2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Automatic Control of a Wheelchair using a Brain Computer Interface and Real-time Decision-making

Frida Heskebeck*[1] and Pex Tufvesson*[2]

***Abstract*— In this study, we simulate the automatic control of an electric wheelchair for indoor Pac-Man-style navigation using solely thought commands. We delve into the decision-making mechanisms of an operational EEG-based brain computer interface that employs a visual oddball paradigm. We investigate strategies to enhance the efficiency of decision-making processes, aiming to accelerate response times while maintaining a defined error rate. Furthermore, we explore methodologies to decrease the user's cognitive load by reducing the number of stimuli needed before an action.**

***Index Terms*— Applications in neuroscience; Statistical learning; Emerging control applications**

I. INTRODUCTION

Controlling a wheelchair via a Brain Computer Interface (BCI) enables persons with limited motor skills to move around freely. This provides greater autonomy and quality of life to individuals with profound motor impairments, such as individuals afflicted by locked-in syndrome [10]. Electroencephalography (EEG) is one of the most popular methods to record brain activity in BCIs, enabling the interpretation of user intentions [7, pp. 15–25]. EEG has been used for controlling a wheelchair in numerous studies, with a recent advancement using a combination of EEG alongside other input modalities, such as muscle and eye tracking, to provide more dependable control options [6]. Another example uses a camera to identify surrounding objects, allowing the user to select their desired destination [11].

A few well-known patterns of brain activity, so-called BCI paradigms, can be used in BCIs. One is the motor imaginary (MI) paradigm, where the user imagines the movement of a limb, such as the right hand. One is the steady-state visual evoked potentials (SSVEP) paradigm, where the frequency of attenuated flickering stimuli is identified. Another example is the P300 paradigm, which relies on the brain's spontaneous response to anticipated and unanticipated stimuli, specifically the detection of P300 signals. In this paper, we use the P300 paradigm. The scenario is such that when the user needs to decide on a direction to turn, they are presented with arrows symbolizing various choices, like continuing straight ahead or turning left. The brain will involuntarily elicit a P300 signal when the target stimulus (the arrow representing the user's intended choice) is shown. This P300 signal can be detected and used for identifying the target stimulus. The difficulty lies in the fact that P300 signals are prone to noise, and repeated measurements are needed to decode the user's intent. The cognitive load for reacting to stimuli for a prolonged time is high, which is tiresome for the user. This paper studies how to reduce the number of stimuli in a BCI system that acts upon the measured signals, while guaranteeing the performance of the system. In our setup, the controller decodes the user's intent, known as the target stimuli, and determines which stimuli to present to the user, referred to as the control signal. In this paper, we leverage the stimuli selection algorithm presented in [12] by concentrating on the performance assurances that such a controller can give.

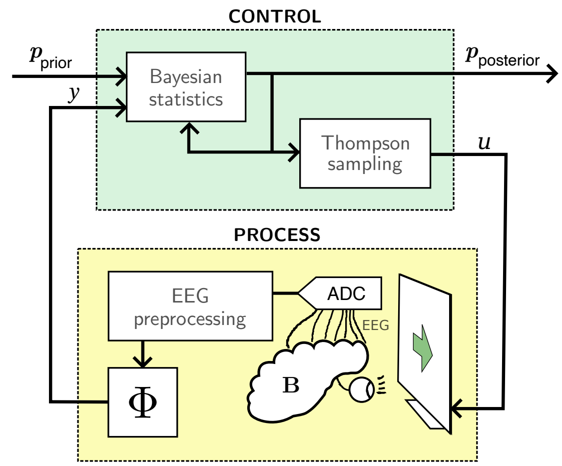


Fig. 1: Block diagram of the control loop. The process consists of the human user's brain $\mathbf{B}$, Analog to Digital Conversion (ADC), EEG preprocessing, and a machine learning feature map Φ, outputting the signal y corresponding to how well this specific stimulus matches the user's intention. The $\boldsymbol{p}_{\text{prior}}$ is used as input to the controller, and the $\boldsymbol{p}_{\text{posterior}}$ is the output, representing the probability for each possible choice. Before the user decides, the number of choices and their initial probabilities $\boldsymbol{p}_{\text{prior}}$ are estimated, either through historical data or through situational analysis. The probabilities $\boldsymbol{p}_{\text{posterior}}$ are then improved by showing visual stimuli to the user, and the user's EEG response gives a hint of their preference. When the error rate is low enough, the Action, decided by $\boldsymbol{p}_{\text{posterior}}$, can be taken.

* The authors contributed equally and share first authorship.

[1]Frida Heskebeck is with the Faculty of Engineering, Department of Automatic Control, Lund University, Sweden. Email: frida.heskebeck@control.lth.se

[2]Pex Tufvesson is with Ericsson Research, Lund, Sweden and with the Faculty of Engineering, Department of Automatic Control, Lund University, Sweden. Email: pex.tufvesson@ericsson.com

The authors are members of the ELLIIT Strategic Research Area at Lund University. This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

The scope of this problem formulation extends beyond wheelchair control. At its core, the challenge involves swiftly identifying the user's intent within a constrained time-frame, minimizing the number of stimuli presented. However, wheelchair control is a concrete illustrative example of the problem, which we will primarily focus on in this paper. Furthermore, an autonomous wheelchair solves an actual problem faced by persons with limited motor skills, rendering it a pertinent example.

Given the critical importance of safety, most BCI-controlled wheelchairs incorporate collision-avoidance systems based on sensors like infrared or ultrasonic rangefinders. These systems override or correct the wheelchair's path to prevent a collision [10]. Sensors can also be used in a corridor setting to detect upcoming turns that need a decision. This situation could be compared to the 80's arcade game Pac-Man (パックマン) [9].

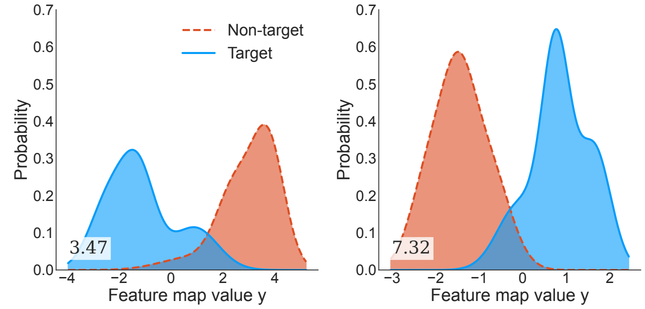
This paper is a feasibility study of how the number of shown stimuli before an action correlates to the *error rate* of the action where the controller's task is to decode the intent of the user and choose the next stimulus to show the user (the control signal). The paper also studies how *branch prediction* and *threshold early completion* could be used to reduce the number of stimuli, i.e., shorten decision time and thus reduce the user's cognitive load. We assume that the navigation system would be able to keep a steady course in an indoor setting (using technologies like lidar, depth-sensing cameras, and SLAM algorithms [4]) and detect upcoming turns from a five-meter distance for a wheelchair moving at 1 m/s. We have chosen to show visual stimuli to the user twice per second. This means that the controller could show up to ten stimuli for the user before identifying the user's intention. If the user's intention is not identified within those ten stimuli, the wheelchair must stop or guess what the user wants, resulting in a sub-par user experience. On the contrary, if the BCI can identify the user's intention in fewer than ten stimuli, the wheelchair can move faster.

In our simulations, we use publicly available BCI data from a P300 EEG experiment for target and nontarget stimuli (see details in the next section). Even though the data is not from a wheelchair setting, the brain's response to target and non-target stimuli is the same regardless of the specific details of the stimuli [7, pp. 503–506]. Thus, it is motivated to use our selected dataset. EEG data from other BCI paradigms, such as SSVEP, have been in other studies [6]. We chose to use the P300 paradigm since it allows all users with a functioning vision to use the system and is generally less tiresome than the SSVEP paradigm due to the less intrusive way of presenting stimuli. Other types of inputs, such as joystick movement or gaze tracking, could also have been used to control the wheelchair but are not considered in this paper as they are not pure BCI solutions.

II. Methods

A. Control structure

The control structure for a BCI-controlled wheelchair can be described with a block diagram as in Fig. 1, where the internal flowchart of the process and controller is also visible. Prior probabilities $\boldsymbol{p}_{\text{prior}}$ go into our controller, which selects a stimulus u, using Thompson sampling, to show the user. The Thompson algorithm used for stimuli sequence selection is discussed in [12]. Statistically, in our setup, the brain is a highly non-linear function $\mathbf{B}(u)$, taking an image as input u corresponding to a certain choice, producing a time series of 500 ms of EEG data, represented as a matrix $\mathbf{X}$ of size (channels, samples). We decimate the 512 Hz 16 channel $\mathbf{X}$ by a factor 10, bandpass filter 0.5-20 Hz, and normalize the amplitude in the preprocessing block. The feature map Φ is then applied, producing the feature map value $y \in \mathbb{R}$. This y is interpreted into a target probability in the controller via a Gaussian mixture model (GMM), and the posterior probabilities $\boldsymbol{p}_{\text{posterior}}$ are then updated using Bayesian statistics. The action the wheelchair takes is based on $\boldsymbol{p}_{\text{posterior}}$.



(a) Φ ERPcov TS LR norm (b) Φ SVM score

Fig. 2: Two feature maps Φ and their resulting PDFs, where red is non-target and blue is target, in (a) a Riemannian tangent space logistic regressor normalized with z-scoring, and in (b) the distance to the hyperplane in a Support Vector Machine (SVM). The Kullback-Leibler divergence score, printed in the lower left corner of each plot, states the degree of separation between the red non-target and blue target areas, and a higher value means better separation.

B. Decoding the user intent from EEG data

As described above, the controller updates the probabilities $\boldsymbol{p}_{\text{posterior}}$ based on the feature map value $y \in \mathbb{R}$ which is the EEG data mapped to y with a feature map Φ, as described in [12]. Fig. 2 shows the data distribution of y for target and non-target stimuli for two different feature maps. In this paper, the SVM score feature map is used. The target and non-target distributions' separation can be rated with the Kullback-Leibler divergence score [12]. Since the distributions are overlapping, it is clear that only one sample from the target/non-target choice may not always be enough to decode the user intent. Showing more stimuli generally means that the probability $\boldsymbol{p}_{\text{posterior}}$ for the user's intended choice gets closer to one, and the probability for the other choices goes to zero. The rate at which this will happen corresponds to the Kullback-Leibler divergence score of the PDF.

C. Statistical distribution of feature maps

1) Bayes' theorem: The formula for Bayes' theorem is

$$\boldsymbol{p}_{\text{posterior}} = \frac{\boldsymbol{p}(y|x)\boldsymbol{p}_{\text{prior}}}{p(y)}.$$

The controller uses this formula to update the estimated probabilities $\boldsymbol{p}$ of the choices.

2) Gaussian mixture models: Mathematically, the GMM can be represented as:

$$p(\mathbf{x}) = \sum_{i=1}^{K} \pi_i \mathcal{N}(\mathbf{x}|\mu_i, \mathbf{\Sigma}_i),$$

where K is the number of Gaussians in the mixture, π_i denotes the mixing coefficient of the i^{th} Gaussian ($\sum_{i=1}^{K} \pi_i = 1$), μ_i is its mean, and $\mathbf{\Sigma}_i$ is its covariance matrix. We use the expectation-maximization (EM) algorithm [2] to fit a two-dimensional GMM with three Gaussian components for the target and non-target data respectively [3, 8].

D. Error rate

The error rate is used to evaluate the real-time decision-making limitations for a BCI-controlled wheelchair. The error rate states the ratio of wrong actions the controller takes. In other words, an error rate of 10^{-3} means that one time out of a thousand, the wheelchair will make a turn the user did not intend to take. Due to the collision-avoidance system the wheelchair will never crash into a wall but might continue straight when the user wanted to turn. Which error rate is considered acceptable is up for debate. In this paper, we have chosen an error rate of 10^{-3} and 50000 Monte Carlo simulations of the decision process to provide error rate data. The decision process refers to the process of the controller selecting a stimulus u, receiving input y from the process, updating the probabilities $\boldsymbol{p}_{\text{posterior}}$, selecting a new stimulus u, and so on. The number of choices for a decision is denoted C. For example, if the user can choose to turn left, right, or continue straight, C equals 3.

E. Branch prediction

If the controller knows how the user usually turns in an intersection, this information can be used as prior knowledge for the probabilities, $\boldsymbol{p}_{\text{prior}}$, meaning that fewer stimuli are needed before identifying the target choice. This is done with so-called branch prediction, a term borrowed from computer architecture used for achieving high performance in modern pipelined microprocessors. The simulations in this paper for branch prediction are run with the condition that one choice has 90 % probability and the remaining choices have even probabilities. For instance, if $C = 3$, then $\boldsymbol{p}_{\text{prior}} = [0.9, 0.05, 0.05]$ and in the intersection where the user's office is to the left, $\boldsymbol{p}_{\text{prior}}$ implies that the user turns left into their office 90 % of the time, goes straight 5 % of the time, and turns right 5 % of the time. These percentages are illustrative examples utilized in this paper. In an actual scenario, these figures are revised each time the user chooses a direction at an intersection. Thus, each intersection has its own $\boldsymbol{p}_{\text{prior}}$.

F. Threshold early completion

To improve user experience, as few stimuli as possible should be shown. In other words, once the controller knows, with a certain accuracy, which action to take, it should stop showing stimuli. By choosing a certainty threshold, ϵ, for the condition

$$1 - \max(\boldsymbol{p}) < \epsilon, \tag{1}$$

where $\max(\boldsymbol{p})$ is the current highest probability for the choices, the controller can stop showing stimuli when it knows the target choice and the wheelchair can take the corresponding action. Eq. (1) gives no guarantee that the correct action is taken, but indicates when the controller can consider being sufficiently sure of the user's intended choice.

G. Simulations

The simulation in this paper takes some ten minutes to run and uses roughly 3GB of downloaded EEG data and 6GB of RAM.

1) Dataset: The Brain Invaders 2013 dataset from the GIPSA-lab, [1], was used for empirical analysis. The dataset contains EEG data from many users recorded when the user was shown target and non-target stimuli from a Space Invaders game. The MOABB Python package was used to access the data [5].

2) Machine learning approach: In all simulations, we have used subject #1 from the dataset. For training the feature map Φ, sessions 1 and 2 have been used. For validating the hyperparameters of the feature map Φ, sessions 3 and 4 have been used. For creating the probability density functions, seen in Fig. 2, which are used for the Bayesian probability estimation in the controller, we have used session 5. To be able to generate an unlimited amount of test data for simulations, we augment data from session 6 by training a GMM and creating test data by sampling from this GMM. Using data from multiple users for transfer learning of the feature maps is discussed in [12].

3) Source Code: The source code, MIT licensed, can be downloaded from `bci.lu.se/wheelchair`

III. Results

The results are generated from 50000 Monte Carlo simulated user decisions based on real recorded EEG data, using the controller and methods described above.

A. Error rate

Fig. 3 shows the error rate versus the number of shown stimuli. The more choices C the user has, the harder it gets to estimate the user's intent. For example, if there are two possible choices, $C = 2$, and we aim for an error rate of 10^{-3}, five stimuli will be needed. If there are six choices, $C = 6$, 13 stimuli will be needed. The required number of Monte Carlo simulations depends on the desired accuracy, especially for small error rates such as "one in 10,000", seen below the dotted line in Fig. 3. Using fewer simulations would make these lines deviate for higher error rates.

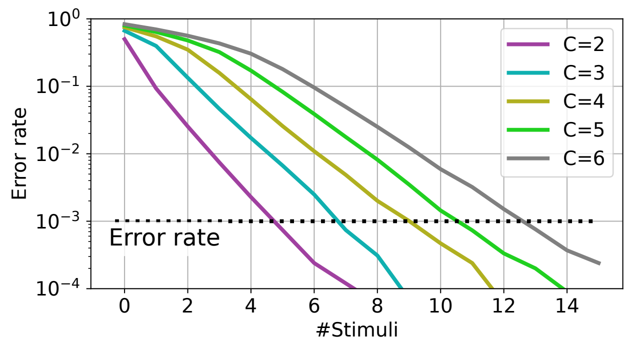


Fig. 3: Error rate as a function of the number of shown stimuli for different number of available choices, C. The error rate shows the risk of the wheelchair taking the wrong action. If more choices are available, more stimuli will need to be shown to the user before reaching our selected error rate of 10^{-3} compared to when fewer choices are available.

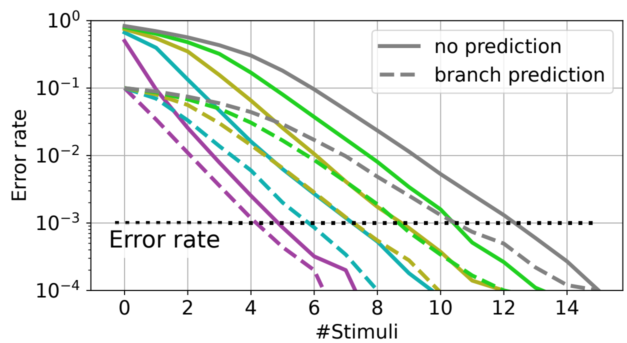


Fig. 4: Error rates when using branch prediction. The solid lines correspond to the lines in Fig. 3, and the dashed lines are the error rates using branch prediction, where the user nine times out of ten makes the predicted choice.

B. Branch prediction

Fig. 4 shows the error rate over the number of stimuli when using branch prediction. Regardless of the number of choices available, branch prediction leads to fewer needed stimuli to achieve the same error rate. For example, with six choices, $C = 6$, and no branch prediction, 13 stimuli are needed to reach an error rate of 10^{-3}. However, if branch prediction is used, only 11 stimuli are needed.

C. Threshold early completion

Fig. 5 shows $1 - \max(\boldsymbol{p})$ versus the number of stimuli for $C = 3$. The red points show simulations where a non-target choice has the highest probability, meaning that if an action were to be taken, the wheelchair would make a wrong turn. However, the certainty threshold ϵ is chosen such that few enough red points are below the threshold, thus preserving the chosen error rate. The colored areas show how the action is decided. If the user's intent is not identified after seven shown stimuli, an action will be taken based on the fact that for $C = 3$ seven stimuli is enough to guarantee an error

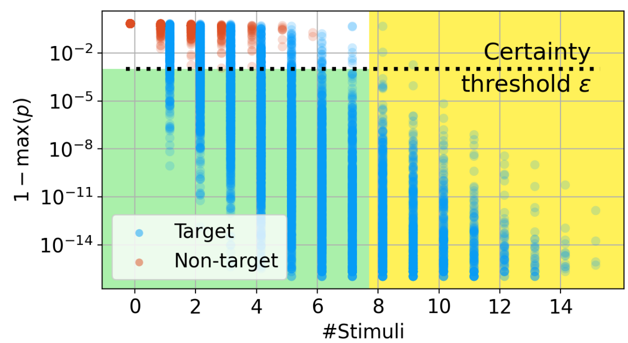


Fig. 5: Simulations of threshold early completion for $C = 3$. The y-axis is $1-\max(\boldsymbol{p})$ plotted against the number of shown stimuli. The *red dots* are simulations where the estimation of the user's intended choice is currently wrong, and the *blue dots* are simulations where the estimation is correct. The *horizontal black dotted line* is our chosen certainty threshold ϵ. The points in the top-left *white area* represent simulations where the controller is not yet sure about the user's intended choice, the points in the *green area* where the controller is sure of the user's intended choice based on the threshold early completion method, and finally, the points in the *yellow area* where the controller is forced to make a decision by running out of time, leading to the specified error rate.

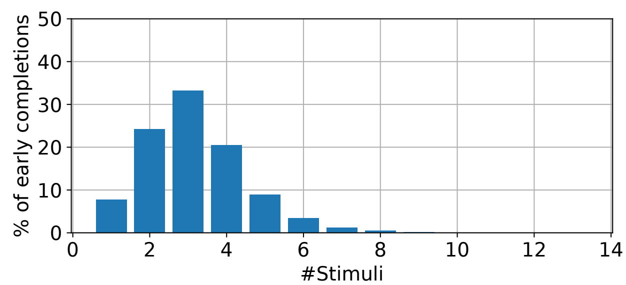


Fig. 6: Percentage of simulations early ending at each time step with threshold early completion and $C = 3$.

rate of 10^{-3}. However, if the user's intent can be identified earlier, based on the certainty threshold ϵ in the green lower-left area, the action can be taken sooner.

Fig. 6 shows the percentage of early completions after each stimulus for $C = 3$. For example, in about 8 % of the cases, the user's intent is known after the first stimuli. From the error rate analysis above, we know that seven stimuli are needed to guarantee an error rate of 10^{-3} when $C = 3$. However, using the method for early completion, the target choice is identified before seven stimuli in the majority of cases, and the user's cognitive load is thus significantly reduced.

D. Branch prediction and threshold early completion combined

Fig. 7 shows the percentage of early completions, similar to Fig. 6, but when both branch prediction and threshold

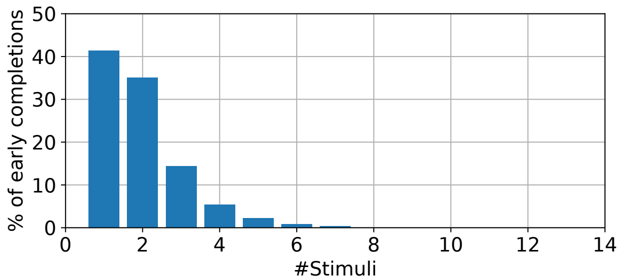


Fig. 7: Percentage of simulations early ending at each time step with threshold early completion, using branch prediction and $C = 3$.

early completion are used. As can be seen, in more than 40 % of cases, the user's intention is identified after the first stimuli. Compared to the seven stimuli required to guarantee the error rate of 10^{-3} when $C = 3$, the target choice is identified in the vast majority of cases, reducing the user's cognitive load even further.

E. Results in numbers

Table I summarizes the number of needed stimuli to reach the specified error rate when branch prediction is used versus not used and shows the advantage of using threshold early completion. From the table, it can be seen that using only branch prediction means that fewer stimuli are needed to guarantee the specified error rate. It can also be seen that using threshold early completion reduces the number of stimuli by 55-58 % when no branch prediction is used and 65-83 % when branch prediction is used. In either case, threshold early completion significantly reduces the user's cognitive load regardless of the number of choices C.

IV. Discussion

In a BCI-controlled wheelchair, various user experience factors must be taken into account. Among these is the level of stimuli presented, with fewer stimuli reducing cognitive load and enhancing user experience. This paper examines the necessary number of stimuli for wheelchair performance and techniques to maintain performance while reducing stimuli.

A. Ethical Considerations

When dealing with BCIs, addressing ethical issues like user privacy is crucial. The system discussed in this paper must store the feature map for determining user's indended targets, along with the parameters of target and non-target GMMs. Historical navigation choices may also be stored if branch prediction is employed. The feature map and GMM parameters could potentially identify individuals if EEG data is accessible, though no EEG data is retained in the system.

B. Error rate

The performance of the BCI wheelchair is measured by the error rate, as described previously. Fig. 3 can be used to determine the time it takes for the controller to understand the user's intent. For example, if the user gets three choices, $C = 3$, up to seven stimuli might be needed to guarantee the specified error rate. We choose to show two stimuli per second, thus the wheelchair should start showing stimuli at least 3.5 seconds before the decision is needed.

C. Branch prediction

Branch prediction is our method to reduce the required number of stimuli to reach a certain error rate, as seen in Fig. 4. Branch prediction works best when the user's usual behavior is known, i.e., an accurate estimate for $\boldsymbol{p}_{\text{prior}}$ is known, but an initial estimate can be made based on the room layout, for instance. If the wheelchair speed is 1 m/s, giving the user three choices roughly corresponds to start showing stimuli to the user 4 meters ahead. As seen in the first row of Table I, when using our suggested branch prediction, the number of required stimuli is reduced, decisions can be made 12 % to 17 % faster, and the wheelchair can potentially run faster.

D. Threshold early completion

Threshold early completion is our method to reduce the cognitive load of the user by reducing of the number of shown stimuli, thus improving the user experience. The certainty threshold, ϵ, is the limit for when the controller is sufficiently sure of the user's intended choice. Raising ϵ leads to making decisions based on less accurate probability estimates, meaning fewer stimuli are needed, but more errors will be made. A too-high ϵ means more actions will be incorrect, and the error rate increases. Lowering ϵ leads to fewer errors but requires more stimuli. In Fig. 5 increasing the certainty threshold means raising the dotted line, which would mean more incorrect actions. For the results presented in Fig. 6 and Table I $\epsilon = 10^{-3}$ was used.

Returning to the required stimuli to reach a specific error rate as discussed in the two previous sections, that corresponds to the cutoff for the yellow area. If the error rate was to be reduced, more stimuli would be needed to reach that error rate (see Figs. 3 and 4). For Fig. 5 that would mean that the yellow border is moved further to the right.

E. Branch prediction and threshold early completion combined

The threshold early completion gets more effective when combined with branch prediction as seen in Fig. 7. The controller can take action on the user's intent after the first stimuli in more than 40 % of the cases. This is attributed to branch prediction, which increases the likelihood of presenting the stimuli aligned with the user's intention as the first stimulus. If the feature map output y ends up in a non-overlapping area, the target choice is identified directly. In contrast, if a non-target stimulus is shown, the controller can be sure that this stimulus is not the target but doesn't know which of the other stimuli is the target and thus needs to show more stimuli before reaching a conclusion. Branch prediction increases the probability of showing a target stimulus first,

TABLE I: Numerical results. The number of available choices is C. The first row is the number of required stimuli to guarantee the specified error rate, gotten from the intersection between the error rate and the lines in Figs. 3 and 4. The second row is the number of required stimuli on average before an action, using threshold early completion. This is the weighted sum of the bars in Figs. 6 and 7. The last row is the percentual reduction of required stimuli in a practical setting when threshold early completion is used compared to when neither threshold early completion nor branch prediction is used.

	$C=2$		$C=3$		$C=4$		$C=5$		$C=6$	
Branch prediction	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes
#Stimuli required for desired error rate, non-threshold	5	4	7	6	9	7	11	9	13	11
Average #Stimuli using threshold early completion	2.1	1.7	3.2	2.0	4.0	2.1	4.7	2.2	5.5	2.2
% reduced stimuli vs. non-threshold no branch prediction	58	65	55	72	56	77	57	80	58	83

which, in turn, increases the chance of identifying the user's intended choice immediately.

F. Takeaways

This paper's key finding is that our techniques decrease the number of stimuli displayed, enhancing user experience. Table I numbers serve as motivation for method usability rather than precise improvement indicators. We demonstrate EEG data utilization for simulating BCI wheelchair performance and propose branch prediction and threshold early completion to cut stimuli. These methods constitute a significant contribution to enhancing user experience in BCI-controlled applications like wheelchairs.

V. Conclusion

This paper simulates the decision process for a brain-computer interface controlled wheelchair. We analyze how long time prior to the wheelchair's action, the decision-making process should be initialized to guarantee a predefined error rate for the actions of the wheelchair. We introduce branch prediction and threshold early completion as methods to reduce the time for the decision-making process and thus improve the user experience. Through our proposed methods, the time required for the user to make decisions can be reduced by half. Our approach is a step towards ready-to-use brain-computer interfaces with the potential of expanding the boundaries of BCI appliances and research.

References

[1] M. Congedo et al. “"Brain Invaders": a prototype of an open-source P300- based video game working with the OpenViBE platform”. In: *BCI 2011 - 5th International Brain-Computer Interface Conference*. Sept. 2011, pp. 280–283.

[2] A. P. Dempster, N. M. Laird, and D. B. Rubin. “Maximum Likelihood from Incomplete Data via the EM Algorithm”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 39.1 (1977), pp. 1–38. ISSN: 00359246.

[3] Chuong B. Do and Serafim Batzoglou. “What Is the Expectation Maximization Algorithm?” In: *Nature Biotechnology* 26.8 (Aug. 2008), pp. 897–899. DOI: 10.1038/nbt1406.

[4] Hugh F. Durrant-Whyte, David C. Rye, and Eduardo Mario Nebot. “Localization of Autonomous Guided Vehicles”. In: 1996. URL: https://api.semanticscholar.org/CorpusID:60535400 (visited on 11/03/2023).

[5] V. Jayaram and A. Barachant. “MOABB: trustworthy algorithm benchmarking for BCIs”. In: *Journal of Neural Engineering* 15.6 (Sept. 2018), article no. 066011.

[6] Landu Jiang et al. “SmartRolling: A Human–Machine Interface for Wheelchair Control Using EEG and Smart Sensing Techniques”. In: *Information Processing & Management* 60.3 (May 2023), p. 103262. DOI: 10.1016/j.ipm.2022.103262.

[7] Chang S. Nam, Anton Nijholt, and Fabien Lotte, eds. *Brain–Computer Interfaces Handbook: Technological and Theoretical Advances*. Collection of papers about BCI. New York: CRC Press, Jan. 2018. ISBN: 978-1-351-23195-4. DOI: 10.1201/9781351231954.

[8] Shu Kay Ng, Thriyambakam Krishnan, and Geoffrey J. McLachlan. “The EM Algorithm”. In: *Handbook of Computational Statistics: Concepts and Methods*. Ed. by James E. Gentle, Wolfgang Karl Härdle, and Yuichi Mori. Springer Handbooks of Computational Statistics. Berlin, Heidelberg: Springer, 2012, pp. 139–172. ISBN: 978-3-642-21551-3. DOI: 10.1007/978-3-642-21551-3_6.

[9] “*Pac-Man*”. In: *Wikipedia* (Sept. 2023). (Visited on 10/12/2023).

[10] Z. T. Al-qaysi et al. “A Review of Disability EEG Based Wheelchair Control System: Coherent Taxonomy, Open Challenges and Recommendations”. In: *Computer Methods and Programs in Biomedicine* 164 (Oct. 2018), pp. 221–237. DOI: 10.1016/j.cmpb.2018.06.012.

[11] Jingsheng Tang et al. “Towards BCI-actuated Smart Wheelchair System”. In: *BioMedical Engineering OnLine* 17 (Aug. 2018), p. 111. DOI: 10.1186/s12938-018-0545-x.

[12] Pex Tufvesson and Frida Heskebeck. “Automatic Control of Reactive Brain Computer Interfaces”. In: *IFAC Journal of Systems and Control* 27 (Mar. 2024), p. 100251. ISSN: 2468-6018. DOI: 10.1016/j.ifacsc.2024.100251.