Proceedings of the
8th Graz Brain-Computer Interface Conference 2019 DOI: 10.3217/978-3-85125-682-6-63

A SELF-PACED P300 SPELLER WITH IMPROVED TYPING SPEED USING CONTINUOUS STIMULUS PRESENTATIONS

W. Speier[1,2], C. Arnold[1,2], N. Chandravadia[3], D. Roberts[4], S. Pendekanti[3], A. Marttini[3], N. Pouratian[3,4,5,6]

[1] Department of Radiology, University of California, Los Angele, United States
[2] Medical Imaging and Informatics Group, University of California, Los Angeles, United States
[3] Neuroscience Interdepartmental Program, University of California, Los Angeles, United States
[4] Department of Neurosurgery, University of California, Los Angeles, United States
[5] Department of Bioengineering, University of California, Los Angeles, United States
[6] Brain Research Institute, University of California, Los Angeles, United States

E-mail: Speier@ucla.edu

ABSTRACT: Communication is a critical human function that can be severely compromised in patients with neurological diseases such as amyotrophic lateral sclerosis (ALS). The P300 speller is a brain-computer interface (BCI) device that restores communication in these patients by detecting evoked responses in subjects' electroencephalography signals. One of the bottlenecks of these systems is the pause after character selections. This pause has been necessary for the P300 speller because it signals users that a character selection has been made and gives them time to transition to the next character. If this pause is too long, the system is slowed down unnecessarily. If it is too short, stimuli for the next character begin before the user is ready. We propose a system that does away with the pause entirely and continually flashes stimuli. We employ a joint model that determines the target characters as well as the transition times so that users can change between characters at their own pace. A preliminary study on eight subjects showed a selection rate of 16.35 characters/minute and an average accuracy of 94.85%, both significant improvements over performance in an equivalent system with standard flashing. These results suggest that the P300 speller could be improved by implementing a continuous flashing paradigm.

INTRODUCTION

Communication is a critical human function that can be severely compromised in patients with neurological diseases such as amyotrophic lateral sclerosis (ALS). Brain-computer interfaces (BCI) provide a unique opportunity to restore communication in severe cases where traditional augmentative and alternative communication devices are unusable due to lack of motor control. The P300 speller is one BCI system that provides this communication ability by detecting evoked responses after flashing characters on a visual display [1]. Limits to the signal to noise ratio require multiple stimuli before making a selection, leading to slow typing rates. Several projects have improved performance by incorporating optimizations such as varying the dimensions of the character matrix [2], [3]; optimizing system parameters [4], [5]; and employing various signal-processing methods [6], [7].

Recent work has involved the incorporating of language models into the classifier [8]. This movement in BCI research integrates knowledge about the domain of natural language to improve classification, similar to methods used in other domains such as speech recognition [9]. Several BCI studies have shown incremental improvements in system speed and accuracy using n-gram language models, first using Naïve Bayes [10], [11] and later using a partially observable Markov decision process [12] and a hidden Markov model [13]. Recently, a particle filter (PF) algorithm made the use of more complicated language models possible, which was shown to have superior results [14].

One area that language models have had particular impact is the ability to provide prior probabilities for dynamic stopping [8]. These methods compute the probability distribution over all characters after each stimulus and select that character after the probability of a character reaches a threshold [11], [15]. This method has the potential to drastically improve typing speed as selections are made once the system is confident in a selection rather than continuing until a set number of stimuli is reached. It also allows the system to spend more time on selections where the confidence is lower so that it can collect more information rather than forcing a selection.

A P300 speller with continuous stimulus presentations is an extension of the dynamic stopping paradigm in that it allows subjects to move at their own pace rather than forcing them to wait a predefined period of time between character selections. It also allows them to take more time if they are unable to find a character or are unsure of what they want to say next. It accomplishes this task by

 DOI: 10.3217/978-3-85125-682-6-63

computing the joint probability distribution over possible target characters and transition times between characters after every stimulus presentation.

In this study, we propose a novel P300 speller system that incorporates continuous stimulus presentations. This system was incorporated into the BCI2000 [16] framework using a previously published particle filtering classifier [14]. A pilot study of eight healthy subjects was conducted to compare typing performance using this paradigm to the traditional P300 speller system.

MATERIALS AND METHODS

Data Collection All data collection was performed using g.tec amplifiers, active EEG electrodes, and electrode cap (Guger Technologies, Graz, Austria); sampled at 256 Hz; referenced to the left ear; grounded to AFz; and filtered using a passband of 0.1 – 60 Hz. Additional artifact detection (e.g., eye blinks) was not performed as it was left to the classifier to determine whether a signal contained a valid ERP. The electrode montage consisted of a previously reported set of 32 electrodes [5]. The subjects for the online study consisted of eight healthy volunteers with normal or corrected to normal vision between the ages of 20 and 35. The system used a 6 x 6 character grid, famous faces stimuli [17], row and column flashes, and a stimulus onset asynchrony of 125 ms. Using the standard interface, a 3.5-second gap was included between characters to allow subjects time to find the next character in the sequence.

To validate our methods, we implemented the continuous P300 speller in BCI2000 [16]. In a pilot study, eight subjects used the continuous speller to type out the sentence: "THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG." Eye tracking was not used, so training sessions consisted of two copy spelling sessions using the traditional P300 speller. When using the online speller, subjects were instructed to focus on each character for approximately three seconds (about two complete sets of flashes) before moving to the next character. Feedback was turned off to avoid distraction.

Continuous Speller The continuous speller formulation is similar to the traditional p300 speller with dynamic stopping [11]. After each flash, the probability distribution across the set of characters is estimated based on a set of observation probabilities and transition probabilities based on a language model. The main difference is the introduction of a variable d_t that represents the amount of time the subject will remain at the current state, x_t.

$$p(x_t, d_t | y_t, x_{0:t-1}, d_{t-1}) \propto p(x_t, d_t | x_{0:t-1}, d_{t-1}) p(y_t | x_t)$$

In this model, d_t is reduced after each flash until it reaches zero. At that point, the transition probability is determined by the language model as in the traditional speller.

$$p(x_t, d_t | x_{0:t-1}, d_{t-1}) = \begin{cases} p(x_t | x_{0:t-1}) p(d_t | x_t) & d_{t-1} = 0 \\ \delta_{x_{t-1}}^{x_t} \delta_{d_t}^{d_t+1} & d_{t-1} > 0 \end{cases}$$

A Gaussian distribution was used to estimate the time taken to transition between characters, $p(d_t|x_t)$. Initially, this distribution was set empirically at a mean of one second and a standard deviation of 0.5 seconds (with a minimum of zero seconds). To further tailor this distribution, expectation maximization was used to find the distribution for each subject in an unsupervised manner. This process was similar to the methods used in previous studies that trained the P300 speller with unlabeled data [18], [19]. In this version, the empirical distribution was used to find preliminary labels for and transition times for all characters. These labels were then used to find more accurate parameters for the transition distribution. Iteration between these two steps continued until the distribution stabilized.

Because it will always take time for the subject to find new characters in the grid, an additional state needs to be made for when the subject is transitioning between letters. During this time, the stimulus responses will look different from those when the subject's attention is on a character. The observation probability distribution needs to take this into account.

$$p(y_t|x_t) = \begin{cases} f(y_t|\mu_a, \sigma_a^2) & x_t \in A_t \\ f(y_t|\mu_{trans}, \sigma_{trans}^2) & x_t = x_{trans} \\ f(y_t|\mu_n, \sigma_n^2) & otherwise \end{cases}$$

Language Model: A language model is used to determine the transition probabilities, $p(x_t|x_{0:t-1})$. This probability can be simplified using the nth-order Markov assumption to create a n-gram model [11], [20]. While n-gram models are able to capture character patterns, they allow for strings that are not valid words on the language. A probabilistic automaton (PA) creates a stronger prior by creating states for every substring that starts a word in the corpus. Thus, the word "the" would result in four states: "t", "th", "the", and the start state which corresponds to a blank string. Each state then links to every state the represents a superstring that is one character longer. Thus, the state "th" will link to the states "the" and "tha" (Fig. 1).

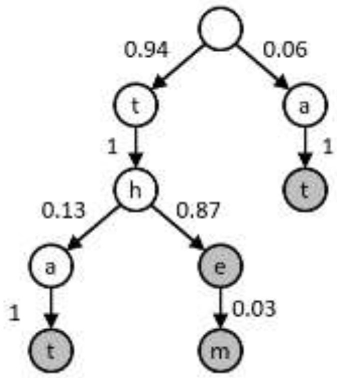

Figure 1: Example language model containing only the words at, that, the, and them.

Transition probabilities are determined by the relative frequencies of words starting with the states' substrings in the Brown English language corpus [21].

 DOI: 10.3217/978-3-85125-682-6-63

Particle Filter: As more sophisticated language models are used, the ability to fully represent the probability distribution over possible output sequences becomes impractical. Particle filtering (PF) is a method for estimating this distribution by creating a set of realities (called particles) and projecting them through the model based on the observed data [22], [23]. Each of these particles contains a reference to a state in the model, a history of previous states, and an amount of time that the particle is going to remain in the current state. The distribution of states occupied by these particles represents an estimation of the probability distribution.

When the system begins, a set of P particles is generated and each is associated with the root node of the language model. At the start of a new character, samples are drawn from the proposal distribution defined by the transition probabilities from the previous state.

$$x_t^{(L)} \sim p\left(x_t \middle| x_{t-1}^{(L)}\right)$$

The time that the particle will stay in that state is drawn from a distribution representing how long the subject is expected to spend looking at that character. After each stimulus response, the probability weight is computed for each of the particles.

$$\widehat{w}_t^{(L)} \propto \widehat{w}_{t-1}^{(L)} p\left(y_t \middle| x_t^{(L)}\right)$$

The weights are then normalized and the probability of possible output strings is found by summing the weights of all particles that correspond to that string. When feedback is enabled, the string with the highest probability is then displayed to the user. The effective number of particles is then computed.

$$P_{eff} = \frac{1}{\sum_i \left(\widehat{w}_t^{(L)}\right)^2}$$

If the effective number falls below a threshold, P_{thresh}, a new set of particles are drawn from the particle distribution.

After each stimulus, the amount of time for a given particle to remain in a state is decremented. Once that counter reaches zero, the particle transitions to a new state in the language model based on the model transition probabilities $p(x_t|x_{0:t-1})$.

Evaluation: Performance of the system is evaluated in terms of the speed and accuracy of typing characters. Selection rate (SR) is the average number of characters selected per minute and accuracy (ACC) is the percentage of those characters that match the target sentence. Because of the tradeoff between speed and accuracy, performance is also evaluated in terms of information transfer rate (ITR), which measures the bits of information conveyed through the output message divided by time. ITR has been shown to be an imperfect measure of BCI communication due to assumptions about the uniform probability across characters and the independence of selections [24], [25]. However, it remains a standard metric used in the BCI field and can compare relative performance on identical sequences. Calculation of ITR starts by computing the number of bits of information contained per symbol in the output sequence.

$$B = \log(N) + ACC\log(ACC) + (1 - ACC)\log\left(\frac{1 - ACC}{N - 1}\right)$$

Where N is the number of characters in the grid (36). ITR can then be found by multiplying the by SR. Significance was tested using Wilcoxon rank-sum tests.

RESULTS

Even though subjects were all instructed to type at the same speed (three seconds per letter), each subject typed at a slightly different pace (Fig. 2).

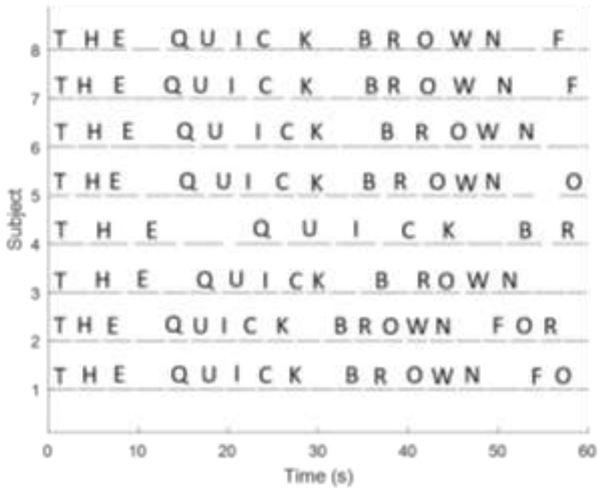

Figure 2: Output for each subject during the first 60 seconds of use of the continuous speller. Each subject was attempting to spell “THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG.”

Table 1. Selection rate (SR), typing accuracy (ACC) and information transfer rate (ITR) for the traditional P300 speller and the continuous speller (CS).

	SR (characters/minute)		ACC (%)		ITR (bits/minute)	
Subject	P300	CS	P300	CS	P300	CS
1	14.06	17.79	93.33	100.00	62.92	89.89
2	13.81	18.18	95.00	90.91	63.91	77.50
3	13.77	16.51	100.00	91.11	71.21	75.74
4	7.58	11.94	75.00	100.00	23.33	61.72
5	13.43	16.28	95.00	90.70	62.13	69.12
6	12.63	16.31	100.00	95.35	65.30	72.64
7	13.69	17.45	65.00	97.73	33.42	85.47
8	13.04	17.09	75.00	93.02	40.13	75.98
Mean	12.71	16.35	89.05	94.85	54.60	76.01

 DOI: 10.3217/978-3-85125-682-6-63

Overall, subjects types an average of 16.35 characters/minute using the continuous speller, which was a significant improvement over the 12.71 characters/minute achieved using the traditional P300 speller (p=0.004, Tab. 1). When using a generic model for dwell and transition times, the average selection accuracy for the continuous speller (85.4%) was lower than the P300 speller (89.05%), although the difference was not statistically significant (p=0.42). The resulting ITR for the continuous speller (64.5 bits/minute) was higher than that of the P300 speller (54.6 bits/minute), although the difference was not statistically significant (p=0.13). Once subject-specific transition distributions were learned, the accuracy rose to 94.9%, resulting in an average ITR (76.0 bits/minute) that was significantly higher than that of the P300 speller (p=0.004).

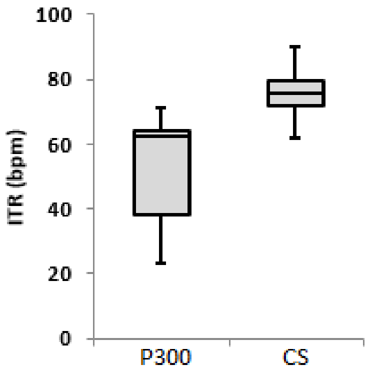

Figure 3: Boxplots of ITR values using the standard P300 and the continuous speller (CS).

In general, subjects' gaze times were close to the suggested value of three seconds (Tab. 2). One subject (subject 4), however, spent over one second longer per letter on average. The other subjects' averages were between 2.93 and 3.32 seconds. The average time spent transitioning between characters was also fairly consistent across the subjects with an average of 0.36 seconds.

Table 2: Average gaze and transition times

Subject	Gaze Time (s)	Transition Time (s)
1	3.28 (0.58)	0.14 (0.06)
2	2.93 (0.65)	0.31 (0.23)
3	2.98 (0.75)	0.35 (0.26)
4	4.39 (0.73)	0.59 (0.28)
5	3.13 (0.70)	0.41 (0.31)
6	3.32 (0.61)	0.32 (0.21)
7	3.06 (0.61)	0.32 (0.27)
8	3.07 (0.63)	0.43 (0.31)
Mean	3.27 (0.47)	0.36 (0.13)

DISCUSSION

The average typing speed achieved using the continuous speller was 16.35 characters/minute, which was significantly faster than the 12.71 characters/minute achieved by the same subjects using the traditional P300 speller. The bit rate for the P300 speller was similar to values achieve in previous studies using similar systems [14].

Only one subject failed to achieve a typing speed greater than 16 characters/minute. While this subject was still able to type at a speed faster than she did using the standard P300 speller, it was substantially slower than the other subjects in this study, taking over a minute longer to type the sentence than any other subject. This difference is likely because she dwelled longer on each character than the suggestion of three seconds.

The results achieved using continuous spelling demonstrate that this system has the potential to outperform existing ERP-based BCI systems. However, the results from this study are likely far from optimal, as most of the parameters were not optimized for this system. For instance, the flashing rate and ISI values used were optimized for the traditional p300 speller, and do not necessarily reflect the best configuration for this system. We also used the row/column flashing paradigm as it is the standard stimulus paradigm for the P300 speller. There are various other paradigms that have been introduced, including the checkerboard [2], combinatorial [3], and asynchronous [26] paradigms, which could improve typing performance in this system as well. Finally, the strategy of looking at each character for approximately three seconds was chosen empirically as it gave approximately four positive stimuli for each character, which we felt would be sufficient to make an accurate classification. It is possible that less information is needed, which could allow the system to achieve even greater speeds.

This study was conducted using healthy volunteers who did not have the same constraints as "locked-in" patients, such as restrictions on eye gaze. While a similar P300 speller system was previously tested in the ALS population [27], it is unclear whether the continuous flashing will be more difficult and therefore offset the gains seen by applying continuous flashing. The healthy subjects in this study generally had no problems with the additional cognitive task, and therefore appreciated the added speed that continuous flashing afforded. However, it is possible that his additional task will make the system more taxing for ALS patients, which could make it less practical despite the performance increase. Future studies in the ALS population should be conducted to determine how these results in healthy subjects translate to affected population. If continuous flashing is a hindrance to some subjects, those subjects could continue to use the traditional P300 speller with the continuous flashing paradigm included as an option for those subjects who would benefit.

The results presented in this study are promising, but they represent offline performance which does not include several factors that occur in an online implementation. For instance, offline systems do not include feedback to

 DOI: 10.3217/978-3-85125-682-6-63

the user, which can provide additional motivation or allow the user to adjust their strategy. Particularly in this system, feedback can be difficult to implement because the user does not have specific built in to allow for checking past input. Also, the fact that the system optimizes over multiple characters at once means that the system will likely make changes to the displayed text, which can distract the user's focus from the current character. The lack of feedback can make the task more difficult, however, as subjects can lose track of where they are in the target word or phrase. They are also unable to adjust their strategy such as slowing down if the system is unable to make accurate selections.

CONCLUSION

Overall, continuous stimulus presentation allowed subjects to type an average of 16.35 characters/minute with an accuracy of 94.85%, resulting in an average ITR of 76.01 bits/minute, all significantly higher than the values achieved using the standard P300 speller. Future work involves optimizing system parameters for the continuous flashing paradigm, and implementing feedback in an online system.

REFERENCES

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

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. 510–23, Dec. 1988.

[2] 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. 1109–1120, Jul. 2010.

[3] J. Jin, P. Horki, C. Brunner, X. Wang, C. Neuper, and G. Pfurtscheller, "A new P300 stimulus presentation pattern for EEG-based spelling systems.," *Biomed. Tech. (Berl).*, vol. 55, no. 4, pp. 203–10, Aug. 2010.

[4] D. J. McFarland, W. A. Sarnacki, G. Townsend, T. Vaughan, and J. R. Wolpaw, "The P300-based brain-computer interface (BCI): Effects of stimulus rate," *Clin. Neurophysiol.*, vol. 122, no. 4, pp. 731–737, Apr. 2011.

[5] J. Lu, W. Speier, X. Hu, and N. Pouratian, "The effects of stimulus timing features on P300 speller performance," *Clin. Neurophysiol.*, vol. 124, no. 2, pp. 306–314, Feb. 2013.

[6] M. Kaper, P. Meinicke, U. Grossekathoefer, T. Lingner, and H. Ritter, "BCI Competition 2003--Data set IIb: support vector machines for the P300 speller paradigm.," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 1073–6, Jun. 2004.

[7] N. Xu, X. Gao, B. Hong, X. Miao, S. Gao, and F. Yang, "BCI Competition 2003 - Data Set IIb: Enhancing P300 Wave Detection Using ICA-Based Subspace Projections for BCI Applications," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 1067–1072, 2004.

[8] W. Speier, C. Arnold, and N. Pouratian, "Integrating language models into classifiers for BCI communication: a review," *J. Neural Eng.*, vol. 13, no. 3, p. 31002, 2016.

[9] F. Jelinek, *Statistical methods for speech recognition*. MIT press, 1998.

[10] P.-J. Kindermans, H. Verschore, and B. Schrauwen, "A Unified Probabilistic Approach to Improve Spelling in an Event-Related Potential-Based Brain–Computer Interface," *Biomed. Eng. IEEE Trans.*, vol. 60, no. 10, pp. 2696–2705, 2013.

[11] W. Speier, C. Arnold, J. Lu, R. K. Taira, and N. Pouratian, "Natural language processing with dynamic classification improves P300 speller accuracy and bit rate," *J. Neural Eng.*, vol. 9, no. 1, p. 16004, Feb. 2011.

[12] J. Park and K.-E. Kim, "A POMDP approach to optimizing P300 speller BCI paradigm," *Neural Syst. Rehabil. Eng. IEEE Trans.*, vol. 20, no. 4, pp. 584–594, 2012.

[13] W. Speier, C. Arnold, J. Lu, A. Deshpande, and N. Pouratian, "Integrating language information with a hidden markov model to improve communication rate in the P300 speller," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 3, pp. 678–684, 2014.

[14] W. Speier, C. W. Arnold, A. Deshpande, J. Knall, and N. Pouratian, "Incorporating advanced language models into the P300 speller using particle filtering.," *J. Neural Eng.*, vol. 12, no. 4, p. 46018, Aug. 2015.

[15] H. Serby, E. Yom-Tov, and G. F. Inbar, "An improved P300-based brain-computer interface.," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 13, no. 1, pp. 89–98, Mar. 2005.

[16] 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. 1034–43, Jun. 2004.

[17] T. Kaufmann, S. M. Schulz, A. Köblitz, G. Renner, C. Wessig, and A. Kübler, "Face stimuli effectively prevent brain-computer interface inefficiency in patients with neurodegenerative disease.," *Clin. Neurophysiol.*, vol. 124, no. 5, pp. 893–900, May 2013.

[18] W. Speier, J. Knall, and N. Pouratian,

 DOI: 10.3217/978-3-85125-682-6-63

"Unsupervised training of brain-computer interface systems using expectation maximization," in *Neural Engineering (NER), 2013 6th International IEEE/EMBS Conference on*, 2013, pp. 707–710.

[19] P.-J. Kindermans, M. Schreuder, B. Schrauwen, K.-R. Müller, and M. Tangermann, "True Zero-Training Brain-Computer Interfacing–An Online Study," *PLoS One*, vol. 9, no. 7, p. e102504, 2014.

[20] C. D. Manning and H. Schütze, *Foundations of statistical natural language processing*. MIT Press, 1999.

[21] W. N. Francis and H. Kucera, *Brown Corpus Manual*. Providence, RI: Dept of Linguistics, Brown University, 1979.

[22] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *Signal Process. IEEE Trans.*, vol. 50, no. 2, pp. 174–188, 2002.

[23] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in *IEE Proceedings F (Radar and Signal Processing)*, 1993, vol. 140, no. 2, pp. 107–113.

[24] P. Yuan, X. Gao, B. Allison, Y. Wang, G. Bin, and S. Gao, "A study of the existing problems of estimating the information transfer rate in online brain–computer interfaces," *J. Neural Eng.*, vol. 10, no. 2, p. 26014, 2013.

[25] W. Speier, C. Arnold, and N. Pouratian, "Evaluating True BCI Communication Rate through Mutual Information and Language Models.," *PLoS One*, vol. 8, no. 10, p. e78432, Jan. 2013.

[26] G. Townsend and V. Platsko, "Pushing the P300-based brain–computer interface beyond 100 bpm: extending performance guided constraints into the temporal domain," *J. Neural Eng.*, vol. 13, no. 2, p. 26024, 2016.

[27] W. Speier, N. Chandravadia, D. Roberts, S. Pendekanti, and N. Pouratian, "Online BCI Typing using Language Model Classifiers by ALS Patients in their Homes," *Brain-Computer Interfaces*, vol. 4, no. 1–2, pp. 1–8, Nov. 2016.