'Something's Fishy': Application of Performance-Accommodation Mechanisms in BCI-like Games to Reduce Frustration and Increase Perceived Control

Ingeborg Goll Rossau*
irossa14@student.aau.dk

Jędrzej Jacek Czapla*
jczapl15@student.aau.dk

Rasmus Bugge Skammelsen*
rskamm15@aau.student.dk

ABSTRACT

This paper identifies five different Performance-Accommodation Mechanisms (PAMs) in established literature and systems that modified the input-output mapping of the user, as well as attempts to create a unified language (models and vocabulary) to describe them. For this paper, we implemented three of them: (1) sham input, (2), assisted success, and (3) assisted failure, all being 'explanatory narative'-explicitly visualised, in a low recall fishing game, to examine the effect of them on frustration and perceived control. Results of our experiment indicate that how people attribute the causes of the feedback has a major impact on these, as people rated sham input (a non-playable character taking over) lower and rated assisted success (the user representation getting stronger) higher. The results also contradict the results from our previous study [38], indicating that the effect between hidden and explicit PAMs is stark.

KEYWORDS

Brain-Computer Interface, Frustration, Perceived Control, Game Balancing, Skill-Accommodation, Performance-Accommodation

ACM Reference Format:
Ingeborg Goll Rossau, Jędrzej Jacek Czapla, and Rasmus Bugge Skammelsen. 'Something's Fishy': Application of Performance-Accommodation Mechanisms in BCI-like Games to Reduce Frustration and Increase Perceived Control.

1 INTRODUCTION

17 million people worldwide suffer a stroke every year, and many more are affected by it. As treatments improve, more and more people get to live through a stroke [43], resulting in more than 1% of the world's population currently living with the after-effects [11], especially paresis of the arms, affecting approximately 75.5% of stroke survivors [35]. The frequency of cerebrovascular incidents is predicted to increase over the next 15 years [43], and with increasingly better treatment, the amount of people living post-stroke will naturally also increase. Therefore, rehabilitation focused on regaining arm and hand movement is especially important.

Current state of the art stroke rehabilitation uses motor imagery, a supplementary rehabilitation technique [22], in combination with brain-computer interfaces (BCIs), to take advantage of the brain's neuroplasticity and reshape functions from damaged brain areas to other, more healthy ones [11, 33]. This technique has been found effective and often improves motor function [10], but BCI systems are not reliable, and it is discouraging and frustrating to perform lengthy rehabilitation sessions in a low recall system [45, 50].

Our previous study showed a significant difference in ratings of frustration and perceived control when hidden sham input (i.e. replacing true or false negatives with positives to trigger successful feedback) was added [38], making it a suitable addition to motor imagery rehabilitation. Despite this, we realised that sham input might not be the optimal way to get this effect, as sham input essentially takes away the user's agency, by replacing their input with one from the system. This prompted us to research how mechanisms for helping users struggling with poor performance (either caused by the system or by themselves) have been used in other systems; what we refer to as Performance-Accommodation Mechanisms (PAMs).

This paper provides a literature review resulting in a classification of PAMs based on a model of input-output mapping. Furthermore, our literature review uncovers a number of characteristics to describe specific implementations of the different PAMs. Three of these PAMs were implemented into a BCI-like system, to measure their effect on frustration and perceived control. The paper offers several contributions to the field: (1) an initial model describing the relationship between user input and system output, (2) various PAMs and their characteristics, and (3) several implications of how using and implementing these PAMs can have an effect on the user experience of an unreliable system.

2 BACKGROUND RESEARCH

In this chapter, we give a brief overview of stroke and brain-computer interfaces (BCIs), introduce implications of changing a user's input (or lack thereof) and introduce the concept of Performance-Accommodation Mechanisms (PAMs).

2.1 Stroke and Brain-Computer Interfaces

As the prevalence of stroke continues to rise, so fortunately does the survival rate [43]. Due to this, more than 1% of the world's population is currently living with the after-effects of stroke [11], including paresis (weakness or partial paralysis of muscles, which affects over 80% of all stroke survivors [35]. However, by participating in the proper rehabilitation practices, patients can regain some control due to the brain's neuroplasticity, which makes it possible for the brain to reshape its cortical networks to make other parts of the brain take over the responsibilities of the areas that have been damaged [11]. Typical rehabilitation includes practices such as muscle strengthening exercises and mirror therapy, depending on the needs of the individual patient. However, traditional methods are not always sufficient, which has led to the development of complementary methods such as the use of BCIs [33].

For post-stroke rehabilitation of limbs, BCIs are operated using motor imagery (MI) signals. When performing an MI task, the patient imagines making a movement (e.g. making a fist) without physically doing it, which creates a similar pattern of brain waves as the 'regular' movement [11, 15]. The signal is then collected by the BCI and used to create some kind of feedback, which is often in the form of a hand orthosis that closes the patient's hand in order to reestablish the connection from the muscles to the brain. While this can be beneficial for rehabilitation, using an MI BCI is not necessarily straight-forward, as 15-50% of users struggle with producing the correct brain signal due to the lack of any inherent feedback from the interaction itself [2, 41]. Furthermore, even if they produce the correct signal, the limited recall of BCI systems (the average being around 80%) means they will not necessarily get the feedback associated with a successful interaction [36, 50]. This can lead to frustration with the system [45, 50],

*The Technical Faculty of IT and Design
Rendsburggade 14, Aalborg, Denmark

which can in turn lead to decreased motivation and thus affect the quality of the rehabilitation [4, 6, 48].

Another potential cause of decreased motivation is the monotony of the MI BCI tasks, which usually involve doing the same thing repeatedly during a session. To distract from this monotony, some BCIs have turned to gamification, which has been shown to increase user enjoyment and satisfaction [1, 6, 48]. However, while gamifying the task might make it less boring, it does not change the instability of the system, where both false negatives and false positives can cause poor performance in-game.

2.1.1 BCI interaction paradigms. In our previous study, we conducted a review of literature on BCIs. Based on this, we were able to identify six different phases that are often present in the interaction paradigms of such systems. These are: (1) a *preparation* phase, where the system captures baseline measurements from the user to remove noise; (2) a *cue* phase, where the user is informed about their upcoming task; (3) a *task* phase, in which the user performs the assigned task (e.g. MI); (4) a *signal*, used to indicate the start (or end) of the task phase; (5) a *feedback* phase, where feedback is given for the task; and (6) an inter-trial-interval, in which the user gets a break in-between trials. [38]

An illustration of a 'typical' interaction paradigm, where all of these phases are included, can be seen in Figure 1. Note that not all BCIs include all of the different phases, and some phases (often the task and feedback phases) may overlap.

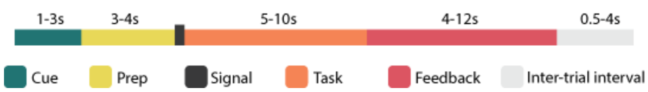


Figure 1: An example of a hypothetical BCI interaction paradigm which includes all six phases identified in [38]. The times listed in above the paradigm are the time-ranges found for all the different phases.

2.2 Control, agency and causality

Users being frustrated with using a low-recall rate system can potentially be explained by their inability to control it. This will especially be the case when the system's reward and punishment system is relying on a precise input, such as is often the case in games. Therefore, the level of control (true or perceived) is arguably especially important in a gamified BCI.

The sense of being in control is often referred to as the sense of agency. Agency is often talked about in two distinct categories: the feeling of agency (an inherent feeling of control, often through sensorimotor stimuli, that does not need further consideration) and judgement of agency (determining control of an unexpected outcome upon reflection on the context) [28]. Increased judgement of agency causes an ownership effect, making the user more positive about an object, in turn leading to higher motivation and engagement [8]. Vlek et al. [47] therefore argue that judgement of agency is important in BCIs.

Judgement of agency is tied to causality, as people register agency based on causal attribution. Wegner and Wheatley [49] point to three conditions necessary for people to perceive a causal relationship. According to them, the cause must: occur before the effect, be consistent with the effect, and be the only cause. A different theory, attribution theory, points to whether the cause was internal (the person themselves) or external (an outside agent), consistent with the effect, controllable, and whether the cause is global (i.e. whether it applies to all aspects of the person's lives or is specific to the circumstances) [16].

Sense of agency in an interactive system will increase if the causal chain is easily perceivable, as it makes it easier to connect the user's actions to effects [26]. Seinfeld et al. [39] suggest that the transparency of this causal chain can be increased by using explicit user representations. These are virtual extensions of the users' bodies and are mapped to their motor actions. This can be a cursor, an avatar, a virtual hand (as has been implemented in several BCIs [12, 29, 47]), etc. However, this is only the case when the predicted and actual effect is congruent [13].

Agency is fickle, and many things can be controlled to manipulate it to some extent. According to Evans et al. [18], visual feedback is very important in BCIs, as "*in the absence of strong internal motor cues, external visual feedback dominates*"; they showed that congruency between predicted and actual sensory feedback are associated with a robust system and that the opposite also holds, and that this effect was stronger with visual feedback. Lynn et al. [27] showed that participants using a BCI reported more intentions to move a line when it moved frequently, and Fard and Grosse-Wentrup [19] showed that users felt more in control when they completed a task. This indicates that agency can be increased in BCI systems by assisting users in achieving the set goals.

2.3 Performance-Accommodation Mechanisms

In a previous study, we introduced the idea of 'sham feedback' into a game created for BCI rehabilitation, as a method of accommodating for the poor performance of the system. While the system had an artificially induced recall rate of 50%, we introduced varying levels (0%, 15% and 30%) of 'false positives', i.e. where positive feedback was not a direct result of the user's action, but rather the system creating a positive outcome without reading any input from the user. Results of an experiment showed that with sham, users' frustration levels fell, while their levels of perceived control over the system rose. [38]

The idea of helping players struggling with poor performance is not exclusive to BCIs. It is used widely in video games, although within this context the accommodation is generally due to poor performance from the players rather than the system. This concept has been called various things, such as 'game balancing' [37], 'skill assistance' [17], 'difficulty adjustment' [3, 5], etc. It is commonly used in multiplayer games to balance the skills of opposing players to provide an equal challenge [5, 9, 17, 21, 46]. The idea is highly related to Csikszentmihalyi's concept of flow, which states that the optimal experience is when there is balance between the level of challenge and the user's level of skill [14].

Bateman et al. [7] used the term 'skill-accommodation mechanisms' to refer to this concept. Because we want to accommodate not for the players' level of skill, but rather for the system's performance, we have chosen to refer to them as Performance-Accommodation Mechanisms (PAMs). Specifically, we define a PAM as:

A mechanism within a game that seeks to increase the player's enjoyment of the game by accommodating for poor performance.

Throughout this project, we have identified a number of different overall PAMs based on how they differ in their mapping of input to output. These are described in Section 3.

3 LITERATURE REVIEW

Based on our previous research, we knew that helping players by turning failures into successes, in a game with a brain computer interface (BCI)-like interaction paradigm and recall rate, had the potential to decrease frustration while increasing perceived control. With this knowledge, we wanted to explore what other options might exist in terms of providing help, and how these could affect frustration and perceived control.

In our previous project, with sham feedback, we essentially replaced any input from the player with a system-created input [38]. While we saw an increase in perceived control in our previous project, players also only played our game for a short amount of time, and very few participants in our experiments had any suspicion that they might be receiving assistance. It is not unreasonable to assume that, with extended playtime, players might realise by themselves that their input is not always what causes successful feedback, which we theorised could then cause a negative effect on perceived control.

Therefore we decided to conduct a literature review to identify what other performance-accommodation mechanisms (PAMs) might exists, and how they impact the relationship between input and output.

3.1 Input-Output Mapping

The way that we implemented sham in our previous project was by replacing what would have been a 'failure' input (or a lack of input) with a 'success' input [38]. In BCIs, you can generally think of inputs in terms of 'failure' and 'success', which are separated by a specific threshold the user will need to get their signal above. This can be a binary case, where the signal is either accepted or not and the feedback provided is either a failure or a success, such as when the task is to get a hand-orthosis to move. In other BCI systems, the signal is read continuously, and the feedback will depend on the strength of the signal. An example of this is the game created by Müller-Putz et al. [31], in which the vertical position of a ball would depend on the strength of the BCI signal, assuming it passed a specific threshold. In this continuous case, the question of failure or success in terms of the task of getting the ball to move at all still depends on whether the signal crosses a threshold, just as for the binary case.

The same categorisation of inputs and outputs in terms of failure and success can be applied to certain tasks in non-BCI video games, namely the types of tasks where a PAM would be applied. For example, in a runner game, the binary task of jumping over an obstacle will result in either success of failure depending on whether the player pressed the 'jump'-button (i.e. crossed the threshold). In a racing game, the output for the continuous task of steering the vehicle through the track can be split into failure and success depending on whether the user is managing to keep the vehicle on the correct path. Therefore, we can think of both the input- and output-space for both of these types of tasks in terms of failure and success, with only one option for each in a binary tasks, while continuous tasks will have a range of successes and a range of failures.

As we will expand upon in the following sections, the way we differentiate between different PAMs is in terms of how they affect the mapping between input and output. In order to be able to explain and visualise this, we draw inspiration from Mary Shaw's 1986 model of input and output in interactive systems [40]. In Shaw's model, input is a signal sent from the external world (the input device) to the program, where it is converted into something that the program can read. The program then sends an output signal, which is sent back to the external world after being converted into something that can be read by the output device, as feedback for the user. Both the input and output signal go through what Shaw calls the 'I/O state', which contains information about the state of the program and the history of inputs and outputs and can therefore affect how both input and output are interpreted.

With this model in mind, we created the models seen in Figures 2, which show the unmodified (i.e. no PAM applied) mapping of a binary 'success' input to a binary 'success' output (Figure 2a), and the unmodified mapping of a continuous 'failure' input to a continuous 'failure' output (Figure 2b).

3.2 State of the Art of Performance-Accommodation Mechanisms

To find relevant articles, we searched mainly in the databases from ACM, IEEE Xplore and Science Direct, using keywords such as 'game balancing', 'difficulty adjustment', 'assistence' and 'video games'. We added more keywords as we found relevant phrases. Based on our database search as well as source-chaining, we ended up with 14 articles which described 29 different cases of PAMs (as several articles described two or more). This chapter describes the results of our literature search, through which we identified five different types of PAM based on their input-output mapping. Additionally, we also identified a number of characteristics which can apply to all the different types of PAMs, and which we believe could also have an impact on player frustration and perceived control.

3.2.1 Sham input. As described previously, when sham is employed, the system replaces user-generated failure inputs (which can be both true or false negatives) with a completely system-generated success input. Therefore, the user has no control in the moment when it is employed. Since the thing that is replaced is actually the input, we decided to rename it from 'sham feedback' to 'sham input'. Sham input is illustrated in Figure 3.

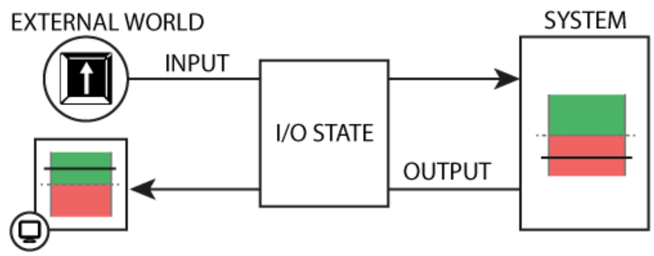
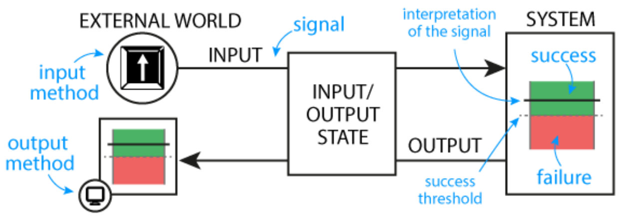


Figure 3: The mapping from input to output when sham input is applied. The input is interpreted by the system as a failure, and is then replaced by a success, which is what the user will get as feedback.

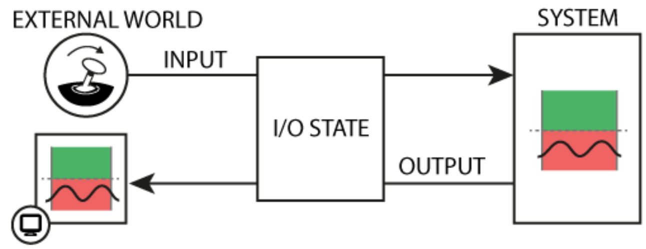
When sham input is used, the outcome is identical to the outcome of the input it replicates. However, the feedback can be different. By this we mean that the end result will be the same, even if how you get to that end result is not the same as you usually would. For example, Vicencio-Moreira et al. [46] implemented sham input for a targeting task, where the system would instantly lock onto the nearest target at the press of a button. The outcome of this is the same as for normal targeting - placing the crosshairs over the target - but the movement of the crosshair is (most likely, depending on the user's skill level) both faster and more even than when performed by the user. Note that for a continuous input such as in a targeting task, sham input will essentially replace the usually continuous output with a binary signal, as it will reduce the range of success outputs to a single 'correct' success output (i.e. moving the crosshairs in a direct line to the target). The same is true in Van Huysduynen et al.'s self-driving car simulation, where they replaced true negative inputs with sham input, reducing the possible options of 'correct' positions of the car on the road to a single 'correct' position [24].

Aside from our own previous work, this type of PAM has also been used in other evaluations of BCIs. As the ground truth of an interaction is non-existent in a BCI system, the system is unable to determine whether an input is a false or true negative; similarly it is also unable to distinguish between the user attempting or not attempting to create the correct input. For this reason, several papers have ignored all user-generated input entirely, replacing all inputs with sham inputs [34, 50].

3.2.2 Shared control. The analogy to driving automation extends also to shared control. The Society of Automotive Engineers established the six Levels of Automation (LOA), describing different levels of human intervention and attentiveness required in the driving task, ranging from fully manual (level 0) to fully automatic (level 5, essentially sham input) [24]. Between these levels are various levels of split ownership between the driver and the system. Traditionally, this split ownership has been explained using the horse-metaphor (H-metaphor): a rider can use loose rein and provide fine-tuning, otherwise letting their horse move more autonomously, and then using tight rein to gain more control. Just like LOA, Flemisch et al. [20]



(a) The mapping from input to output for a binary input (e.g. a key-press), which is interpreted by the system as a successful input. The output that is sent to the output device (e.g. a monitor) is also a success, resulting in successful feedback.



(b) The mapping from input to output for a continuous input (e.g. moving an analogue stick), interpreted by the system as a failure input. The output sent to the output device is also failure, resulting in failure feedback.

Figure 2

suggests that the concept of "*[t]ight rein and loose rein may be the extremes of a continuum rather than two exclusive states of operation.*"

There seem to be three kinds of shared control in the literature: blending, distinct loci of manipulation, and haptic shared control. Both Sykownik et al. [44] and Mulder et al. [30] describe blending, where two or more inputs are mixed before reaching the controlled system itself. Sykownik et al. further describes a system where two or more inputs control distinct loci of manipulation, such as one input controlling the steering of a car, while another controls the speed. Mulder et al. additionally describes haptic shared control which influences the physical interaction with the control interface; it is similar to blending, but closer to how the H-metaphor describes shared control.

The most common type seems to be blending, often used in driving video games, and for targeting in shooting games. Bateman et al. [7] examined the effects of various target assistance on player performance. One of the target assistance schemes was target gravity, where the system pulls the reticle towards a target that the user is simultaneously moving towards. Rakita et al. [32] used a shared control robotic arm to subtly improve the user's movements over time, showing and teaching them the movements to ideally perform a task. They compared the users' movements to a pre-defined ideal curve and created an average movement, similar to the mechanism implemented by Bateman et al. While target gravity in their experiment was deemed inferior to other target assistance schemes because it interferes with the users' movement, creating a conflict, it was seen that Rakita et al.'s robot arm did improve learning and deemed more trustworthy, which might indicate the effect of shared control in a novel system.

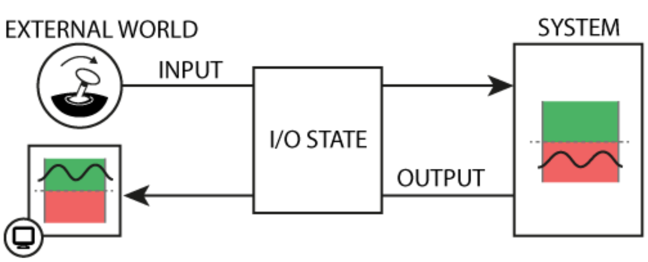


Figure 4: Mapping of input to output when shared control is applied. A continuous output within the 'failure' range is moved to the 'success' range, without reducing it to a single 'correct' output such as in sham input.

What all of these cases have in common is that they make use of continuous input and provide continuous output. In fact, shared control cannot exist for a binary input/output system, as there is no 'room' for a blended input when there are only two options. Moreover, while shared control can technically exist at all times, the user needs to perform sub-optimally, i.e. their input needs to fall within the range of 'failure', for the shared control to have an effect. If the user performed the movements perfectly in Rakita et al.'s experiment, then nothing would change from the input to the output. This is modelled in Figure 4, where we can see that shared control 'moves' a continuous failure input to a success input. In contrast to sham input, the output when shared control is applied is still influenced by the user's input and thus is still continuous, within the range of success outputs.

3.2.3 Assisted success. Sham input and shared control both aid the user in creating an outcome that it would theoretically have been possible for them to attain by themselves, either by completely replacing the user's input or by 'steering' it towards a better one. In contrast to this, assisted success creates an outcome that the user could not have achieved otherwise, specifically when they make a successful input. In this way, it can be said that it modifies - or expands - the output space.

This is a common technique in video games, as it describes power ups, such as speed boosts in driving games or damage modifiers in shooting games. For example, Depping et al. [17] implemented a damage modifier in a multiplayer shooting game, where the damage given and taken was depending on the score of the assisted player; the bigger the score difference, the more damage the 'worse' player would deal. Using the Player Experience of Needs Satisfaction (PENS) scale and Intrinsic Motivation Inventory (IMI), it was seen that the PAM increased perceived competence, enjoyment, suspense, and more. Another example was implemented by Rogers et al. [37], who explored how various PAMs affect the enjoyment and experience of a multiplayer AR table football game. The assisted success they implemented changed how many points a goal would grant depending on which half of the goal it would hit, or which half of the field the ball was shot from.

As is shown in these examples, assisted success takes a success input and 'amplifies' or 'enhances' it, making a successful input from the user have a bigger impact than normal. This is shown in Figure 5, with an example of a binary success input that results in a success output that is 'above' the normal success output.

3.2.4 Assisted failure. Similarly to assisted success, assisted failure results in an output that would otherwise not be attainable without the PAM. Where assisted success is activated in instances of a successful input, assisted failure is activated upon a failure input. The output in this PAM will be something in-between a failure and a success, where a failure is not punished as harshly as it would be otherwise, but also is not replaced with a success such as in sham input, or 'moved' towards it, such as in shared control.

As an example, Rogers et al. [37] also implemented assisted failure in their AR table football game. With this PAM activated, the unassisted players

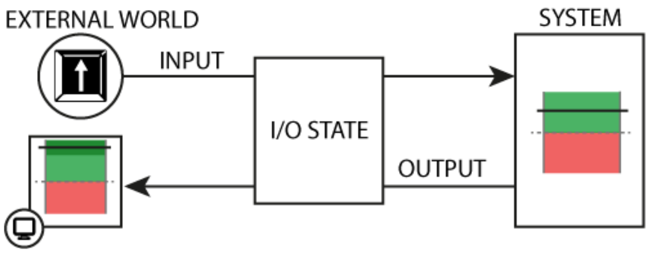


Figure 5: The input-output mapping for assisted success. An input is interpreted as a success, and is turned into an 'amplified' output.

would score a goal on the assisted players, but would not get any points, essentially nullifying the assisted players' failure to stop the ball, but without rewarding them by e.g. giving them points. Another example was implemented by Baldwin et al. [5], by giving the low-performing player in a shooting game a shield. When they then failed to not get hit, their failure was not punished by them losing health or dying. Similarly, Depping et al. [17] implemented that the assisted player would take less damage when hit, making their failure less punishing than usual.

From these examples, we can see that when the output can be described as binary (e.g. stopping the ball from hitting the goal vs. not stopping it), the failure is more or less nullified (e.g. the ball was not stopped, but it had no negative consequences). When the output is continuous (e.g. how much damage you take when getting hit), it *can* nullify your failure (e.g. no damage taken because you had a shield), or it can simply reduce its impact (e.g. you take less damage). Either way, the output will fall in-between normal success and failure, as seen in the model in Figure 6, modifying the output space just as assisted success.

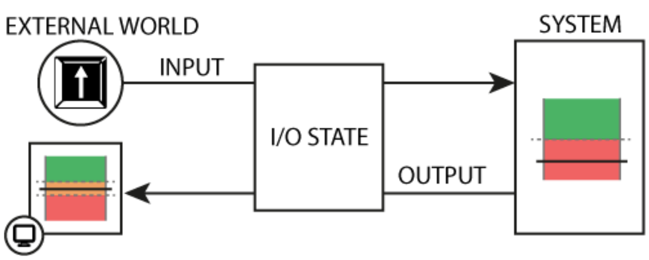


Figure 6: Mapping input to output when assisted failure is applied, here to a binary input. The input is interpreted as a failure, but the output space is modified to include a space in-between failure and success.

3.2.5 Rule change. The last PAM identified through the literature review also requires a sub-optimal input, but translates it into a successful output. In this sense, the input-output mapping is similar to sham input, but where sham input will 'convert' a failure into a success, this PAM instead changes the rules so that what would previously have been a failure is now a success, by changing the threshold for what can be considered a success. It therefore still relies on the user's input rather than just replacing this with a system input.

This PAM is seen in several contexts: Vicencio-Moreira et al. [46] implemented two aim assistance mechanisms relying on the user missing the target (a sub-optimal input), but changing the rules of the game so it translates to a hit (a successful output). One is called 'area cursor', which changes how big the reticle is (though not visually), and the other recalculates the bullet's travel vector to hit the target, called bullet magnetism. The user's initial input is still what is used to create the output, but it no longer needs to be as close to the target in order to result in a hit. Gerling et al. [21] also implemented a rule change in multiplayer rhythm game, where the timing for creating the perfect input was extended for the unassisted player, making an otherwise sub-optimal input provide successful feedback.

It should be noted that while for a continuous task, lowering the threshold is simply a matter of changing a range of what would otherwise be considered failures into successes, such as in the targeting assistance techniques described above. This is not really an option for a binary task, unless the task has a timing aspect, such as in Gerling et al.'s dancing game. A model of this PAM can be seen in Figure 7, where an input is (under normal circumstances) interpreted as a failure, but the output is a success due to the lowered threshold for succeeding.

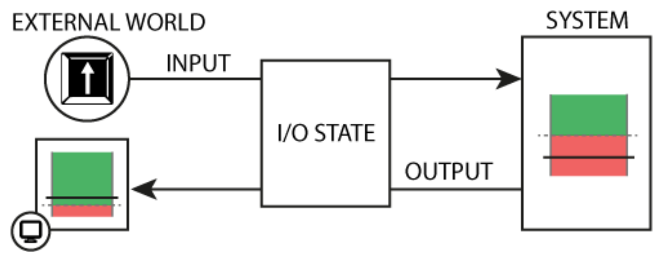


Figure 7: Input-output mapping when the rule change PAM is applied. An otherwise unsuccessful input results in a successful output because of the lowered threshold.

3.3 Characteristics of Performance-Accommodation Mechanisms

While the distinct PAMs can be distinguished between by their input-output relationship, there are several characteristics that can be applied to all of them to create more unique and context-specific variations. The characteristics we have identified concern themselves primarily with autonomy and awareness.

3.3.1 User- and system-employment. Employment refers to the activation of PAMs, and we can differentiate between user-employed PAMs and system-employed PAMs. Van Huysduynen et al. [24] implemented a system with their driving simulator, where the user activated the auto-pilot (requiring the user to only monitor the system) with a button press. In Rogers et al.'s [37] table football game, they differentiated between different types of user-triggered PAM, where it was either triggered simply with the press of a button, or by applying some level of skill, where the user had to hit an icon with the ball to activate the PAM, implying that some level of granularity needs to be applied to these definitions.

System-employed PAMs are activated by the system itself, meaning that the user has no control of whether PAM is happening or not. Vicencio-Moreira et al. [46] tested various system-controlled aim-assistance techniques to determine their performance and user-approval. All techniques were employed by the system although they were activated at different times, depending on the type of PAM. For example, area cursor (rule change) is always present, as it just increases the size of the reticle, while target gravity (shared control) pulls the reticle when moving towards a target, only impacting the output when the user steers off course. Regardless, the PAM is always activated at the discretion of the system.

Few papers looked at the differences between user- and system-employed PAM, but both Rogers et al. [37] and Smeddinck et al. [42] ran experiments with conditions that made it possible to compare them. Rogers et al.

used their AR table football game and measured the difference between (1) no employment, (2) system-employed based on various metrics, (3) user-triggered using a button, and (4) user-triggered with the skill-based method described above. Similarly, Smeddinck et al. looked into how to present difficulty-choices in a casual platformer. They looked into how users perceived menu-based, game-embedded, and automatic difficulty-adjustments. While papers showed that users preferred to use user-employed PAMs, Rogers et al. showed that system-employed PAM was still relatively highly approved, and Smeddinck et al. showed that the presence of PAM did not notably impact the game experience. It is not explicitly stated if users were informed about the presence of PAM in Smeddinck et al., as this was omitted from the procedure description, but users were told about it in Rogers et al.'s study. This might have an impact on users' perceived control, as discussed in section 3.3.3.

3.3.2 Adaptive and static accommodation. We defined employment as what determined the overall presence of PAMs in a system, but as indicated by Vicencio-Moreira et al. [46], it is also worth exploring the circumstances behind activating the PAM. Looking at systems with PAMs implemented, this has been split into two distinct categories: adaptive and static accommodation.

Adaptive accommodation PAMs take the user's performance into account and change throughout the game, meaning that the PAM is not always present in the system or that it changes based on the I/O state. This is the case in the previously mentioned table football game by Rogers et al. [37]. The system calculates an 'imbalance score' depending on goal difference, ball possession, velocity of goal shots, number of pin-shots with sideways movement, and previous help through PAM. Depending on this, it will choose a particular level of PAM. Adaptive accommodation is used in many different contexts [5, 9, 23], and it seems that all systems making use of shared control use adaptive accommodation, as the system does not intervene when the user is performing the actions satisfyingly; only when the user steers off the course does the system correct the input [9, 30, 32].

Static accommodation refers to systems that do not change how they are applied during gameplay. This is for example the auto-pilot from Van Huysduynen et al. [24], which is simply either activated or not. A different example is the rhythm game implemented by Gerling et al. [21], where they implemented various PAMs to balance competition between able-bodied and disabled users. In their study, they implemented three types of PAMs: input balancing (a combination of rule change and assisted success which requires fewer steps and multiplies scores), time balancing (previously mentioned rule change where the timing for a 'perfect' step is increased), and score balancing (assisted success where the score is multiplied). While the amount of PAM that was applied was static throughout a specific round, it was determined by the score differential between the two competitors obtained in an ability pretest in which they measured performance using step count and score, meaning it still took the performance of the player into account on some level.

Bateman et al. [7] measured the difference in player performance (using in-game score) and player experience (using an unspecified survey) between adaptive and static accommodation, specifically for target assistance. In the static condition, the levels of assistance were pre-determined through a pilot study. In the adaptive condition, the assisted player got 10% of the static level for each point behind the unassisted player. They found no significant difference between static and adaptive accommodation, though static accommodation allowed for the assisted player to win more often, but adaptive accommodation resulted in more close plays.

3.3.3 Awareness. Awareness refers to whether users have been told that the PAM is part of the system (which will of course only be relevant for system-employed PAM), and if it is perceivable, i.e. explicit in the form of some kind of indicator. Bateman et al. [7] noted that, after learning about the presence of PAM in their experiment, both the assisted and non-assisted players found PAM fair for both parties and that it had a positive effect on the group play experience, but assisted players' opinions were less favourable than non-assisted players. Baldwin et al. [5] examined the effect of awareness in a multiplayer game, where they employed an adaptive assisted failure PAM (a shield), which was activated by the system when their performance was low. They compared three versions of their game: no PAM, hidden PAM, and PAM with 'full' awareness. Measures were taken using the PENS and electrodermal activity (EDA) to measure arousal. Results showed that assisted players' level of arousal increased in the full awareness condition, indicating that the explicit indicator of performance affected them. The full awareness condition also had a negative effect on levels of autonomy for both the assisted and unassisted player.

As mentioned, this is also tied to how explicit the PAM is. For example, in Gerling et al.'s [21] study, they found that the visibility of the PAM is important to the user experience. While they found that input balancing (as described previously) was effective in increasing the assisted player's performance, it also caused the largest negative effect on their self-esteem due to it being more perceivable than the other PAMs that were implemented. Depping et al. [17] explored the difference between PAM that was hidden and PAM with an explicit indicator in the form of a UI element displaying the 'level of assistance', in a multiplayer shooting game. They found that making it explicit had no significant negative effect on the player experience, although the assisted players' level of perceived competence was lower when the PAM was explicit.

All four of these studies were conducted with multiplayer games and involved an assisted player comparing themselves to an unassisted player. None of the studies we found explored the effect of explicit PAM in a single-player game. It should also be noted that, aside from Baldwin et al.'s study, it is unclear exactly how granular the awareness was, i.e. whether they were aware of only its *presence* or also aware of the *effect* it had. Baldwin et al. [5] do distinguish between the two types of awareness in their study, however they did not explore the difference between them, only between no awareness and full (i.e. both presence and effect) awareness. We believe this to be an important distinction, as awareness of the effect of the PAM makes it possible to potentially exploit it by e.g. deliberately performing worse in order to receive the benefits of it. However, this was not explored in any of the papers we found.

It is important to note also that just because something is not disclosed or has an explicit indicator, this does not mean players cannot perceive the presence of a PAM. For example, in our previous study, we deliberately attempted to hide the presence of sham input by making its feedback identical to the players' success input [38]. However, as normal success input would trigger success feedback instantly, and our sham input would trigger success feedback at a variable time delay, players may have been able to perceive the presence of the PAM. The way Évain et al. [50] implemented their BCI paradigm - by always triggering feedback at the end of the task phase, rather than immediately upon received input - made it easier to hide sham input, as the timing of feedback triggered by sham input was (in theory) indistinguishable from feedback triggered by user input. As evidenced by this, the timing of the PAM feedback is especially important in sham input in terms of the users' awareness.

3.4 Discussion of literature review

Based on our analysis of the literature, we were able to identify a number of PAMs based on how they change the mapping from input to output, in terms of success and failure. It is important to note however that this is only one way of thinking about different PAMs, and classifying them based on different criteria may be just as valuable. It is also likely that there are a number of PAM characteristics that we have not listed here which may affect the user experience. We make no assumption that our literature review offers a complete overview of the field of PAMs, as it is a very wide field with a very varied vocabulary, making it difficult to define specific search terms.

The models we have created to describe the PAMs we identified are relatively simple, and it is not unlikely that there will be methods of assisting players that cannot necessarily be described in this way. Nevertheless, since we are working with systems that struggle with a relatively large amount of false negatives, we find it interesting to see how the player experience is affected based on how the player's successes and failures are interpreted by the system.

As we have not seen any other study which differentiates between PAMs in this way, but instead have mainly seen studies which examine the effects of the different characteristics, we therefore decided to conduct an experiment with this purpose. In order to limit our experiment, we decided to focus only on three of the different PAMs: sham input, assisted success, and assisted failure, as these lend themselves more easily to a binary input.

4 SYSTEM DESIGN

To explore how the different types of PAMs affect users' experience in a brain computer interface (BCI)-like system, we implemented a simple game. Our choice of game was based on a number of criteria. In order to ensure all players experienced the same amount of failure and success within the game, we needed to control how the system would respond to every input. Restraining the decision-making of the users requires a linear narrative, and for the user to not feel restrained, this narrative needed to lend itself naturally to making few choices. To make it easier to control, we decided to work with a binary input. However, this does not mean that every task should only have one or two narratives (i.e. success or failure); chaining multiple binary inputs as subtasks creates the opportunity to change the circumstances of success or failure for the main task, hopefully increasing the users' engagement. The need to control all inputs also meant we could not work with a real motor imagery (MI) BCI-system. Instead the 'stand-in' for BCI input is a key sequence performed in a specific order within a short amount of time.

4.1 Game design

In the end, we decided that a fishing game lent itself well to the constraints above. A screenshot from our game can be seen in Figure 8. As we knew from our background research, explicit user representations can help increase players' sense of agency, the user is represented through a fisherman. The goal of the game is for the fisherman to catch fish swimming in a lake. The lake is split into a 3 by 7 grid, with fish spawning in one of the three lanes. Once it spawns, the user is supposed to move their hook to the corresponding lane using the 'up' and 'down' arrow keys. When the fish is hooked (which will always happen in the middle column), the user will have to catch it.

The task of reeling in the fish is the part of the game where BCI-input would be required, or in case of the experiment, key sequence input. This process is split into multiple subtasks of reeling the fish in, the amount depending on which lane the fish spawned in (the further down, the more subtasks to perform). Upon getting the fish on the hook, the BCI interaction paradigm (described further below) starts, with a 2 second preparation phase, explicitly shown through a progress bar (see Figure 8b). After this, the user has a 1 second input window in which to perform the correct input to reel the fish in by one lane. While the user is told that they will need to perform the correct key sequence ('HKJL'), the game was implemented so that any input of four keys would be accepted. This was to ensure a more uniform experience between players, regardless of typing skills.

Careful consideration was put into how to convey failures while not making the user feel discouraged. Therefore, a single failure makes the fish swim one column away instead of down, so as not to regress, and the user has to fail three times to lose the fish. On a success, the fish will move one column up. The different types of feedback are expanded upon later in this chapter. More details about the design of our game and the thoughts that went into it can be found in Worksheets, Chapter 1.

Hopefully, the fishing narrative will be seen as an enjoyable, leisurely activity, while also offering a narrative explanation for the frequent failures. The art style is kept vibrant and fun, and the fish, once caught, are highly detailed and colourful. Additionally, various sound effects are used to indicate various in-game events and to add ambience. Details about the implementation of our game can be found in Worksheets, Chapter 2.

4.2 Interaction paradigm

The interaction paradigm used in this system draws from the BCI literature mentioned in section 2.1.1, but has been slightly altered to account for test-time, input-type, and context. Therefore, the interaction is less rigid than in other BCI systems, and the length of various phases changes depending on user actions and outcomes. An illustration of our interaction paradigm can be seem in Figure 9.

Figure 9: The interaction paradigm for our game. The inter-trial interval is only a part of the paradigm once the fish has been either caught or lost.

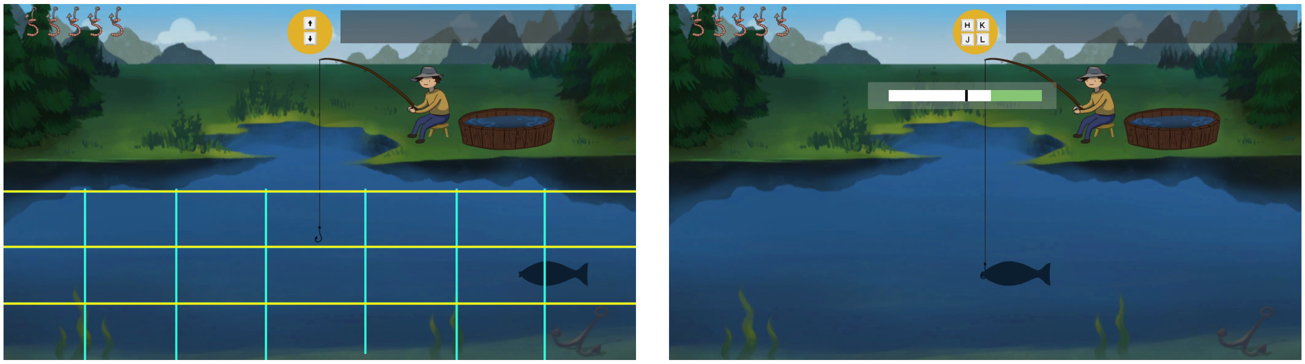
The cue phase is initiated by the fish appearing on the screen. It takes the fish 7.5 seconds to reach the hook, but if the user misses, it takes approximately an extra 10 seconds for it to come back to the same spot. In theory, this means that the cue phase could go on forever, if the user keeps missing the fish. As described above, the preparation phase is only 2 seconds long, which is shorter than the average preparation phase as seen in Figure 1. As this phase is usually included to minimise noise, and this was not a concern for us with this input method, it was lowered for time-saving. This phase is followed by a short 1 second task phase window, also to limit time as well as to make the task harder to accomplish and lower the players' access to the ground truth of whether they succeeded or not.

The feedback phase is between 1 and 5 seconds, depending on the feedback the user is getting, as the length depends on when the feedback type is determined and how long the animation is. After the fish has been caught, there is a 2 second window before the next fish appears.

4.3 Interaction blocks

For the sake of our experiment, the game experience should be the same for every participant. For example, people experiencing 18 successes and 4 failures will not be rating frustration in the same context as people experiencing the opposite ratio. Therefore, the narrative was completely pre-determined to control the recall rate, subtask amount, and feedback-type ratio. The output of the four random key-presses was not determined by anything in control of the user (although four key-presses were required to trigger an output; failing to do this would simply restart the interaction paradigm).

As stated above, catching a fish takes between 1 and 3 successful subtasks, while losing a fish takes 3 failed subtasks. This creates a somewhat complicated situation, as individual subtasks cannot be lined up randomly, as this could create different main task success-rates for different people.



(a) A screenshot from our game during the phase of the game where the user moves the hook with the arrow keys. The horizontal yellow lines indicate rows, while the vertical light blue lines indicate the columns within which the fish can appear.

(b) A screenshot from our game during the phase of the game where the user reels in the fish by performing the key sequence.

Figure 8: Screenshots from our game.

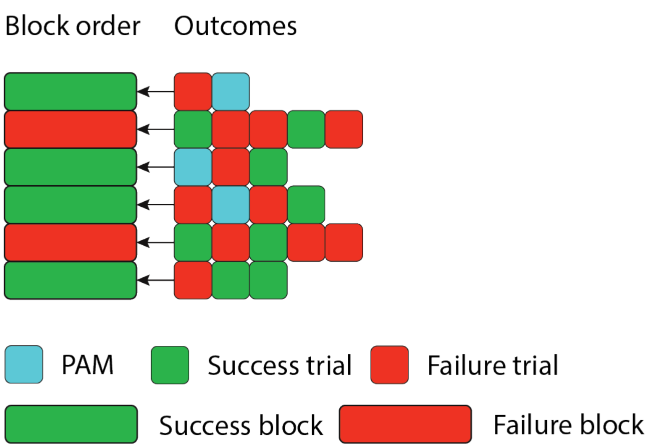


Figure 10: An example of a set of interaction blocks.

To fix this, the main task outcomes need to be lined up instead. This means that subtasks cannot be seen as individual, unconnected interactions, but as part of a group making up a main task. For this, sets of interaction blocks were implemented. Interaction blocks have a pre-defined outcome (catching or losing a fish), and the block is filled up with various subtask outcomes (reeling in fish, fish swimming away, or a PAM) that amount to this outcome. This combination needs to be carefully designed to match the main task outcome.

Additionally, it had to be designed in the context of the experiment. This means that any memory biased experiences should be avoided; specifically, repetitions and peak-end bias was of concern. Kahneman et al. [25] noticed that people were more accepting of pain if the end was less painful, indicating that people do not rate their aversive experiences on duration, but as a relationship between peak and end discomfort. In theory this means that an interaction block ending with three failures will be rated differently than a block where three failures are spread throughout the experience. Moreover, we avoided repetitions to limit the effect of a peak on the users' ratings. An example of a set of interaction blocks for assisted success can be seen in Figure 10. Each set of blocks includes 22 trials, with 11 of them resulting in a failure and 11 resulting in a success. The different PAMs then replace either success or failure trials, depending on the type of PAM. We chose to implement 3 PAM trials, corresponding to 13.6% of the trials. The reason for this ratio is that, in our previous study, we found the biggest effect on frustration and perceived control when adding 15% sham to a similar system with a 50% recall rate [38]. The ratio of trial outcomes can be seen in Table 1.

	Control	Sham	Assisted success	Assisted failure
Positives	11	11	8	11
Negatives	11	8	11	8
PAM		3	3	3

Table 1: Trial outcomes for the four conditions

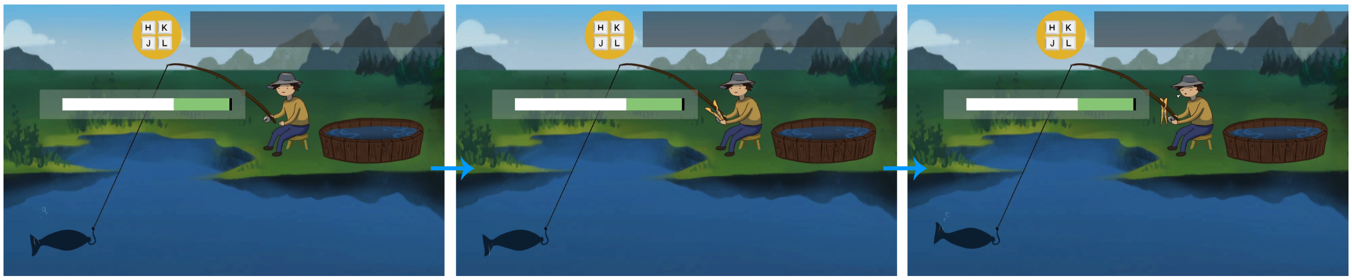
The users were not told about this prior to the experiment, and were made to believe that they (and not the system as it actually was), were in control of the game. How they were informed and what they were told about the game is described in Chapter 5.

4.3.1 Feedback types. There are five different feedback types for the subtask implemented in the game: success, failure, sham input, assisted success, and assisted failure. All five feedback types have in common that they will be executed only if the input is accepted, i.e. if a key-sequence of four key-presses has been registered within the input window. The specific feedback type is then determined based on the specific interaction block.

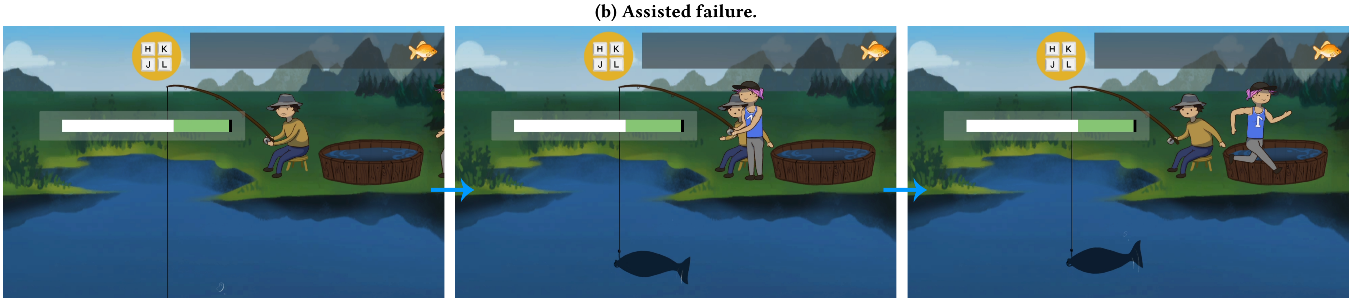
The success feedback is executed at the moment the input is accepted. It is experienced as the user representation smiling and turning the reel on the fishing rod and the fish moving up one lane. It is accompanied by sounds of spooling in the reel. Failure feedback is instead executed when the input window closes, to indicate that the player has failed (even if they did not). Failure feedback is experienced as the fish swimming more frantically and away from the current column. The user representation lets go of the reel and looks concerned. Additionally, there are splashing sounds when the fish struggles.

Just as for success feedback, assisted success is executed the moment the input is accepted. The user representation rummages in their pocket to get a leaf, which they eat. This increases their muscle mass and rips their sleeve. They then turn their reel fast and reel the fish up by two lanes (see Figure 11a). Sound effects indicate all these actions. The other two PAM feedbacks are executed at the same time a normal failure would have happened. Assisted failure also shows the fish struggling, but before it tugs away, the user representation puts a clothes pin on the rod and breathes a sigh of relief, stopping the fish so it remains in its current column (see Figure 11b. Sham input shows a woman running in to pick up the user representation's fishing rod as she says "Let's get to work". When she does

(a) Assisted success.



(b) Assisted failure.



(c) Sham input.

Figure 11: Storyboards of the feedback shown when the different PAMs are activated.

so, the user representation exclaims in surprise, and she reels the fish in by one lane (see Figure 11c).

As evidenced by this, we have chosen to incorporate narrative explanations for our PAMs. Our reason for this is that we hope to make the input-output mapping more clear to users in order to get a better idea of how this affects players, as well as to increase their judgement of agency (see section 2.2). This means all our PAMs rate highly in terms of awareness (see section 3.3.3).

Aside from the feedback for the individual interactions, there is also feedback for when the player catches or loses a fish. When catching a fish, the fisherman will excitedly hold the fish above his head, at which point it will be revealed which type of fish it is (instead of the silhouette seen previously). A sound effect of the fisherman cheering plays while he celebrates. When losing the fish, the fisherman will look dejected as he reels in the line back to its starting point, while a sighing sound effect plays.

5 METHOD

To evaluate the effect of the different PAMs on the player experience, we designed an experiment, described in this chapter.

5.1 Experimental design

Our experiment was a within-subjects experiment, where participants played four conditions each (a control condition without PAM, and one condition per PAM). To avoid any order bias, the order was decided using a latin square, where each condition appeared four times in each experimental position. The dependent variables for the experiment were perceived control and frustration.

Our hypotheses for the experiment were: ***H0:*** *The presence of performance-accommodation mechanisms will have no effect on players' levels of frustration and perceived control when applied to a system with a 50% recall rate.* ***HA:*** *The presence of performance-accomodation mechanisms will have an effect on players' levels of frustration and perceived control when applied to a system with a 50% recall rate.*

5.2 Participants

The purpose of this study was not necessarily to see the effect of PAMs on stroke victims using BCIs, but to see their effect on experiences in low recall systems. Therefore, we decided that not testing with stroke patients would not decrease the validity of the experiment (though, considering their prior experiences, stroke patients might be more forgiving of being helped). 17 people were contacted directly with an inquiry about their willingness to participate, and accepted. These people were of various backgrounds, ages and gender, and differed in their technical knowledge. Participants were told only about the overall purpose of our experiment (how different types of 'help' would affect the player experience in a game with a 'difficult' input method), and were not aware of the controlled nature of the experiment.

Because of problems with logging data from the game for one participant, we ended up having 16 usable participants.

5.3 Apparatus

Because the study was conducted during the COVID-19 pandemic, the evaluation had to be performed remotely through various communication channels. Participants therefore played the game on their own PCs. We sent builds of the game to participants (in the various versions for the different conditions) and asked them to turn on screen-sharing during the experiment, so that we could confirm that the procedure was followed.

We measured the participants' self-reported levels of perceived control and frustration using Likert scales. For both variables, the Likert scales were used to both gather measurements about the participants' overall and more incident-level self-reported levels.

The perceived control questions were formulated as "I felt in control [...]" and were rated on a 7-point Likert scale with the following items: (1) Strongly disagree and (7) Strongly agree. Participants were asked to rate their overall perceived control per condition (i.e. in regard to a full playthrough of a single condition), and to rate their perceived control on incident-level, such as when they reeled the fish in, when it tugged away, and when it escaped. They were also asked to rate their control in the context of the individual PAMs. Additionally, people were asked to rate, on a scale from 1 to 100, the probability of them reeling a fish up by one lane in that particular condition.

Participants were also asked to rate their frustration in context of the PAM by rating an "I felt frustrated when [...]" prompt on a 7-point Likert scale ranging from: (1) Strongly disagree and (7) Strongly agree. Participants' overall and incident-level frustration ratings were answered on a 7-point Likert scale with the following items: (1) Absent, (2) Barely perceptible, (3) Faintly present, (4) Light, (5) Marked, (6) Pronounced, (7) Strongly pronounced, on a prompt starting with "How much frustration did you feel [...]". Participants were asked about their frustration level per individual condition and for the whole experiment, and during the same incident-levels as the perceived control questions.

Additionally, participants were asked to rate their perceived control and frustration during the current condition to their experience in the previous condition. They were asked whether they would rate them higher, the same, or lower than the previous. The full questionnaires used in our experiment can be found in Worksheets, Chapter 3.

The game also logged data from the system itself. Essentially, two kinds of data were logged: game-event data and input-event data. Game-event data logs visual, in-game events, that indicate some changes to the user. This is feedback, indicators, signifiers, game stage, etc. Input-event data logs what keys the user is pressing. This data tells us what objective experience the user was having playing the game.

5.4 Procedure

Participants were informed about prerequisites of the experiment prior to meeting up in a virtual meeting room on Teams, Discord, or Messenger, and were asked to sign a consent form, through an e-mail. (The material that was sent to participants prior to the experiment can be found in Worksheets, Chapter 3.) They were told in general terms what they would be tasked to do. They were also told to prepare a piece of fabric, microphone and headphones, and that they had a PC to perform the experiment on. They were informed that we would use any communication software they preferred, but that we could not guarantee how the data collected by the software would be used.

When the participant joined the meeting, they were sent a Google Drive link to a .zip-file with the game builds they should play, and it was confirmed that they had a piece of fabric at hand. As they were downloading the file, they were told to fill out a demographic pre-experiment survey. They were then told about the game, and about the (fake) key sequence they would have to input.

Participants were told that the key sequence they were supposed to perform was very specific and generally hard to do, and that the reason for this was to forcefully lower their success rate in order to introduce helpful elements. They were therefore led away from the actual implementation, that four random keys would work. They were told to perform the key sequence 'HKJL' within one second and with a consistent rhythm (spacing between key presses). The rhythm was specifically kept vague to make it more difficult for them to discern whether that aspect was performed correctly. Additionally, people were told to only use one hand and that they should cover that hand with a piece of fabric. To get them familiar with it (to play into the narrative of their input having an effect) and the game, they were told to play a short practice mode with just 3 trials, ending in catching the fish. As they progressed through different parts of the game, each part and the UI elements was explained to them.

After the explanation and practice, they were told which build to start with. Every condition consisted of catching four fish and losing two, but the ratio of trial outcomes were different, as seen in Table 1. After finishing a condition, they were sent the appropriate survey, and as they were filling it out, we asked them to elaborate on some of their answers.

The full script used in the experiment can be found in Worksheets, Chapter 3.

After having played all the conditions and filled out all the surveys, the participants were told to send an auto-created folder with the log data to the facilitator, after which the participants were debriefed and told how the game actually functioned.

6 RESULTS

The quantitative data was analysed using appropriate statistical methods, which will be described in the following sections. The results of these analyses were supplemented by qualitative data, which has been coded in order to perform a content analysis. We conducted non-parametric statistical tests due to our data violating assumptions of normal distribution, homogeneity of variance and interval data (as Likert scales are not interval).

6.1 Perceived control

Results of Friedman's ANOVA indicated no significant differences in overall (i.e. in regard to a full playthrough of a single condition) perceived control between the four conditions. However, results did indicate a significant difference in participants' perceived control when asked how much in control they felt specifically when helped by the different PAMs ($\chi^2(2, N = 16) = 9.63$, $p < 0.01$). A follow-up post-hoc test specified that the significant difference was between sham input and assisted success (AS), with the mean scores being significantly lower for sham input than AS (see Table 2). The test yielded a large effect size of 0.524. This indicates that participants felt more in control during the interactions where PAM was employed when the PAM was AS compared to sham input.

This trend can also be seen in how people phrased their feelings about the two PAMs, more often using the word 'she' and not attributing credit to themselves when sham input was employed (*"Oh no, she took my fish!"* and *"I haven't caught any fish yet! ... I only caught two fish and both times this girl helped me."*), while more predominantly using pronouns like 'I' and 'me' within the AS condition (*"The power-up coming from the character itself really made the difference - I am both the power-up and the main character."* and *"[...] me doing some kind of trick, it's like, oh I didn't know I could do that, but cool."*).

Further tests did not indicate any significant differences between the four conditions in questions about perceived control regarding specific game events (when they reeled the fish in, when it tugged away, and when it escaped). Participants were generally confused about what caused a PAM event to be employed, thinking that they were most likely random.

6.2 Frustration

Similarly to perceived control, a Friedman test indicated no significant differences in overall (i.e. in regard to a full playthrough of a single condition) frustration between the four conditions. Despite an insignificant difference, the means show that the scores for the sham input condition were the highest out of the four groups (see Table 3). Moreover, we found a significant difference in participants’ frustration when asked how they felt specifically when helped by the different PAMs (χ^2(2, N = 16) = 16.33, p < 0.01). A follow-up post-hoc test again showed that the significant difference was between sham input and AS, with the mean scores being significantly higher for sham input than AS (see Table 2). The test yielded a large effect size of 0.522. Similarly to perceived control, this indicates that participants were more frustrated by the instances of PAM in the sham condition compared to AS.

The qualitative data also shows that sham input was the only condition that received any negative comments from the participants, expressing that they felt the non-playable character intruded on them playing, saying for example, *“[t]he runner kind of felt like ‘Oh, you’re so bad at this, I’m just gonna do it for you’”*.

There was no indication of significant differences between the four conditions in questions about frustration regarding specific game events (when they reeled the fish in, when it tugged away, and when it escaped).

6.3 Correlation between frustration and perceived control

Regression analysis indicates a significant negative relationship between overall frustration and perceived control for all conditions, b = -0.25, t(62) = 4.02, p = 0.049. As frustration decreases, the perceived control increases, with the trend being most apparent within the sham condition as seen in Figure 12, b = -0.61, t(14) = 8.08, p = 0.013).

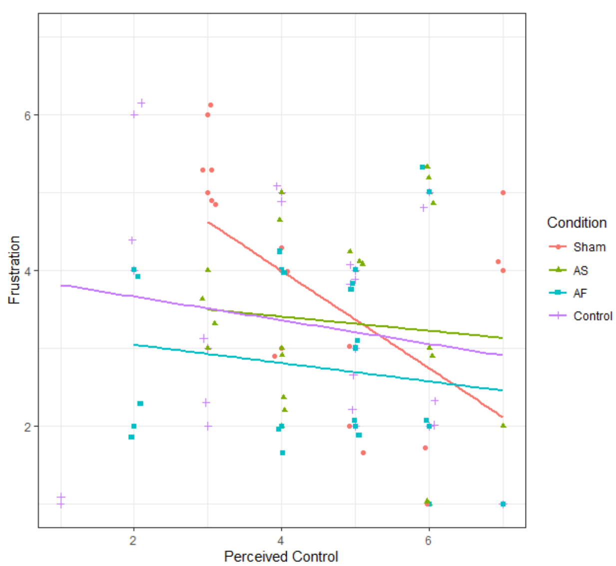


Figure 12: Scatter plot visualising the relationship between overall frustration and perceived control, for all four conditions.

Moreover, looking at the scores for frustration and perceived control specifically when being helped by a PAM, we also found a significant negative relationship for all three PAM conditions, b = -0.51, t(46) = 16.31, p < 0.01. As frustration decreases, the perceived control increases, with the trend again being the most apparent within the sham condition as seen in Figure 13, b = -0.64, t(14) = 9.42, p < 0.01.

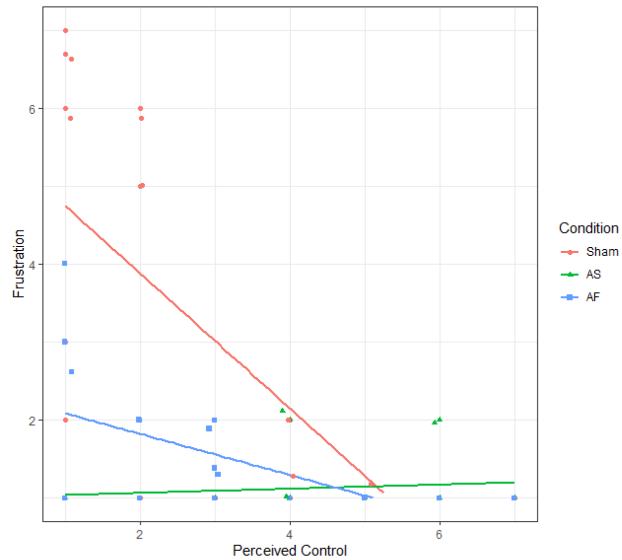


Figure 13: Scatter plot visualising the relationships between frustration and perceived control in regard to PAM being employed within the three PAM conditions.

6.4 Blame attribution

Based on the qualitative data gathered during the testing sessions, we categorised each individual playthrough of each participant with a blame attribution factor, specifying where they attributed the reason for the incongruency between their actions and the events playing out in the game. They could be categorised either as blaming themselves, the system, or neutral (when they did not express any thoughts in regard to blame attribution). Out of 52 comments identified as indicating attribution, we found that 27 of them were towards the system, while 25 were towards the participants themselves (note that some participants’ attributions would change between conditions). People attributing blame on the system showed confidence in their abilities (*“Sometimes I felt like I did it correctly, but it didn’t work”*, *“I didn’t feel guilty for failing to catch the fish ’cause I don’t really feel like I caused it”*, and *“I did it this time and it still failed, I’m like sure of it.”*), while participants blaming themselves expressed that they had trouble getting the input right (*“I’m trying to get my fingers to remember the key sequence!”*, *“I guess I’m just bad at it”*, *“[...] I didn’t get the key sequence right, but I think I was messing up, so I don’t know.”*).

An ANOVA showed no indication of significant difference in terms of reported frustration or perceived control overall, depending on how the participants attributed blame. However, as shown in Figure 14, for participants who attributed blame to themselves, there was a significant negative relationship between frustration and perceived control, b = -0.55, t(21) = 9.05, p < 0.01.

A Friedman’s ANOVA indicated that, depending on the condition, the participants’ blame attribution was significantly different (χ^2(3, N = 16) = 8.97, p = 0.0322), however a post-hoc test did not reveal which groups were significantly different. Looking at the boxplots seen in Figure 15, we can see that the largest difference in blame attribution occurred between sham (10 comments blaming the system vs 4 blaming themselves) and AS (10

	Sham Input	Assisted Success	Assisted Failure
Frustration	3.56	1.13	1.63
Perceived Control	2.31	4.31	2.75

Table 2: The means of scores for frustration and perceived control in regard to PAMs being employed.

	Control	Sham Input	Assisted Success	Assisted Failure
Frustration	3.31	3.56	3.31	2.75
Perceived Control	4.31	4.69	5.00	4.50

Table 3: The means of scores for frustration and perceived control in regard to the whole playthrough.

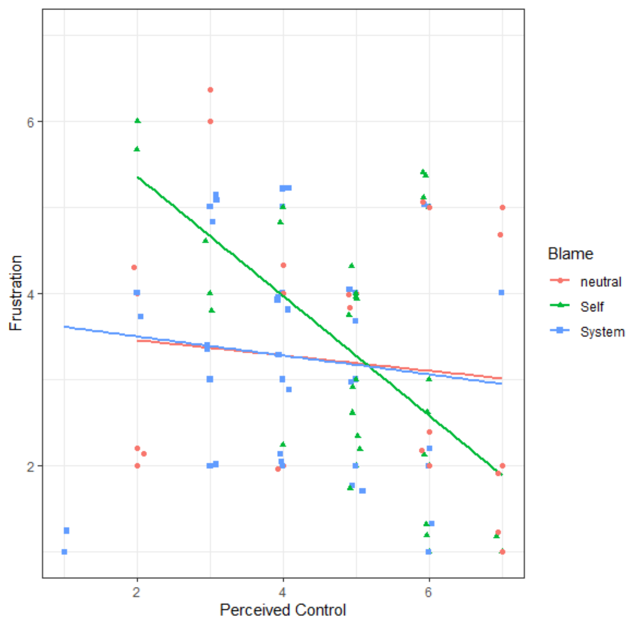


Figure 14: Scatter plot visualising the relationships between overall frustration and perceived control, across all four conditions, grouped based on whom the participants attributed the blame for the events to.

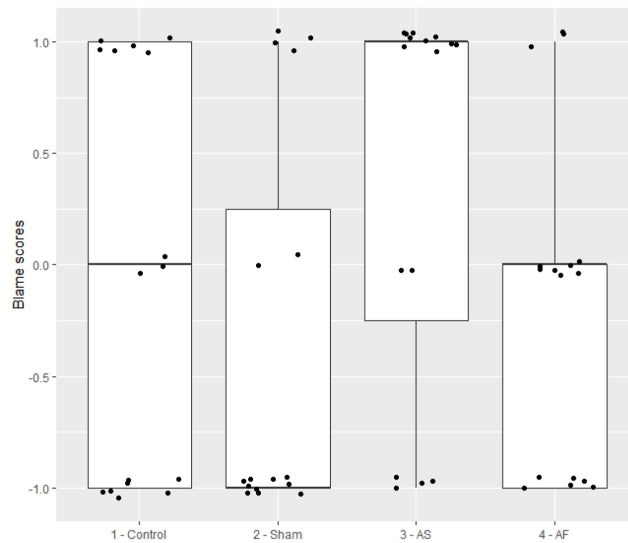


Figure 15: Boxplots showing the scores for blame attribution between the four conditions. Blame attribution to self is coded with a "1", a lack of blame attribution is coded with a "0" and blame attribution to the system is coded with a "-1".

comments blaming themselves vs 4 blaming the system). A Wilcoxon signed-rank test confirmed that the significant difference in blame attribution is between sham input and AS (p = 0.0357, r = 0.04). This indicates that participants were more likely to blame the system for in-game events during the sham condition, while they were more likely to blame themselves in the AS condition.

6.5 Estimate of probability

As a last question within the survey in our experiment, participants were asked to estimate how likely they were to succeed in reeling a fish up by one lane based on the experience of a particular condition.

A Wilcoxon signed-rank test indicated no significant difference between participants' perceived control in regard to reeling a fish up by one lane, and their estimate of probability (p = 0.0707). However, after looking at boxplots seen in Figure 16 it can be deduced that generally, participants estimated their probability of reeling up a fish slightly higher than how much they felt in control.

6.6 Learning effect

Despite the precautions we took in order to eliminate any possible learning effect amongst the participants, we decided to check for whether there actually was one. This was done by looking at various ratings from the participants based on how many times they had been exposed to the system.

A number of Friedman's ANOVAs reported that there was no significant learning effect when comparing scores of frustration, perceived control, blame attribution and estimate of probability depending on how many times they have been exposed to the system. In fact, in the case of frustration, even though the test was insignificant, after looking at box plots of the data (which can be seen in Figure 17) it can be inferred that there seems to be general decrease in terms of frustration, the later throughout the experiment it was reported.

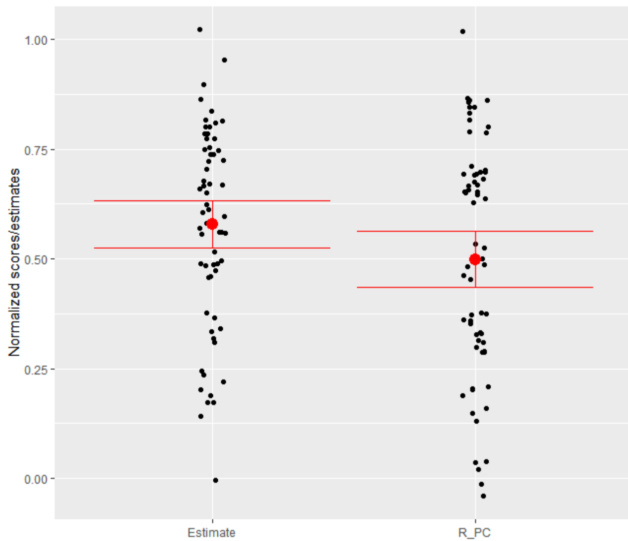


Figure 16: Error bars showing normalised scores across all of the conditions for both the estimate of probability and perceived control in regard to reeling up a fish by one lane.

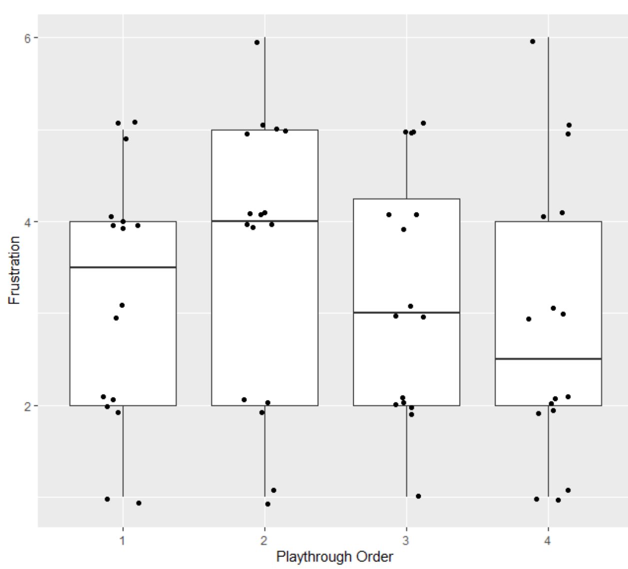


Figure 17: Box plots showing the scores for frustration depending on the amount of times they have been exposed to the system.

7 DISCUSSION

Looking at the results of our analysis, we can reject our alternative hypothesis, as we did not find a significant effect of PAMs on participants' frustration and perceived control with our system. However, our results do indicate other implications for the implementation of PAMs in a brain computer interface (BCI)-like system.

We can deduce from the analysis of both quantitative and qualitative data, that participants preferred assisted success, scoring the lowest in frustration and highest in perceived control, while sham input was the least preferred. People tended to attribute their actions more to assisted success than sham input, because the action and feedback happened to the user representation in the former, but not in the latter. This is in line with research done by Fard and Grosse-Wentrup [19], showing that people tend to rate their perceived control higher when they attribute successes to themselves.

As seen in the results in Chapter 6, sham input was rated as the most frustrating and inducing the lowest perceived control. It would be natural to conclude that, as participants were more likely to reel the fish in when sham input was applied, it would reduce frustration and increase perceived control, as seen in previous studies [45, 50] as well as our own [38]. Interestingly, we do not observe the same lowered frustration and increased perceived control compared to the control condition when introducing sham input in this study, as we did in our previous study. In our previous study, we observed that increasing the amount of sham input provided decreased frustration and increased perceived control, but this time, sham input was rated as highly frustrating with low perceived control. This could be explained by the hidden nature of sham input in the previous experiment and the explicitness of sham input in this experiment. As people had no knowledge of the presence of sham input in our previous experiment (unless they were able to perceive it themselves), they would naturally attribute any kind of input to themselves, thereby increasing their perceived control. Explicitly showing them that sham input happened, breaks the illusion and seems to be rather displeasing for the user.

In terms of attribution theory [17], it seems that changing the narrative around the sham input caused participants to change it from an internal attribution to an external attribution. Based on qualitative data, this seems to be very tied to how we designed the narrative explanation for sham input, i.e. having an outside agent perform the action instead of the user representation. Based on this, it seems that our attempt to make the causal chain (i.e. the input-output mapping) more apparent and increase the users' judgement of agency [28] ended up negating any benefits the increase in positive outcomes had in our previous experiment. This is in line with previous research showing that a higher level of awareness of PAMs can have a negative effect on the player experience [5, 7, 17, 21], although all of these studies applied PAMs in multi-player games and did not measure perceived control or frustration (nor did they implement sham input specifically). In this sense, sham input that is hidden and sham input with a narrative explanation are seemingly so different that they should not be assumed to have the same effect.

Some of the negative effects of this explicit, 'explanatory narrative'-sham input could potentially have been mitigated by implementing it as a user-employed PAM. Several participants expressed some frustration with the fact that they did not know when the PAM would be employed, or why, i.e. that they could not control it themselves. As stated in section 3.3.1, we found only little literature comparing system- and user-employment, although we do know from Rogers et al.'s [37] study that their participants preferred user-employment for explicit, high awareness PAMs (although just as previously, their study was concerned with a multi-player game, and they did not measure frustration and perceived control). This implies that the characteristics of PAMs are not only useful for contextualising them, but also to modulate their effectiveness.

As seen in Figures 12 and 13, we found a significant negative relationship between perceived control and frustration in the sham condition, both overall and specifically when sham input was employed. Notably, we only saw this for sham input, while we did not see this effect for the rest of the conditions. Especially for PAM-specific frustration and perceived control, we can see that assisted success barely has a slope, and the frustration level is generally very low, despite that the ratings for perceived control vary from low to high. This could indicate that perhaps the frustration participants experienced in the sham input condition was not caused by them feeling less in control but by something different, such as the characteristics discussed above.

Based on our results, we can see that sham input did not appear to have the same effect on overall perceived control as it did on perceived control specifically when PAM was applied, despite the presence of the PAM being the only difference between conditions. The natural assumption would be that the effect of the PAM would be seen on the overall scores as well. The results, as they are, indicate that there is a difference in how people experience PAMs, but that implementing them in a system in the same manner and form as we did, does not have a significant effect on the overall experience. One reason for this somewhat surprising result could be that, because of the way we framed the experiment, participants were led to focus mostly on the PAMs themselves, and put less thought into their answers regarding the rest of the system. Another explanation could be that, while this type of sham input had an effect on player experience in the moment of it happening, it was not pervasive enough to have an effect on the rest of the experiment, perhaps due to the small quantity of it.

We considered implementing false positives in the system, to make it more similar to actual BCIs. We decided that this would possibly convolute the experiences for the players, as the input-timing and then the execution of a false positive could give different impressions of cause and effect between different users, and take away trials for the more essential outcomes (for the sake of keeping testing time low).

Based on the analysis of various scores per number of exposures to the game, we found no learning effect, indicating that our experimental design did not negatively affect our results. However, regardless of the lack of significant statistical difference, the box plot of overall frustration rating depending on how many exposures to the system the participant had had (Figure 17), does seem to show a general negative trend. People seemed to have a higher initial frustration playing the game, which is then increased on the second exposure before decreasing over the last two. Our theory is that people were surprised about the low recall rate in the game, which initially increased their frustration on the second exposure, as they still could not get the hang of it. Over the last two exposures, people realised that the unreliable input method is just the nature of the game, either accepting the difficulty of it or suspecting that the system was interfering. Under different circumstances, we would not have had every participant play every condition but would have designed a between-subjects experiment instead. Whether it would be better to measure people in their initial state or after a few exposures we do not know. The first exposure is the initial reaction, but future users would presumably spend more time in the acceptance state, meaning that measuring the effect after several exposures would perhaps provide a better image of how the system would be perceived in a context such as BCI rehabilitation.

We believe that our experiment offers an initial look into the nature of various PAMs in a low recall system, as it revealed many aspects that should be considered and studied further: is there a significant difference in hidden and explicit PAMs of the same kind, does the narrative explanation of the PAM make a difference, are some PAMs more effective than others and does it depend on context, what effect do the different characteristics have, and so on.

As mentioned in Chapter 3, we also recognise that the input-output mapping models introduced in this paper are not exhaustive, and that there potentially are several instances where a mapping cannot be visualised using the models. Nevertheless, we hope that the models can be used to inspire more encompassing models in the future.

8 CONCLUSION

Our study showed that sham input did not cause the same increase in perceived control and decrease in frustration as we saw in our previous study [38], indicating that explicit sham input with a narrative explanation is very different than sham input that is deliberately hidden from the user. We also saw that assisted success, where the players' successful input is amplified, causes a larger positive effect on both perceived control and frustration compared to sham input, which replaces the players' failed input with system-based successful input. Qualitative data points to this being due to how players attributed the causes for the different effects, indicating that players respond more positively to a PAM when they feel that their input is part of why they performed better than expected.

We believe that this experiment was only the first step to understanding how different designs of PAMs within unreliable systems can improve the users' experience and that there is much more research work to be done within this domain.

8.1 Future works

Our experiment required the users to play the game four times, but only within a short amount of time. Ideally, a more extensive experiment would be carried out, since the context of our target group requires them to use such system for prolonged amounts of time.

As much as we can theorise that it was the difference in explicitness that caused the difference between results within this study and our previous study, we believe that a more thorough exploration of the different PAM characteristics could reveal even more interesting implications and design suggestions for the future.

Moreover, we only tested three of the five PAMs we suggested within this study - depending on the context of a system, different PAMs might lend themselves better or worse, and therefore we believe that testing both shared control and rule change might provide interesting results as well.

ACKNOWLEDGMENTS

Acknowledgements

REFERENCES

[1] Gazihan Alankus, Rachel Proffitt, Caitlin Kelleher, and Jack Engsberg. 2011. Stroke Therapy through Motion-Based Games: A Case Study. *ACM Transactions on Accessible Computing* 4, 1 (Nov. 2011), 3:1–3:35. https://doi.org/10.1145/2039339.2039342

[2] Maryam Alimardani, Shuichi Nishio, and Hiroshi Ishiguro. 2018. Brain-Computer Interface and Motor Imagery Training: The Role of Visual Feedback and Embodiment. *Evolving BCI Therapy - Engaging Brain State Dynamics* (Oct. 2018). https://doi.org/10.5772/intechopen.78695 Publisher: IntechOpen.

[3] Dennis Ang and Alex Mitchell. 2017. Comparing Effects of Dynamic Difficulty Adjustment Systems on Video Game Experience. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play (CHI PLAY '17)*. Association for Computing Machinery, Amsterdam, The Netherlands, 317–327. https://doi.org/10.1145/3116595.3116623

[4] Dollaporn Anopas, Massamon Horapong, and Yodchanan Wongsawat. 2013. BCI-based neurorehabilitation and prediction system for stroke patients. In *Proceedings of the 7th International Convention on Rehabilitation Engineering and Assistive Technology (i-CREATe '13)*. Singapore Therapeutic, Assistive & Rehabilitative Technologies (START) Centre, Gyeonggi-do, South Korea, 1–4.

[5] Alexander Baldwin, Daniel Johnson, and Peta A. Wyeth. 2014. The effect of multiplayer dynamic difficulty adjustment on the player experience of video games. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems (CHI EA '14)*. Association for Computing Machinery, Toronto, Ontario, Canada, 1489–1494. https://doi.org/10.1145/2559206.2581285

[6] N. Barrett, I. Swain, C. Gatzidis, and C. Mecheraoui. 2016. The use and effect of video game design theory in the creation of game-based systems for upper limb stroke rehabilitation:. *Journal of Rehabilitation and Assistive Technologies Engineering* (May 2016). https://doi.org/10.1177/2055668316643644 Publisher: SAGE PublicationsSage UK: London, England.

[7] Scott Bateman, Regan L. Mandryk, Tadeusz Stach, and Carl Gutwin. 2011. Target assistance for subtly balancing competitive play. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. Association for Computing Machinery, Vancouver, BC, Canada, 2355–2364. https://doi.org/10.1145/1978942.1979287

[8] James K. Beggan. 1992. On the social nature of nonsocial perception: The mere ownership effect. *Journal of Personality and Social Psychology* 62, 2 (1992), 229. https://doi.org/10.1037/0022-3514.62.2.229 Publisher: US: American Psychological Association.

[9] Jared E. Cechanowicz, Carl Gutwin, Scott Bateman, Regan Mandryk, and Ian Stavness. 2014. Improving player balancing in racing games. In *Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play (CHI PLAY '14)*. Association for Computing Machinery, Toronto, Ontario, Canada, 47–56. https://doi.org/10.1145/2658537.2658701

[10] María A. Cervera, Surjo R. Soekadar, Junichi Ushiba, José del R. Millán, Meigen Liu, Niels Birbaumer, and Gangadhar Garipelli. 2018. Brain-computer interfaces for post-stroke motor rehabilitation: a meta-analysis. *Annals of Clinical and Translational Neurology* 5, 5 (March 2018), 651–663. https://doi.org/10.1002/acn3.544

[11] Ujwal Chaudhary, Niels Birbaumer, and Ander Ramos-Murguialday. 2016. Brain–computer interfaces for communication and rehabilitation. *Nature Reviews Neurology* 12, 9 (Sept. 2016), 513–525. https://doi.org/10.1038/nrneurol.2016.113 Number: 9 Publisher: Nature Publishing Group.

[12] F. Cincotti, F. Pichiorri, P. Aricò, F. Aloise, F. Leotta, F. de Vico Fallani, J. del R. Millán, M. Molinari, and D. Mattia. 2012. EEG-based Brain-Computer Interface to support post-stroke motor rehabilitation of the upper limb. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 4112–4115. https://doi.org/10.1109/EMBC.2012.6346871 ISSN: 1558-4615.

[13] David Coyle, James Moore, Per Kristensson, Paul Fletcher, and Alan Blackwell. 2012. I did that! Measuring Users' Experience of Agency in Their Own Actions. 2025–2034. https://doi.org/10.1145/2207676.2208350

[14] Mihaly Csikszentmihalyi. 1990. Flow: The Psychology of Optimal Experience.

[15] S de Vries and T Mulder. 2007. Motor imagery and stroke rehabilitation: a critical discussion. *Journal of Rehabilitation Medicine* 39, 1 (2007), 5–13. https://doi.org/10.2340/16501977-0020

[16] Ansgar E. Depping and Regan L. Mandryk. 2017. Why is This Happening to Me? How Player Attribution can Broaden our Understanding of Player Experience. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. Association for Computing Machinery, Denver, Colorado, USA, 1040–1052. https://doi.org/10.1145/3025453.3025648

[17] Ansgar E. Depping, Regan L. Mandryk, Chengzhao Li, Carl Gutwin, and Rodrigo Vicencio-Moreira. 2016. How Disclosing Skill Assistance Affects Play Experience in a Multiplayer First-Person Shooter Game. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, San Jose, California, USA, 3462–3472. https://doi.org/10.1145/2858036.2858156

[18] Nathan Evans, Steven Gale, Aaron Schurger, and Olaf Blanke. 2015. Visual Feedback Dominates the Sense of Agency for Brain-Machine Actions. (2015). https://doi.org/10.1371/journal.pone.0130019 Issue: 6 Library Catalog: infoscience.epfl.ch Number: ARTICLE Pages: e0130019 Publisher: Public Library of Science Volume: 10.

[19] Pouyan Fard and Moritz Grosse-Wentrup. 2014. The Influence of Decoding Accuracy on Perceived Control: A Simulated BCI Study. (Oct. 2014).

[20] Frank O. Adams Flemisch. 2003. *The H-Metaphor as a Guideline for Vehicle Automation and Interaction*. Technical Report. https://ntrs.nasa.gov/search.jsp?R=20040031835

[21] Kathrin Maria Gerling, Matthew Miller, Regan L. Mandryk, Max Valentin Birk, and Jan David Smeddinck. 2014. Effects of balancing for physical abilities on player performance, experience and self-esteem in exergames. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. Association for Computing Machinery, Toronto, Ontario, Canada, 2201–2210. https://doi.org/10.1145/2556288.2556963

[22] Samar M. Hatem, Geoffroy Saussez, Margaux della Faille, Vincent Prist, Xue Zhang, Delphine Dispa, and Yannick Bleyenheuft. 2016. Rehabilitation of Motor Function after Stroke: A Multiple Systematic Review Focused on Techniques to Stimulate Upper Extremity Recovery. *Frontiers in Human Neuroscience* 10 (2016). https://doi.org/10.3389/fnhum.2016.00442 Publisher: Frontiers.

[23] Guy Hawkins, Keith Nesbitt, and Scott Brown. 2012. Dynamic Difficulty Balancing for Cautious Players and Risk Takers. *International Journal of Computer Games Technology* 2012 (June 2012). https://doi.org/10.1155/2012/625476

[24] Hanneke Hooft van Huysduynen, Jacques Terken, and Berry Eggen. 2018. Why Disable the Autopilot? 247–257. https://doi.org/10.1145/3239060.3239063

[25] Daniel Kahneman, Barbara L. Fredrickson, Charles A. Schreiber, and Donald A. Redelmeier. 1993. When More Pain Is Preferred to Less: Adding a Better End. *Psychological Science* 4, 6 (Nov. 1993), 401–405. https://doi.org/10.1111/j.1467-9280.1993.tb00589.x Publisher: SAGE Publications Inc.

[26] Takahiro Kawabe, Warrick Roseboom, and Shin'ya Nishida. 2013. The sense of agency is action-effect causality perception based on cross-modal grouping. *Proceedings. Biological Sciences* 280, 1763 (July 2013), 20130991. https://doi.org/10.1098/rspb.2013.0991

[27] Margaret T. Lynn, Christopher C. Berger, Travis A. Riddle, and Ezequiel Morsella. 2010. Mind control? Creating illusory intentions through a phony brain–computer interface. *Consciousness and Cognition* 19, 4 (Dec. 2010), 1007–1012. https://doi.org/10.1016/j.concog.2010.05.007

[28] James W. Moore. 2016. What Is the Sense of Agency and Why Does it Matter? *Frontiers in Psychology* 7 (2016). https://doi.org/10.3389/fpsyg.2016.01272

[29] Giovanni Morone, Iolanda Pisotta, Floriana Pichiorri, Sonja Kleih, Stefano Paolucci, Marco Molinari, Febo Cincotti, Andrea Kübler, and Donatella Mattia. 2015. Proof of principle of a brain-computer interface approach to support poststroke arm rehabilitation in hospitalized patients: design, acceptability, and usability. *Archives of Physical Medicine and Rehabilitation* 96, 3 Suppl (March 2015), S71–78. https://doi.org/10.1016/j.apmr.2014.05.026

[30] Mark Mulder, David A. Abbink, and Erwin R. Boer. 2012. Sharing Control With Haptics: Seamless Driver Support From Manual to Automatic Control. *Human Factors* 54, 5 (Oct. 2012), 786–798. https://doi.org/10.1177/0018720812443984

[31] Gernot Müller-Putz, Reinhold Scherer, and Gert Pfurtscheller. 2007. Game-like training to learn single switch operated neuroprosthetic control. (Jan. 2007).

[32] Daniel Rakita, Bilge Mutlu, Michael Gleicher, and Laura M. Hiatt. 2018. Shared Dynamic Curves: A Shared-Control Telemanipulation Method for Motor Task Training. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction (HRI '18)*. Association for Computing Machinery, Chicago, IL, USA, 23–31. https://doi.org/10.1145/3171221.3171278

[33] Ander Ramos-Murguialday, Doris Broetz, Massimiliano Rea, Leonhard Läer, Özge Yilmaz, Fabricio L Brasil, Giulia Liberati, Marco R Curado, Eliana Garcia-Cossio, Alexandros Vyziotis, Woosang Cho, Manuel Agostini, Ernesto Soares, Surjo Soekadar, Andrea Caria, Leonardo G Cohen, and Niels Birbaumer. 2013. Brain-Machine-Interface in Chronic Stroke Rehabilitation: A Controlled Study. *Annals of neurology* 74, 1 (July 2013), 100–108. https://doi.org/10.1002/ana.23879

[34] Ander Ramos-Murguialday, Markus Schürholz, Vittorio Caggiano, Moritz Wildgruber, Andrea Caria, Eva Maria Hammer, Sebastian Halder, and Niels Birbaumer. 2012. Proprioceptive Feedback and Brain Computer Interface (BCI) Based Neuroprostheses. *PLoS ONE* 7, 10 (Oct. 2012). https://doi.org/10.1371/journal.pone.0047048

[35] Saif S. Rathore, Albert R. Hinn, Lawton S. Cooper, Herman A. Tyroler, and Wayne D. Rosamond. 2002. Characterization of incident stroke signs and symptoms: findings from the atherosclerosis risk in communities study. *Stroke* 33, 11 (Nov. 2002), 2718–2721. https://doi.org/10.1161/01.str.0000035286.87503.31

[36] Boris Reuderink. [n. d.]. Games and Brain-Computer Interfaces: The State of the Art. ([n. d.]), 11.

[37] Katja Rogers, Mark Colley, David Lehr, Julian Frommel, Marcel Walch, Lennart E. Nacke, and Michael Weber. 2018. KickAR: Exploring Game Balancing Through Boosts and Handicaps in Augmented Reality Table Football. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, Montreal QC, Canada, 1–12. https://doi.org/10.1145/3173574.3173740

[38] Ingeborg Goll Rossau, Jedrzej Jacek Czapla, Mózes Adorján Mikó, and Rasmus Bugge Skammelsen. [n. d.]. Examining the Effects of Sham Feedback on User Frustration in a Brain-Computer Interface Game. ([n. d.]), 14.

[39] Sofia Seinfeld, Tiare Feuchtner, Antonella Maselli, and Jörg Müller. 2020. User Representations in Human-Computer Interaction. *Human-Computer Interaction* (Feb. 2020), 1–39. https://doi.org/10.1080/07370024.2020.1724790

[40] M. Shaw. 1986. An input-output model of interactive systems. *ACM SIGCHI Bulletin* 17, 4 (April 1986), 261–273. https://doi.org/10.1145/22339.22382

[41] Xiaokang Shu, Shugeng Chen, Lin Yao, Xinjun Sheng, Dingguo Zhang, Ning Jiang, Jie Jia, and Xiangyang Zhu. 2018. Fast Recognition of BCI-Inefficient Users Using Physiological Features from EEG Signals: A Screening Study of Stroke Patients. *Frontiers in Neuroscience* 12 (Feb. 2018). https://doi.org/10.3389/fnins.2018.00093

[42] Jan D. Smeddinck, Regan L. Mandryk, Max V. Birk, Kathrin M. Gerling, Dietrich Barsilowski, and Rainer Malaka. 2016. How to Present Game Difficulty Choices? Exploring the Impact on Player Experience. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, San Jose, California, USA, 5595–5607. https://doi.org/10.1145/2858036.2858574

[43] Eleanor Stevens, Eva Emmet, Yanzhong Wang, Christopher McKevitt, and Charles Wolfe. 2017. *The Burden of Stroke in Europe: The challenges for policy makers*. Technical Report. King's College London for the Stroke Alliance for Europe (SAFE). 32 pages. http://www.strokeeurope.eu/downloads/The_Burden_of_Stroke_in_Europe_-_Challenges_for_policy_makers.pdf

[44] Philipp Sykownik, Katharina Emmerich, and Maic Masuch. 2018. Exploring Patterns of Shared Control in Digital Multiplayer Games. In *Advances in Computer Entertainment Technology (Lecture Notes in Computer Science)*, Adrian David Cheok, Masahiko Inami, and Teresa Romão (Eds.). Springer International Publishing, Cham, 847–867. https://doi.org/10.1007/978-3-319-76270-8_57

[45] Bram van de Laar, Danny Plass-Oude Bos, Boris Reuderink, Mannes Poel, and Anton Nijholt. 2013. How Much Control Is Enough? Influence of Unreliable Input on User Experience. *IEEE Transactions on Cybernetics* 43, 6 (Dec. 2013), 1584–1592. https://doi.org/10.1109/TCYB.2013.2282279 Conference Name: IEEE Transactions on Cybernetics.

[46] Rodrigo Vicencio-Moreira, Regan Mandryk, Carl Gutwin, and Scott Bateman. 2014. The effectiveness (or lack thereof) of aim-assist techniques in first-person shooter games. *Conference on Human Factors in Computing Systems - Proceedings* (April 2014). https://doi.org/10.1145/2556288.2557308

[47] Rutger Vlek, Jan-Philip van Acken, Evine Beursken, Linsey Roijendijk, and Pim Haselager. 2014. BCI and a User's Judgment of Agency. In *Brain-Computer-Interfaces in their ethical, social and cultural contexts*, Gerd Grübler and Elisabeth Hildt (Eds.). Springer Netherlands, Dordrecht, 193–202. https://doi.org/10.1007/978-94-017-8996-7_16

[48] Pan Wang, Gerald Choon-Huat Koh, Christian Gilles Boucharenc, and Ching-Chiuan Yen. 2017. Designing Two-player Competitive Games for the Rehabilitation of Upper-Limb Motor Function after Stroke. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '17*. ACM Press, Denver, Colorado, USA, 2201–2209. https://doi.org/10.1145/3027063.3053069

[49] Daniel M. Wegner and Thalia Wheatley. 1999. Apparent mental causation: Sources of the experience of will. *American Psychologist* 54, 7 (1999), 480. https://doi.org/10.1037/0003-066X.54.7.480 Publisher: US: American Psychological Association.

[50] Andéol Évain, Ferran Argelaguet, Anthony Strock, Nicolas Roussel, Géry Casiez, and Anatole Lécuyer. 2016. Influence of Error Rate on Frustration of BCI Users. In *Proceedings of the International Working Conference on Advanced Visual Interfaces - AVI '16*. ACM Press, Bari, Italy, 248–251. https://doi.org/10.1145/2909132.2909278