Combining EEG and a Cognitive Model to Infer the Time Course of Game Play

Jon M. Fincham (fincham@cmu.edu)
Department of Psychology, 5000 Forbes Ave
Pittsburgh, PA 15213 USA

Caitlin S. Tenison (ctenison@ets.org)
Educational Testing Service
Princeton, NJ 08541 USA

John R Anderson (ja@cmu.edu)
Department of Psychology, 5000 Forbes Ave
Pittsburgh, PA 15213 USA

Abstract

We have developed an analysis stream for integrating a cognitive model with EEG data to reconstruct the cognition of individual subjects. A critical component of this method is the Sketch level that combines cognitive modeling and classification of EEG data using an HSMM to identify and place critical events over the timeline of a task. Multiple factors can influence sketch accuracy. In this study, we investigated the effect of game play elements on sketch accuracy across two EEG experiments where subjects interacted with the Space Fortress video game. Experiment 1 consisted of elaborate interface elements that accompanied game events (multiple sound effects, visual explosions). Subjects in Experiment 2 performed the same task, but audio and visual feedback elements were greatly reduced. We find that sketch accuracy while still much better than chance in Experiment 2, was significantly worse than in Experiment 1.

Keywords: EEG, cognitive modeling, cognitive reconstruction, HSMM, MVPA, Space Fortress, video game, BCI

Introduction

Considerable research has studied classifying electroencephalography (EEG) signals and the results have been applied to a number of domains such as brain-computer interfaces (BCI; Lotte et al., 2018), emotion recognition (Kim et al., 2013), understanding human memory (Noh et al., 2014), estimating workload (Brouwer et al., 2012), among others. Much of this research is conducted using a limited set of interaction paradigms (Abiri et al., 2019; Saeidi et al., 2021). In *active* BCI systems, classification methods are used to identify specific brain signals consciously and purposefully generated by the participant. *Reactive* BCI systems involves tasks where the experimenter has control over the presentation of stimuli and examines activity in predefined intervals, typically locked to the presentation of these stimuli. Research on *passive* BCI focuses on the classification of brain states that occur within complex, operational environments such as driving or aviation. Within passive BCI systems the sequence of events emerges as an interaction between the subject and the environment. These events can reflect a complex interplay between the cognitive process and task context and the uncertain timing of these events adds an additional challenge to their detection. Although this research is often conducted within realistic situations, the focus of the detection is often limited to considering only a few, highly distinguishable cognitive states(Aricò et al., 2016). The ability to decode diverse, time-variable events has valuable implications for enabling the development of neuroadaptive technologies to support complex tasks and greater interactivity (Krol et al., 2018)

Video games can provide a rich testbed that begin to bridge the gap between doing traditional EEG experiments in tightly controlled lab studies and the complex tasks in which people routinely engage every day. In recent research, Anderson, et al, (2020) decoded cognitive, perceptual, and motor events from EEG data gathered from participants playing the video game Space Fortress (Donchin, 1989; Frederiksen & White, 1989; Gopher et al., 1989). In that work, they presented a Sketch and Stitch method that was successful in reconstructing an entire sequence of actions to capture the play of a subject in a game. The Sketch component of that procedure was used to infer a chronology of the critical events of a subject's gameplay by using a hidden semi-Markov model (HSMM) to combine cognitive modeling and EEG data. The critical events they tried to identify were

1. **Kills:** when a player succeeds in destroying fortress;
2. **Deaths:** when a player's ship is destroyed;
3. **Resets:** when a player slips in trying to build up the vulnerability of the fortress and is reset to 0.

They exploited the fact that such events during gameplay tend to produce robust EEG signals while a cognitive model can provide probabilities of various transitions between critical events as well as the distribution of intervals between these events. The approach identifies the most probable sequence of critical events and when they happened.

While Anderson et al (2020) had success identifying critical events in a subject's game play, the Space Fortress interface accompanies these critical events by special visual and auditory effects, raising the question if this success just depended on detecting perceptual responses in the EEG. For example, the destruction of a ship was accompanied by a sound effect and an elaborate visual element meant to indicate an explosion. In this paper, we explore the question of how well the method would work in a situation where these events occurred without the strong perceptual correlates. We ran an experiment that replicated the one described in Anderson et al (2020) but reduced the audio and visual events that accompanied game play. Necessarily, something in the interface must change to indicate to the subject that the event has happened, but we eliminated strong visual and auditory signals. We will compare the results with this reduced interface to the prior results with the original Space Fortress interface.

Space Fortress Game

Figure 1 illustrates the critical elements of the game. Players are instructed to fly a ship between the two hexagons. They are firing missiles at a fortress in the middle, while trying to avoid being hit by shells fired by the fortress. The ship flies in a frictionless space. To navigate, the player must combine thrusts in various directions to achieve a path around the fortress. Mastering navigation in the Space Fortress environment is challenging; while subjects are overwhelmingly video game players, most have no experience in navigating in a frictionless environment. We use the Pygame implementation of Space Fortress (Destefano, 2010)where all actions are key presses.

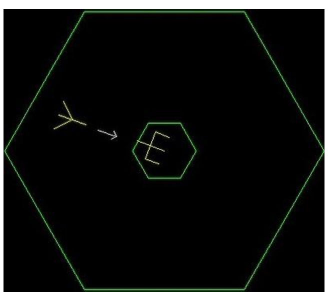

Figure 1: Snapshot of ship (nearest outer hexagon) shooting missile (arrow) at fortress (inside inner hexagon) .

We used the Autoturn version of the game introduced in Anderson et al. (2019) and described in detail in that paper. In this variant of the game, the ship is always aimed at the fortress and subjects do not have to turn it. There are only two relevant keys: A left-hand press of the W key to add thrust to the ship and a right-hand press of the space bar to fire at the fortress. The ship begins each game aimed at the fortress, at a 9:00 starting position (Figure 1), and flying at a moderate speed parallel to the upper left diagonal segment of the outer hexagon . To avoid having their ship destroyed, subjects must avoid hitting the inner or outer hexagons, and they must fly fast enough to prevent the fortress from aiming, firing at, and hitting the ship. When subjects are successful the ship goes around the fortress in a clockwise direction. They can destroy the fortress by shooting missiles at it to build up its vulnerability and then destroying it with a "kill shot" (two shots in rapid succession). If the fortress is destroyed, it leaves the screen for 1 second before respawning. If the ship is destroyed, it respawns after 1 second in the starting position flying along the starting vector. The replay site (http://andersonlab.net/reconstruction/) offers examples of game play.

Anderson et al. (2019) found that subjects can achieve relatively high and stable performance within an hour of playing AutoTurn (much faster than in original Space Fortress where subjects are also responsible for turning their ship among other things). To maintain a constant challenge of game play, a staircase procedure decreased the separation between the inner and outer hexagons as subjects got better. Subjects played 1-minute games. During the first 10 games the inner corners were 40 pixels from the center and the outer corners were 200 pixels from the center producing a width of 160 pixels. After the tenth game, the border width was reduced by 10 pixels if the subject had 0 or 1 deaths in the prior game and it was increased by 30 pixels (to a maximum width of 160 pixels) if they had 2 or more deaths. In this way the death rate in the game was maintained at about 1 death per 1-minute game. For each 10 pixels the border is reduced, subjects get an additional 10 points for each fortress they destroy. Navigation becomes more difficult as one has to fly between narrower borders, with many deaths resulting from thrusting into the inner hexagon, a rare event with the original 160-pixel width.

The Sketch procedure combines classification results from the EEG signal with information about the expected distribution of critical events from a cognitive model of the subject. The cognitive model we use was the ACT-R model that was described in Anderson (2019). We simulated 100 subjects by running the model 100 times on 60 games under the same game conditions as humans to generate behavioral results. We ran the model in over 35,000 games to generate statistics used in the Sketch procedure.

Methods

Here we describe data collection, pre-processing and procedures. We will refer to the reduced interface experiment as "Experiment 2" to contrast it with "Experiment 1" in Anderson, et al (2020).

Subjects

A total of 20 subjects (6 male, 14 female) were recruited from the CMU population of students and researchers between the ages of 18 and 40. None reported a history of neurological impairment. Subjects were paid between $60 and $75 for participation, depending on task performance. The duration of the experiment, including setup and task execution was less than 2 hours. All participants signed a written informed consent form. The experimental protocol was reviewed and approved by the Carnegie Mellon University Institutional Review Board.

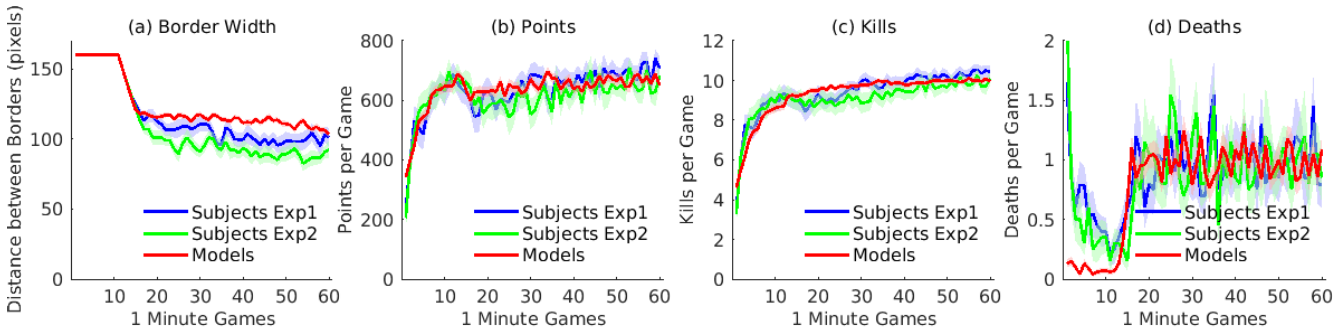

Figure 2. Mean values (line) and standard errors (area around lines) per game for subjects and models as a function of game (a) border width; (b) points before bonuses for kills at narrow borders; (c) number of fortress destructions; (d) number of deaths.

Task

Subjects were given a verbal overview of the time course of the experiment and how to play the game, after which they interacted with the software at their own pace. After reviewing instructions displayed onscreen, they played 60 1-minute games. Each 1-minute game yielded 1800 1/30 sec time frames or game ticks. The full game state is recorded by the software on every game tick. The record of game state included the keyboard (keys down/up) and all aspects of the display screen (direction, speed and location of the ship if alive, fortress orientation, presence of shells or missiles, etc.).

Three changes made from the game used in Anderson et al (2020). First, as already noted, we eliminated all explosions (visual and auditory effects). Second, in the original game one auditory tone accompanied each shot and another auditory tone accompanied a reset. This resulted in a quick double tone when there was a reset. In this version to eliminate the double tone, we used one tone when a shot resulted in an increment to vulnerability and another tone when there was a reset of vulnerability. Half of the subjects had one pairing of tones to the vulnerability changes while this was switched for the other half. Third, we changed the awarding of points. In the original game, as soon as the borders began to narrow (game 11) subjects received double the 100 points for a fortress kill. As described above, in this game they received an additional 10 points for each 10-pixel reduction of the border width. This change was introduced to keep subjects motivated to play at a higher level of difficulty.

EEG Analysis

The EEG was recorded from 128 Ag-AgCl sintered electrodes (10-20 system) using a Biosemi Active II System (Biosemi, Amsterdam, Netherlands). The EEG signal was recorded continuously for the entire experimental session and broken into 1-minute games. Portions of the game periods that included poor signal were excluded. Individual channels within an epoch were flagged based on having extreme values for mean absolute deviation, drift, or range. Flagged channels were interpolated. Epochs that still contained channels with extreme values after these steps were flagged and rejected. This resulted in loss of the signal for an average of 2.3 seconds per game for games used in the decoding (44.4% of the games had no lost signal).

In order to get simple correspondence with the game state data, the 512 Hz data were then down-sampled to 30 Hz with default EEGLab anti-aliasing filtering applied. A one-second window around each game tick (14 game ticks before, the game tick, and 15 game ticks after) was used to classify whether a game tick contained a critical event. Thus each game tick had associated with it a vector of 30*128=3840 electrode readings, representing regional effects, frequency effects below 30 Hz, and their interactions. The first 1000 components of the PCA of these vectors were used for classification.

Classification

We replicated the Sketch procedure described in Anderson, et al (2020). We focused our analysis on the last 55 games for each subject where performance is relatively stable while also employing the same game exclusion criteria used in experiment 1. Of the 1100 games, we excluded 10 games because of border width or relative inactivity by the subjects (the one and only game where the staircase procedure resulted in a border width of 30 pixels, 3 games where subjects failed to destroy a fortress without resetting or being killed, and 6 further games with 12 or fewer critical events) leaving 1090 games.

Classification was performed on the 1000-element vectors produced by the PCA to identify the critical events that determine the critical sketch of game activity. We used a leave-one-game-out method where for a given target game of a particular subject, linear discriminant classifier training was done using all remaining games for that subject and all games from the remainder of the subjects. The classifier was trained to label the EEG activity vectors with the critical event corresponding to the game tick the vector describes. To reflect the point that a subject's own data are likely the most relevant, the training games for each subject are weighted 15 times more than the games of other subjects. This leave-one-game-out procedure was repeated for every game to generate event probabilities across all 1090 games.

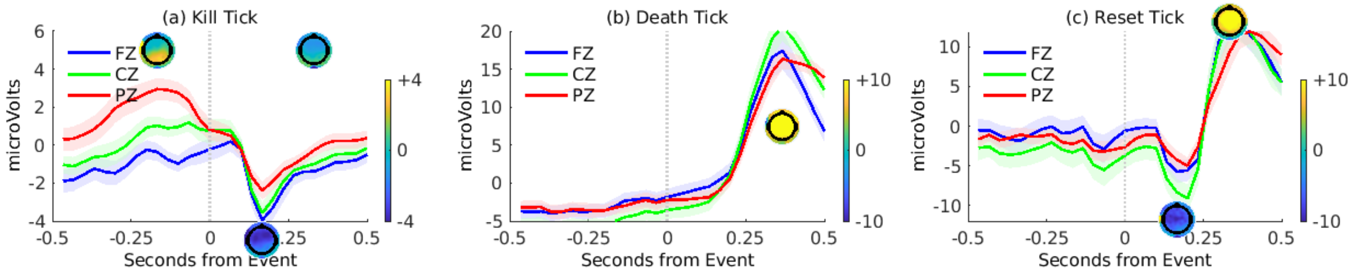

Figure 3. (a) EEG activity around the destruction of the fortress with the scalp profiles ranging from -4 to 4 μV. (b) EEG activity around the death of a ship with the scalp profiles ranging -10 to 10 μV. (c) EEG activity around a vulnerability reset with the scalp profiles ranging -10 to 10 μV. Shaded areas represent a standard error of the mean calculated from the standard deviation of the subject means.

Results

Behavioral Results

The time course of various performance measures over 60 games are shown in Figure 2. Data shown include those from Experiment 1 labeled as 'Subjects Exp1', the reduced-interface-element Experiment 2 described above labeled as 'Subjects Exp2', and the model data from 100 simulated subjects, labeled as Models. Games 1-10 all had a fixed border width of 160 pixels between the small inner hexagon that contains the fortress and the outer hexagon. After game 10, the staircase procedure was employed: border widths for successive games would continue to decrease at 10 pixel decrements until a subject's ship was destroyed 2 times or more, at which point the next game would reset to a larger width.

Part a shows border width. Subject behavior in both experiments results in slightly tighter border widths than those from model gameplay. Considering only games 11-60 where border width could vary according to the staircase procedure, Exp2 subjects attained somewhat tighter border widths (M = 98.2, SD = 13.46) than Exp1 subjects (M = 107.6, SD = 15.59), t(38)=2.04, p=.049 reflecting the change of scoring scheme from Experiment 1. Figure 1b shows canonical point scores by game. Canonical points show what subjects would achieve with the original 100 points per kill without the further bonuses they get for kills at narrow widths. Points were comparable for models and subjects over the course of the experiment, and there was a not a significant difference in points scored between Exp2 subjects (M = 627.1, SD = 122.08) and Exp1 (M = 655.5, SD = 106.76), t(38) = 0.782, p=.44. A similar pattern holds for fortress kills shown in Figure 1c, with roughly 9.5 kills per game in both Exp2 (M = 9.4, SD = 1.67) and Exp1 (M = 9.7, SD = 1.32). Similarly, there was no difference in ship deaths (Figure 1d) between Exp2 (M = 0.9, SD = 0.12) and Exp1 (M = 0.9, SD = 0.13), both averaging just under 1 death per game which was the goal of the staircase manipulation.

Generating a Sketch

While the above performance measures show that behavioral performance is comparable between the enhanced and reduced versions of the game, the essential question we want to answer is whether and how features of the gaming interface affect the ability of the Sketch procedure to accurately assign the identity and timing of critical events throughout a game. There are five critical events that occurred during gameplay:

1. Kills. Player destroys the fortress and scores 100+ points.
2. Fortress Respawns. 1 second after the fortress is killed, it reappears and normal gameplay can resume.
3. Deaths. The player's ship is destroyed and the player loses 100 points.
4. Ship Respawns. The ship is absent for 1 second after death, then reappears and normal gameplay can resume.
5. Resets. If the interval between ship missile firing is less than 250ms and the vulnerability is less than 11, the fortress vulnerability will be set back to zero and the subject must begin rebuilding the vulnerability from scratch.

Table 1: Interface Elements for Game Events

	Experiment 1		Experiment 2	
Event	Hear	See	Hear	See
Ship Death	Whoosh	Explode		
Fortress Kill	Whoosh	Explode		
Missile Fired	HF Beep	→		→
Fortress Fired	LF Beep	<>		<>
Vuln Increase			Beep 1/2	
Vuln Reset	Beep		Beep 2/2	

Table 1 shows the interface elements associated with various game events in both experiments. Experiment 2 has eliminated all unnecessary sounds and visual effects. Missile and shell firing are still accompanied by the visual display of the missile or shell flying across the space. Increments and decrements of vulnerability are indicated by distinctive tones so the subject does not have to be constantly looking at vulnerability on a different part of the screen. In addition, ship deaths and fortress deaths are accompanied by a 1-second removal of the fortress or ship so the subject does not waste actions.

While the classification component of the Sketch procedure is multivariate in nature, it is useful to have a sense of the mean EEG activity around events that will be classified. We show the activity around a subset of critical events in Figure 3. Each of the panels shows a full second of activity (the same time-window used in the classification procedure), from 500 ms before the event to 500 ms after.

There seems to be a post-event positivity that is common to kills, deaths, and resets in both experiments, though in the current experiment, kills show only a return to baseline from negativity as opposed to positivity. Consistent with results reported in Anderson, et al (2020), the magnitude of this positivity in both experiments varies with the rarity of the event. Kills are most frequent and show the smallest positivity while deaths are the least frequent event and show the greatest return to positivity. This is consistent with what would be expected from a P300 (Polich, 2012).

Classification Results

As in Anderson et al (2020), the leave-one-game-out cross validation procedure to predict labels for the 5 classes of critical events also requires inclusion of a sixth class containing null events. To avoid being overwhelmed by null events, for every critical game tick in a single game, 2 non-critical game ticks were chosen randomly to include in the classifier training phase. The overall discriminability d-prime was 1.76. Average accuracy was 54.0% and the average pairwise AUC was .915. This was slightly lower than Anderson et al. (2020) where d-prime was 2.0, average accuracy was 59.6% and the average pairwise AUC was .942.

As detailed in Anderson, et al (2020), the classification results themselves would not give us very good critical event sketches. For example, many of the null events are labeled as being critical events. Further, even if we managed to achieve unrealistically good classification accuracy, an unconstrained critical event sketch would contain sequences of events that are unlikely within the dynamics of the Space Fortress game. We need a way to tell the real critical events from the false labels and sequence events realistically. The Sketch method was developed for this purpose. This procedure combines statistics about what critical events are likely to occur when. This is calculated from a large library of model runs with output from the classifier to produce a critical sketch. The model games are used to estimate probabilities for a critical event transition matrix as well as latency distributions for time elapsed between events. The transition matrices and latency distributions are used to parameterize an HSMM.

The HSMM can efficiently combine the model-based statistics and conditional probabilities from the EEG classifier to estimate the most likely sequence of events in a game. Any sequence of events can be denoted $a_1, a_2, \ldots, a_n$ occurring at game ticks $t_1, t_2, \ldots, t_n$ where a_1 is game start (and so t_1 is game tick 1), a_n is the end (t_n is the 1800th game tick), and $a_2, \ldots, a_{n-1}$ are fortress kills and respawns, ship deaths and respawns, and vulnerability resets. Anderson, et al (2020) derived the following proportionality describing the probability of any such sequence relative to the probability of other sequences:

$$\mathrm{Pr}\,ob(a_1, a_2 \cdots a_n) \approx \prod_{i=1}^{n-1} trans(a_i, a_{i+1}) * f(t_{i+1} - t_i \mid a_i, a_{i+1}) * \frac{P(EEG(t_{i+1}) \mid a_{i+1})}{P(EEG(t_{i+1}) \mid Null)}$$

where $trans(a_i, a_{i+1})$ is the probability of transition between the events a_i and a_{i+1} estimated from the model runs, $f(t_{i+1} - t_i \mid a_i, a_{i+1})$ is the probability of the $t_{i+1} - t_i$ game ticks between the events a_i and a_{i+1}, instantiated with the distributions computed from the model runs, and $P(EEG(t_i+1, t_{i+1}) \mid a_{i+1})$ is the conditional probability of the EEG signal for this period if it ends in a_{i+1} where the conditional probabilities are generated from the classifier.

The Viterbi algorithm (Rabiner, 1989) for hidden semi-Markov models was used to find the assignment of events (event identity and timestamp) that maximized $Prob(a_1, a_2, \ldots, a_n)$. This produced for each game a critical event sketch: a set of inferred events and the time ticks when they occurred. We use two measures to evaluate the goodness of match between sketch and actual game events: recall and precision (Buckland & Gey, 1994). We focus only on kills, deaths and resets (ignoring respawns of ship and fortress as they were directly tied to kills and deaths with a 1 second lag). The recall measure considers all actual game events that occur and the identity of the closest sketch event to each. If the identity of the closest sketch event matched the actual game event, the assigned recall score would be the distance in time ticks between them. If the sketch and actual event time tick were identical, the score would be 0. If the sketch event was further than 2.5 seconds (75 time ticks) away, or if the identity of the sketch event did not match, a score of 75 was assigned. The precision measure used the same scoring procedure but was anchored to predicted sketch events and evaluated match to the closest game events.

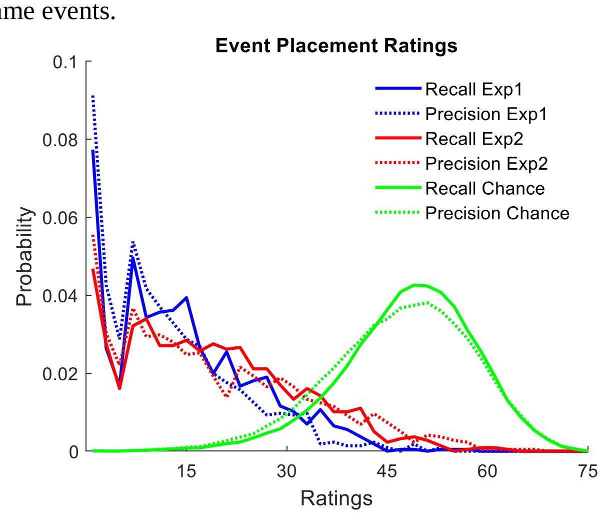

Figure 4: Event placement rating distributions for both experiments and chance performance.

Figure 4 shows the distribution of recall and precision scores for Experiments 1 and 2 and provides a comparison to chance (reconstructions randomly paired with games). The mean recall and precision was 14.1 and 11.8 for Experiment 1, 18.4 and 18.5 for Experiment 2, and 48.1 and 47.0 for chance. While the reconstructions for both experiments are far better than chance, the difference in recall is significant ($t(38)=2.25$, $p< .05$) as is the difference in precision ($t(38)=2.85$, $p <.01$).

Conclusion

A straightforward conclusion seems to emerge when comparing sketch results from the embellished Experiment 1 to relatively impoverished Experiment 2: While there remains enough information in the cognitive response to events to achieve a fairly high-quality sketch of the events in Experiment 2, the sketch accuracy is somewhat lower than in Experiment 1, reflecting the slightly poorer classification performance, likely a result of reduced game feedback elements. As Figure 2 shows, the current ACT-R model only approximately matches subject performance. A direction for improvement of reconstruction in either experiment would be a further improvement in that model.

Acknowledgments

This material is based upon work supported by the Office of Naval Research Grant N00014-15-1-2151 and USAF AFRL under Contract No. FA8649-20-C-0008. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the ONR or USAF AFRL.

References

Abiri, R., Borhani, S., Sellers, E. W., Jiang, Y., & Zhao, X. (2019). A comprehensive review of EEG-based brain–computer interface paradigms. *Journal of Neural Engineering*, *16*(1), 11001.

Anderson, J. R., Betts, S., Bothell, D., Hope, R., & Lebiere, C. (2019). Learning Rapid and Precise Skills. *Psychological Review*, *126*, 727–760.

Anderson, J. R., Betts, S., Fincham, J. M., Hope, R., & Walsh, M. W. (2020). Reconstructing fine-grained cognition from brain activity. *NeuroImage*, *221*. https://doi.org/10.1016/j.neuroimage.2020.116999

Aricò, P., Borghini, G., di Flumeri, G., Colosimo, A., Bonelli, S., Golfetti, A., Pozzi, S., Imbert, J.-P., Granger, G., Benhacene, R., & others. (2016). Adaptive automation triggered by EEG-based mental workload index: a passive brain-computer interface application in realistic air traffic control environment. *Frontiers in Human Neuroscience*, *10*, 539.

Brouwer, A. M., Hogervorst, M. A., van Erp, J. B., Heffelaar, T., Zimmerman, P. H., & Oostenveld, R. (2012). Estimating workload using EEG spectral power and ERPs in the n-back task. *Journal of Neural Engineering*, *9*(4), 45008.

Buckland, M., & Gey, F. (1994). The relationship between recall and precision. *Journal of the American Society for Information Science*, *45*(1), 12–19.

Delorme, A., & Makeig, S. (2004). EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods*, *134*(1), 9–21.

Destefano, M. (2010). *The mechanics of multitasking: The choreography of perception, action, and cognition over 7.05 orders of magnitude*. Rensselaer Polytechnic Institute.

Donchin, E. (1989). The learning-strategies project: Introductory remarks. *Acta Psychologica*, *71*(1–3), 1–15.

Frederiksen, J. R., & White, B. Y. (1989). An approach to training based upon principled task decomposition. *Acta Psychologica*, *71*(1–3), 89–146.

Gopher, D., Weil, M., & Siegel, D. (1989). Practice under changing priorities: An approach to training of complex skills. *Acta Psychologica*, *71*, 147–178.

Kim, M. K., Kim, M., Oh, E., & Kim, S. P. (2013). A review on the computational methods for emotional state estimation from the human EEG. *Computational and Mathematical Methods in Medicine*.

Krol, L. R., Andreessen, L. M., & Zander, T. O. (2018). Passive brain–computer interfaces: a perspective on increased interactivity. In *Brain–Computer Interfaces Handbook* (pp. 69–86). CRC press.

Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., & Yger, F. (2018). A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update. *Journal of Neural Engineering*, *15*(3), 31005.

Noh, E., Herzmann, G., Curran, T., & Sa, V. R. (2014). Using single-trial EEG to predict and analyze subsequent memory. *NeuroImage*, *84*, 712–723.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257–286.

Saeidi, M., Karwowski, W., Farahani, F. v, Fiok, K., Taiar, R., Hancock, P. A., & Al-Juaid, A. (2021). Neural Decoding of EEG Signals with Machine Learning: A Systematic Review. *Brain Sciences*, *11*(11), 1525.