to mitigate refractory effects, e.g. [7, 11]. However, with all these previous approaches, the generation and presentation of flash groups are randomized with limited consideration for maximizing the information content that is presented to the user.

Alternatively, a BCI can be modeled as a noisy communication system, which provides a more principled framework for the design of the stimulus flash groups in terms of information presentation. Coding theory [12] provides a principled approach to packaging information for efficient communication in spite of noisy channel transmission. In the P300 speller, a character is encoded via its presentation pattern, which can be represented by a binary codeword, $\boldsymbol{X}_1^T = [x_1, x_2, ..., x_T]$, where $x_t \in \{0, 1\}$ denotes the absence or presence, respectively, of a character in a flash group. A stimulus presentation paradigm is represented by a codebook, $\mathfrak{C} \in [0, 1]^{M \times T}$, where $\mathfrak{C}(m, :)$ corresponds to the codeword for character C_m. For

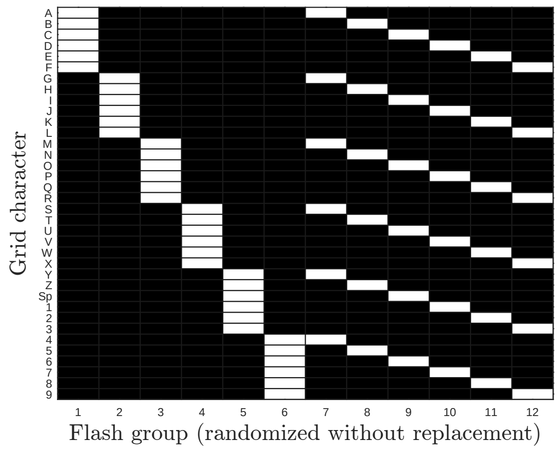

Fig. 2. Codebook for the RCP for the speller grid shown in figure 1. Each column represents a flash group, with presented characters highlighted in white. Each row represents a character's codeword.

the speller grid shown in figure 1, figure 2 shows a sample codebook for the RCP, with a mapping of each character to its codeword.

Some studies have used an information-theoretic approach to design codebooks for the P300 speller, by focusing on minimizing decoding errors by maximizing the dissimilarity or Hamming distance between codewords [13, 14, 15]. However, performances with the proposed codebooks were similar to or worse than that of the RCP in online testing. The codebooks in [13, 14, 15] were characterized by codewords with predominantly short TTIs, a consequence of maximizing Hamming distances with a memoryless channel assumption.

In this work, we focus on using an information-theoretic approach for codebook construction that explicitly considers the physiological limitation of the ERP elicitation. Unlike the previous approaches, we consider a communication channel with long term memory to past target stimulus event presentations. In addition, we use a performance prediction method [16] to objectively compare codebook configurations prior to making a final selection. We denote the stimulus presentation paradigm developed with our new method as the *performance-based paradigm* (PBP). We present results demonstrating the utility of the performance prediction method to select a codebook configuration for the PBP, which statistically significantly improves performance compared to the RCP.

2. THE PERFORMANCE-BASED PARADIGM

Our approach to developing a codebook involves selecting from an exponentially large 2^l search space, a codebook of M l-bit codewords or an (M, l)-code that maximizes performance with a given BCI algorithm. For a dynamic stopping (DS) algorithm, e.g. [17, 18], where the amount of data collection is varied based on a threshold function, this involves a minimization problem of the form:

$$\underset{\mathfrak{C}\in[0,1]^{M\times l}}{\text{minimize}}\ \text{EST}(\alpha_{\mathfrak{C}}),\ \text{subject to } A(\alpha_{\mathfrak{C}}) \geq A_{th}. \tag{1}$$

where EST is the expected stopping time, which is the mean number of stimulus event presentations prior to stopping data collection; A is the selection accuracy; A_{th} is the minimum accuracy desired; and $\alpha_{\mathfrak{C}}$ is a generic parameter which we define to quantify a user's performance level with a given codebook, $\mathfrak{C}$.

2.1. BCI performance prediction

Our performance prediction method is based on a probabilistic model of a generic ERP-based BCI [16]. During a selection process, a user intends to select a target character, C^*. Following each stimulus presentation, the user's EEG response is scored with the system's classifier. The BCI uses a function that quantifies the possibility of each of the BCI choices to be the user's target choice, given data collection. We denote this as the character cumulative response function (CCRF), $\{\Theta_m(t)\}_{m=1}^{M}$, at time index t, for an M-choice BCI. The classifier score, y_t, is used to update $\{\Theta_m(t)\}_{m=1}^{M}$. The BCI terminates data collection at a stopping time, t_s, either when a maximum CCRF value, $\Theta_{\max}(t)$, attains a threshold, Θ_{th}, or a data collection limit, $t_{\max}$, is reached. After data collection, the BCI makes a selection, $\hat{C}^*$, which is usually the character with the maximum CCRF value.

The performance functions can be derived analytically based on the stopping and decision rules. To account for a DS criterion, determining the EST requires averaging over all possible stopping times:

$$\begin{aligned}\text{EST} &= \sum_{t=1}^{t_{\max}} tP(t_s = t),\\ &= \sum_{t=1}^{t_{\max}-1} tP\left(\{\Theta_{\max}(t-1) < \Theta_{th}\} \cap \{\Theta_{\max}(t) \geq \Theta_{th}\}\right)\\ &\quad + t_{\max}P(\Theta_{\max}(t_{\max}-1) \geq \Theta_{th}).\end{aligned} \tag{2}$$

Similar to the EST, accuracy is determined by averaging over all possible stopping times, as well as over all possible character choices:

$$A_i = \sum_{t=1}^{t_{\max}} P\left(\{t_s = t\} \cap \left\{\max_{j\neq i} \Theta_j(t) < \Theta_i(t)\right\} \middle| C_i = C^*\right), \tag{3a}$$

$$A = \sum_{i=1}^{M} A_i P(C_i = C^*) \tag{3b}$$

Alternatively, performance functions can be obtained from simulations of P300 spelling runs, if the solutions to (2)-(3) are intractable.

In this study, we analyze the Bayesian DS algorithm developed for the P300 speller [17]. A probability distribution is maintained over the character choices, $\{P_m(t)\}_{m=1}^{M}$, which represents the level of confidence that each character is the target character at time index t. With each new stimulus presentation, the classifier score is used to update $\{P_m(t)\}_{m=1}^{M}$ by Bayesian inference:

$$P_m(t) = \frac{p_m(t)P_m(t-1)}{\sum_{j=1}^{M} p_j(t)P_j(t-1)},, \tag{4a}$$

$$p_m(t) = \begin{cases} p(y_t|H_0), \text{ if } C_m \notin \mathcal{F}_t \\ p(y_t|H_1), \text{ if } C_m \in \mathcal{F}_t \end{cases}, \tag{4b}$$

where $P_m(t-1)$ and $P_m(t)$ are the prior and posterior probabilities, respectively; $p_m(t)$ is the character likelihood; and $p(y_t|H_0)$ and $p(y_t|H_1)$ are the classifier probability density functions (pdfs) for the non-target and the target EEG responses, respectively. Data collection is stopped when a character's probability attains a threshold probability, P_{th}, within a data collection limit. The character with the maximum probability is selected as the user's intended character.

We derive the performance functions for the Bayesian DS algorithm by parameterizing a user's performance level with the classifier

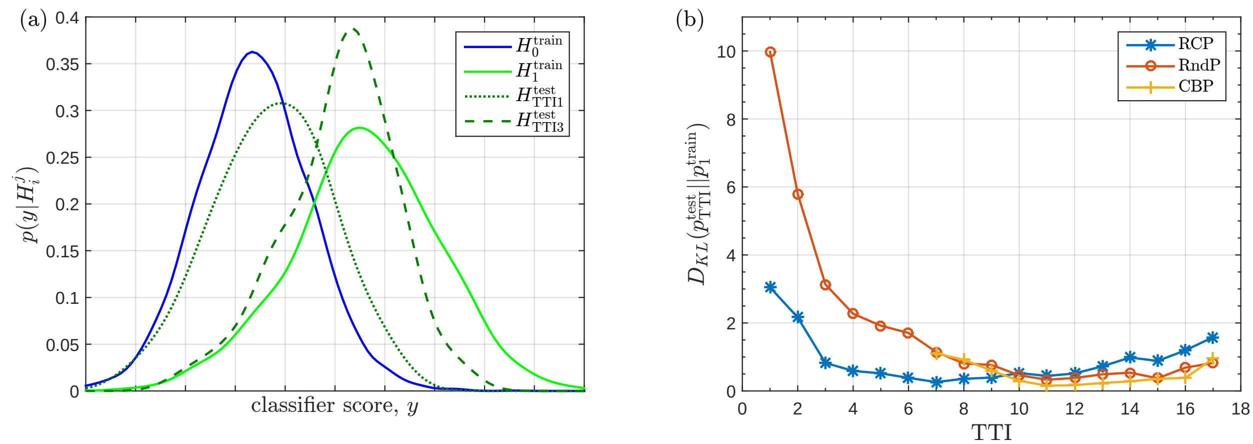

Fig. 3. (a) Illustration of the probability density functions of classifier scores grouped by non-target (H_0) and target (H_1) responses for the training data, and the target responses segregated by TTI for the test data. (b) Kullback-Leibler divergence between TTI-segregated and aggregate target scores for row-column (RCP), random (RndP), and checkerboard (CBP) paradigms, averaged across participants.

detectability index [16]. The detectability index [19], d, is a measure that quantifies the discriminability between two normal pdfs:

$$d = \frac{\mu_1 - \mu_0}{\sqrt{0.5(\sigma_1^2 + \sigma_0^2)}}. \tag{5}$$

In this case, (μ_0, σ_0^2) and (μ_1, σ_1^2), are the parameters of the non-target and target classifier score pdfs, respectively.

2.2. Performance-based parameters

We reduce the 2^l search space based on specific parameters that are tuned to positively affect performance: the minimum Hamming distance, the codeword density and the minimum TTI. Let $d^H(\boldsymbol{c}_i, \boldsymbol{c}_j)$ denote the Hamming distance between codewords $\boldsymbol{c}_i$ and $\boldsymbol{c}_j$. The minimum Hamming distance of a codebook, $d_{\min}^H(\mathfrak{C})$, determines the maximum number of bit classification errors, e_b, that can occur and still be able to correctly estimate the sent message [12]:

$$e_b = \lfloor (d_{\min}^H(\mathfrak{C}) - 1)/2 \rfloor. \tag{6}$$

A minimum TTI is imposed to minimize refractory effects by allowing a "recovery" period between a target character presentations to increase the likelihood of eliciting ERPs with relatively higher SNRs. The codeword density, $w(\boldsymbol{c})$, provides a trade-off between obtaining higher classifier scores for the target character with more presentations and lower classifier scores due to shorter TTIs.

To determine an appropriate minimum TTI that minimizes refractory effects, we analyzed participant EEG data from online implementation of three codebooks: RCP, random (RndP), and checkerboard (CBP) [11] paradigms. In the RndP, the character subsets are randomly generated, with the condition that within a codebook instantiation, a character is only presented again after all of the other characters have been presented. The CBP is a special case of the RndP where a minimum TTI is imposed and spatial restrictions are placed on the composition of flash groups.

For each participant, 10-fold cross validation was performed to train and test the classifier. During each cross-validation block, we obtained the trained classifier likelihoods for non-target and target classifier scores, as well as target classifier scores from the test set, segregated by TTI. Let the patterns, [..11..] and [..1001..], indicate TTIs of 1 and 3, respectively. Figure 3(a) shows example pdfs generated from a participant's data. The pdf for a TTI of 1 is similar to the non-target pdf; hence for a target character presented twice successively, the system is more likely to infer the second presentation as a non-target event. For a TTI of 3, the pdf is similar to the target pdf, as is desirable for more accurate target character selection.

For multiple TTIs, the Kullback-Leibler (KL) divergence between the TTI-segregated and the aggregate target classifier score pdfs were determined. Figure 3(b) shows the KL divergence as a function of TTI, averaged across participants. The TTI-segregated pdfs of shorter TTIs are noticeably dissimilar from the aggregate target pdf, to a larger degree in the RndP than in the RCP. The higher likelihood of generating a low classifier score with shorter TTIs characterizes the potential negative impact on performance due to refractory effects. These effects are minimized in the CBP where a minimum TTI is imposed. Based on these results, a TTI of 3 appears to be a suitable selection to minimize refractory effects and the EST.

2.3. Codebook development

We use a greedy search to iteratively build a codebook by adding a new codeword to a partially-filled codebook such that the objective function is minimized with respect to the other codewords. The degree of refractory effects based on the codebook TTI statistics can significantly affect a user's performance level, $\alpha_{\mathfrak{C}}$ [13, 14, 15]. However, estimating $\alpha_{\mathfrak{C}}$ requires empirical data collection over all codebook configurations, which is infeasible. By considering a codebook space where a user's performance level is maintained due to minimizing refractory effects, i.e. fixed d, we achieve the same objective defined in (1) by maximizing accuracy (3). Algorithm 1 outlines pseudo-code to develop a codebook for the PBP.

In this study, a 6×6 grid was used to design a $(36, 24)$-code. Since performance is compared to the RCP, the number of codebook instantiations for the RCP was doubled so that it is also a $(36, 24)$-code. For a RCP with a $(36, 24)$-code, the performance-based parameters are by definition $d_{\min}^H = 4$, $w(\boldsymbol{c}) = 1/6$, and $\text{TTI}_{\min} = 1$. For a new PBP, a uniform distribution over characters was assumed and an iterative search was performed over parameter values, $d_{\min}^H > 4$, $w(\boldsymbol{c}) \geq 1/4$, and $\text{TTI}_{\min} = 3$, with multiple codebook configurations compared. A final configuration of the PBP

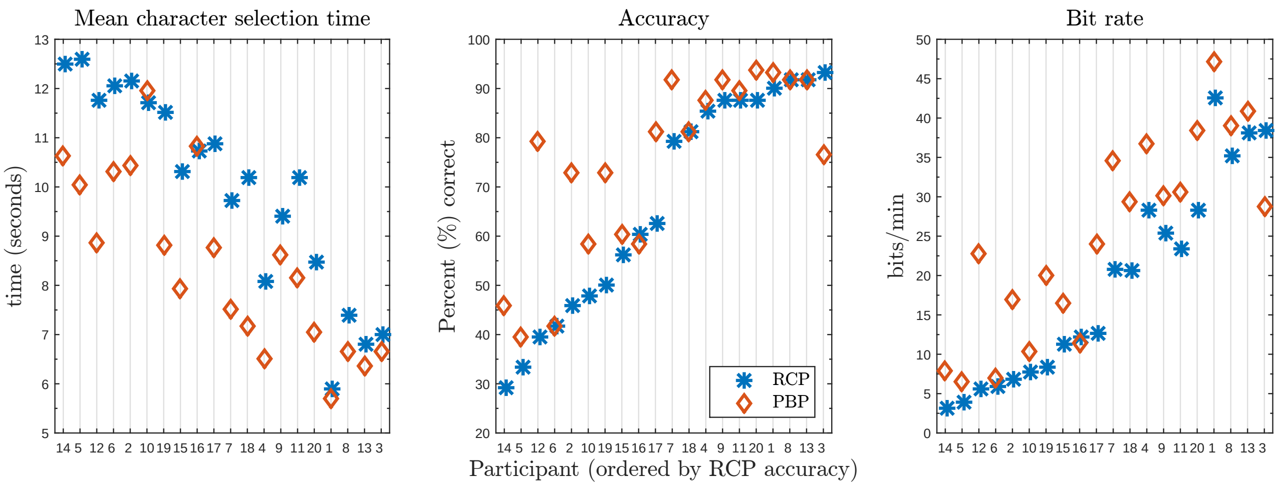

Fig. 4. Online participant performances. The mean character selection time, which includes a 3.5 seconds pause between selections, is the mean amount of time spent per character selection; it is a function of the EST, stimulus duration and inter-stimulus interval. Accuracy is the percentage of correct selections by the BCI. Bit rate is a communication rate measure based on the mean selection time and accuracy.

Algorithm 1: Pseudo-code for codebook development

Function pbpCodebook($M, l, \omega, d^H_{\min}$,$\text{TTI}_{\min}$)
$\mathfrak{C}^{2^l \times l}$ = Space of all l-bit codewords.
$\{\mathfrak{C}\}^r = \mathfrak{C}^{2^l \times l} \overset{\text{remove}}{\rightarrow}$ {codewords with $w(\boldsymbol{c}_i) \notin \omega, \text{TTI}(\boldsymbol{c}_i) <$ $\text{TTI}_{\min}$ and $d^H(\boldsymbol{c}_i, \boldsymbol{c}_j) < d^H_{\min}$}.
$\mathfrak{C}_{\text{new}} = \arg\max_{\boldsymbol{c_i}} \sum_{j=1}^{|\mathfrak{C}^r|-1} d^H(\boldsymbol{c}_i, \boldsymbol{c}_j),\ \boldsymbol{c}_i \neq \boldsymbol{c}_j,\ \{\boldsymbol{c}_i, \boldsymbol{c}_j\} \in \mathfrak{C}^r$
while $|\mathfrak{C}_{new}| < M$ **do**
  for $i = 1 : |\mathfrak{C}_{old}|$ **do**
    $\mathfrak{C}_{\text{temp}} = \mathfrak{C}_{\text{new}} \overset{\text{add}}{\leftarrow} \mathfrak{C}_{\text{old}}(i,:)$
    $A_{\text{temp}}(i)$ = predictedAccuracy($\mathfrak{C}_{\text{temp}}$) ▷ (see [16])
  end
  $\boldsymbol{c}_{\text{new}} = \arg\max_{\boldsymbol{i}} \{A_{\text{temp}}(i)\}_{i=1}^{|\mathfrak{C}_{\text{old}}|}$
  $\mathfrak{C}_{\text{new}} = \mathfrak{C}_{\text{new}} \overset{\text{add}}{\leftarrow} \boldsymbol{c}_{\text{new}}$
  $\{\mathfrak{C}\}^r = \{\mathfrak{C}\}^r \overset{\text{remove}}{\rightarrow} \boldsymbol{c}_{\text{new}}$
end
return $\mathfrak{C}_{\text{new}}$

Notes: $\boldsymbol{X}$ $\{\overset{\text{add}}{\leftarrow} / \overset{\text{remove}}{\rightarrow}\}$ $\boldsymbol{x}$: add/remove codeword $\boldsymbol{x}$ to/from codebook $\boldsymbol{X}$.

was selected and tested online.

3. ONLINE EXPERIMENT AND RESULTS

Twenty healthy participants were recruited at Duke University for a study approved by the university's Institutional Review Board. All participants gave informed consent prior to their experiment session.

The open source BCI2000 software [20] was used to implement the P300 speller. Data collected from electrodes {Fz, Cz, P3, Pz, P4, PO7, PO8, Oz} were used for signal processing. EEG signals were sampled at 256 samples/s and filtered between 0.5-30 Hz prior to signal processing. Feature extraction, P300 classifier weight vector and likelihood estimation were performed as outlined in [17]. In both stimulus presentation paradigms, the stimulus duration, inter-stimulus interval and time pause between character selections were set to 62.5 ms, 62.5 ms and 3.5 s, respectively. The online data collection limit was set at 72 stimulus flashes.

Each participant's BCI experiment session consisted of copy-spelling words using the P300 speller with the RCP or PBP. In a copy-spelling task, the user is instructed by the BCI on which character to focus in each selection process. A BCI session had two blocks, with each block consisting of a calibration run and test run for a codebook. The block order was randomized across participants. The calibration run involved copy-spelling five 6-letter words, without character selection or feedback. Data collected in the calibration run were used to train a user-specific classifier weight vector and cllassifier likelihoods. In the test run, using the trained classifier parameters, participants performed copy-spelling of eight 6-letter words (except participant 1 and 3 with five words) using the Bayesian DS algorithm, with feedback presentation and no error correction.

Figure 4 compares the online participant results for both codebooks, and average results are summarized in table 1. Participants statistically significantly reduced their mean character selection time and improved accuracy with the PBP, resulting in improved communication rates. The online performances follow the predicted trends.

Table 1. Summary of average participant online results

Performance measure	RCP	PBP	p value
Character selection time (s)	9.97 ± 2.06	$\mathbf{8.45 \pm 1.78}$	$< 10^{-4}$
Accuracy (%)	67.08 ± 22.51	$\mathbf{74.96 \pm 18.15}$	$< 10^{-2}$
Bit rate (bits/min)	18.94 ± 12.90	$\mathbf{24.93 \pm 12.50}$	$< 10^{-3}$

4. CONCLUSIONS

We have developed a new performance-based approach to design stimulus presentation patterns or codebooks for the P300 speller, which balance the conflicting interests of information theoretics and physiological limitations. Our results show that this approach significantly improves performance across a wide range of user performance levels. Further validation of the proposed method will be done in a target BCI user population. Future work includes an adaptive codebook design process that incorporates language information to potentially improve performance.

5. REFERENCES

[1] L. A. Farwell and E. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalogr Clin Neurophysiol*, vol. 70, no. 6, pp. 510–523, 1988.

[2] J. R. Wolpaw and E. W. Wolpaw, *Brain-computer interfaces: principles and practice*, Oxford University Press, New York, 2012.

[3] S. Sutton, M. Braren, J. Zubin, and E. R. John, "Evoked-potential correlates of stimulus uncertainty," *Science*, vol. 150, no. 3700, pp. 1187–8, 1965.

[4] J. N. Mak, D. J. McFarland, Theresa M. Vaughan, L. M. McCane, P. Z. Tsui, D. J. Zeitlin, E. W. Sellers, and J. R. Wolpaw, "EEG correlates of P300-based brain-computer interface (BCI) performance in people with amyotrophic lateral sclerosis," *J Neural Eng*, vol. 9, no. 2, pp. 026014, 2012.

[5] R. Fazel-Rezai, "Human error in P300 speller paradigm for brain-computer interface," *Conf Proc IEEE Eng Med Biol Soc*, vol. 2007, pp. 2516–9, 2007.

[6] S. M. M. Martens, N. J. Hill, J. Farquhar, and B. Schölkopf, "Overlap and refractory effects in a brain-computer interface speller based on the visual P300 event-related potential," *J. Neural Eng.*, vol. 6, no. 2, 2009.

[7] J. Jin, E. W. Sellers, and X. Wang, "Targeting an efficient target-to-target interval for P300 speller brain-computer interfaces," *Med Biol Eng Comput*, vol. 50, no. 3, pp. 289–296, 2012.

[8] R. Fazel-Rezai and K. Abhari, "A region-based P300 speller for brain-computer interface," *Can J Elect Comp Eng*, vol. 34, no. 3, pp. 81–85, 2009.

[9] T. Kaufmann, S. M. Schulz, C. Grünzinger, and A. Kübler, "Flashing characters with famous faces improves ERP-based braincomputer interface performance," *J Neural Eng*, vol. 8, no. 5, pp. 056016, 2011.

[10] N. M. Schmidt, B. Blankertz, and M. S. Treder, "Online detection of error-related potentials boosts the performance of mental typewriters," *BMC Neurosci*, vol. 13, pp. 19, 2012.

[11] G. Townsend, B. K. LaPallo, C. B. Boulay, D. J. Krusienski, G. E. Frye, C. K. Hauser, N. E. Schwartz, T. M. Vaughan, J. R. Wolpaw, and E. W. Sellers, "A novel P300-based brain-computer interface stimulus presentation paradigm: moving beyond rows and columns," *Clin Neurophysiol*, vol. 121, no. 7, pp. 1109–20, 2010.

[12] D. J. MacKay, *Information theory, inference and learning algorithms*, Cambridge University Press, 2003.

[13] J. Hill, J. Farquhar, S. Martens, F. Biessmann, and B. Schölkopf, "Effects of stimulus type and of error-correcting code design on BCI speller performance," in *Adv. in Neural Inform. Proc. Systems 21*, pp. 665–672. Curran Associates, Inc., 2009.

[14] S. M. M. Martens, J. M. Mooij, J. N. Hill, J. Farquhar, and B. Schölkopf, "A graphical model framework for decoding in the visual ERP-based BCI speller," *Neural Comput*, vol. 23, no. 1, pp. 160–182, 2011.

[15] J. Geuze, J. D. R. Farquhar, and P. Desain, "Dense codes at high speeds: Varying stimulus properties to improve visual speller performance," *J Neural Eng*, vol. 9, no. 1, pp. 016009, 2012.

[16] B. O. Mainsah, L. M. Collins, and C. S. Throckmorton, "Using the detectability index to predict P300 speller performance," *J Neural Eng*, vol. 13, no. 6, pp. 066007, 2016.

[17] C. S. Throckmorton, K. A. Colwell, D. B. Ryan, E. W. Sellers, and L. M. Collins, "Bayesian approach to dynamically controlling data collection in P300 spellers," *IEEE Trans Neural Syst Rehabil Eng*, vol. 21, no. 3, pp. 508–17, 2013.

[18] B. O. Mainsah, L. M. Collins, K. A. Colwell, E. W. Sellers, D. B. Ryan, K. Caves, and C. S. Throckmorton, "Increasing BCI communication rates with dynamic stopping towards more practical use: an ALS study," *J Neural Eng*, vol. 12, no. 1, pp. 016013, 2015.

[19] A. J. Simpson and M. J. Fitter, "What is the best index of detectability?," *Psychol Bull*, vol. 80, no. 6, pp. 481–488, 1973.

[20] G Schalk, D J McFarland, T Hinterberger, N Birbaumer, and J R Wolpaw, "BCI2000: a general-purpose brain-computer interface (BCI) system," *IEEE Trans Biomed Eng*, vol. 51, no. 6, pp. 1034–1043, 2004.