Measuring the Impact of Reinforcement Learning on an Electrooculography-only Computer Game

João Perdiz
ISR – Coimbra
University of Coimbra
Coimbra, Portugal
joao.perdiz@isr.uc.pt

Luís Garrote
ISR – Coimbra
University of Coimbra
Coimbra, Portugal
garrote@isr.uc.pt

Gabriel Pires
Department of Engineering
Polytechnic Institute of Tomar
Tomar, Portugal
gpires@isr.uc.pt

Urbano J. Nunes
DEEC
University of Coimbra
Coimbra, Portugal
urbano@deec.uc.pt

***Abstract*—In this paper we present an approach for detecting ocular movements, based on Electrooculographic (EOG) signals, that can have applications requiring the detection of ocular events such as saccades and blinks. We use it to implement an interactive go-kart game in which the user's goal is to avoid obstacles. Since horizontal saccades are the most representative of ocular movements, we use them as the main input for driving the kart. Eye blinking is a semi-autonomic and essential function that occurs naturally, so we decided to take advantage of it by using it as a secondary input to control the speed of the kart.**

This interface allows us to test the influence of machine learning techniques on game operation by inexperienced users and to evaluate whether it has a subjective positive impact. Two different versions of the game were implemented, one with a Reinforcement Learning Algorithm (RLA) that moderates users' commands based on outcomes of past commands, trying to prevent collisions, and a version with direct control (without RLA). Five participants tested the two versions of the game, so that we could compare the player's performance and engagement. We obtained promising results that show an improvement in score when RL is applied. We also found that players do not experience significant changes in gameplay feeling when RL is introduced.

***Index Terms*—Reinforcement Learning, Saccade Detection, EOG, HMI, Serious Games**

I. Introduction

The use of ocular signals as indicators of a person's attention relating to its gaze direction go back to the association made between biosignals and a person's physiology [1]. Electrooculographic (EOG) signals are generated by the eyes' behavior as dipoles, a behavior that originates measurable potential changes when the electrical fields of those dipoles are rotated with the eyeballs. Ocular gaze is a good indicator of a person's activity [2], readily detectable, and differs from other biosignals sources in that it is the result of both conscious – saccades, fast ocular movements – and unconscious eye movements; here only the former's detection is concerned.

EOG signals have been used for a wide range of experimental applications. Some aim at the improvement of the quality of life of motor-disabled people; this is the case of Human-Computer Interfaces (HCIs) that use EOG-based gaze for wheelchair driving [3], remote control of a television [4], or use of a virtual keyboard [5]. It has also been used for detecting human activity through ocular activity, either in an office-like setting [6] or in everyday activities [7].

In addition to those applications, EOG has also given promising indications with regard to the area of computer games. An example of an EOG-controlled game is found in [8], where four EOG channels are employed to detect motion in one of four ocular gaze directions (up, down, left and right). Although there are only a few gaming applications addressing the use of EOG, it is not uncommon to find games exploring the use of other biosignals, more specifically electroencephalographic (EEG) signals [9] through, for example, the use of P300 event-related potentials [10], steady-state visually evoked potentials [11], or through the combination of sensorimotor rhythms with the P300 signal [12]. EOG and EEG control commands share similar features in their discrete nature and their general unreliability, although their gaming paradigms can be significantly different. As these games are made to be played so differently from regular video games, one of the main concerns of researchers is to improve their playability. One approach is to adapt some aspects of the game to more closely match a specific player's expectations – a challenging task given the limitations of EOG or EEG signals, and the ways in which the gaming paradigm must be designed around those limitations. An effort to make the game more user-friendly might improve the game flow for people controlling biosignal-driven games for the first time, and for players who might become disappointed by games' playability limitations.

One of the methods through which gaming experience can be adapted to a particular user is Reinforcement Learning (RL). RL modifies a machine's behavior in face of previous actions by the machine's agent and the results (rewards) derived thereof. The behavior is modified in regard to the agent's forthcoming actions with the goal of maximizing the machine's predefined metric for global return. RL has been used in serious gaming interfaces in [13], [14]; however, in both cases the type of signals employed are muscular signals applied in the context of rehabilitation robotics.

In this work we describe a proposal for a game where ocular movements detected from EOG are used to issue commands to drive a go-kart. We explore the possibilities offered by

This work was partially funded by FCT under grants SFRH/BD/104985/2014 and SFRH/BD/88459/2012 and was developed at the Institute of Systems and Robotics – University of Coimbra. This work was partially supported by Project B-RELIABLE: SAICT/30935/2017, funded by FEDER/FNR/OE through PO CENTRO2020 and FCT.

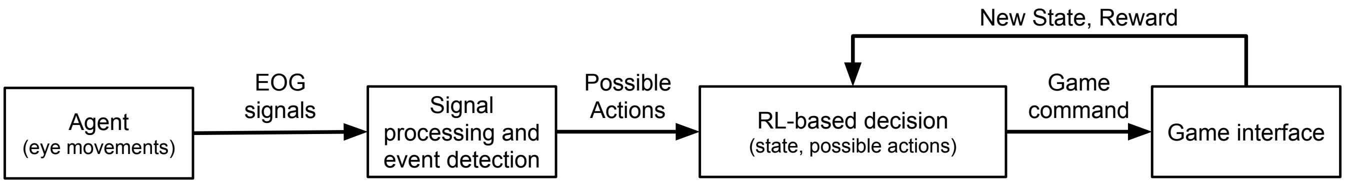

Fig. 1. System architecture showing the three modules used in implementing the game. The first module detects eye movements from the user's EOG signals. It feeds the second model a list of possible actions related to the ocular activity detected. One of these actions is selected using Q-learning RL after updating the Action-State pair values in the Q matrix with the reward from the user's previous detected action, and issued to the game interface. Only state changes triggered by user actions can be accountable for updating the Q matrix.

Reinforcement Learning in adapting the detection algorithm's behavior to suit the specificity of eye movements of different users. A general pipeline of our system can be seen in Figure 1. We have structured this paper in four main sections beyond this Introduction. In Section II we will present some related works in the field of games driven by biosignals. We discuss the reasons their authors gave for their implementation as well as their usefulness in the context of our own work. In Section III we describe the gaming paradigm and physical setup. Section IV examines the methods deployed for online saccade and blink detection, and provides the framework upon which the Reinforcement Learning algorithm is implemented. Finally, in Section V we describe and discuss the experimental results.

II. Biosignal-based games: Related Work

Most biosignal-driven games implement stationary games, which are meant to be played in a static setting, i.e. in front of the interface. An example can be found in [15], where the game's challenge consists in drawing a set of geometric figures at a time with an increasing level of difficulty. EOG electrodes are mounted on a wearable goggle-like device, with horizontal and vertical EOG signals being used to attain very high accuracy rates for all participants – a global average of 91%. In [8] a system was devised that aims for classification of four types of ocular movement – up, down, left, and right – in order to control an EOG-based variant of the game "Dance Dance Revolution". The authors present an EOG platform for the detection of quick saccadic actions, which can detect eye movement events separated by as little as $500ms$. Researchers in [16] use a baseball game to distinguish between saccades directed at nine different positions on a screen (a 3×3 grid) and blinks.

This short review of biosignal-based games shows us that, although some systems already detect user intention with a high degree of reliability, their ability to adapt to each user's distinctive playing characteristics is rather limited. This fact motivated us to develop our go-kart game using an RL algorithm so that the game acquires a certain degree of adaptation to the player.

III. Setup and playing paradigm

A. Game operation and interface

The proposed game was designed specifically for EOG-based control. The game approach is divided into two main components: the game engine/architecture and the user's input interface. The game contains a controllable agent (the go-kart) which can perform multiple combinations of left and right motions in order to avoid obstacles (Fig. 2). Obstacles are generated at a predefined rate but their lateral position on the road is randomly assigned. Collisions are detected with an axis-aligned bounding box chain between the obstacles and the agent. In case of collision a penalizing score is incremented, and in the end of the run that score is compared to the total number of obstacles to provide a global success rate. The scenario consists in a 3D infinite scrolling model where the central road is the drivable region for the agent. The game was developed using C++/QT and uses OpenGL to draw the game at a frame rate of $\approx 24Hz$. The game was configured so as to allow great latitude in adjusting both the go-kart's speed – from a fixed base value – as well as the frequency with which obstacles appear in the go-kart's path, thus enabling the setting of several levels of difficulty.

The user interface consists of the screen on which the go-kart is shown progressing along the road. From a user's perspective, the go-kart can move to its left and right directions, and must be steered to avoid obstacles along the way. This is done by executing leftward and rightward saccades that must be followed by a return of the eyes to a central position, a move which enables detection of a new command. In our game there are no visual cues for the user to change direction, as saccades do not require visual stimuli to be carried out by a person. It is therefore the user who decides when to deviate from obstacles. If a more considerable lateral detour is required, the user has to perform more than one saccade in a certain direction, which should be planned in advance. EOG signals are recorded with a g.tec gUSBamp amplifier to a Matlab/Simulink environment which processes and classifies the EOG signals. Then, the generated commands are sent via TCP/IP to the application running the game's visualization component, acting as TCP/IP server.

As maintaining eyes open during long periods of time can be difficult and inadvisable to the user, we account for blinks, which serve the additional purpose of speeding up or slowing down the go-kart in its path, allowing the user to regulate the game's pace, thereby increasing its playability. A single blink decreases the go-kart's speed along the road, and a double blink increases it by a fixed percentage relative to the starting speed. Blinks cause well-defined spikes in both horizontal and

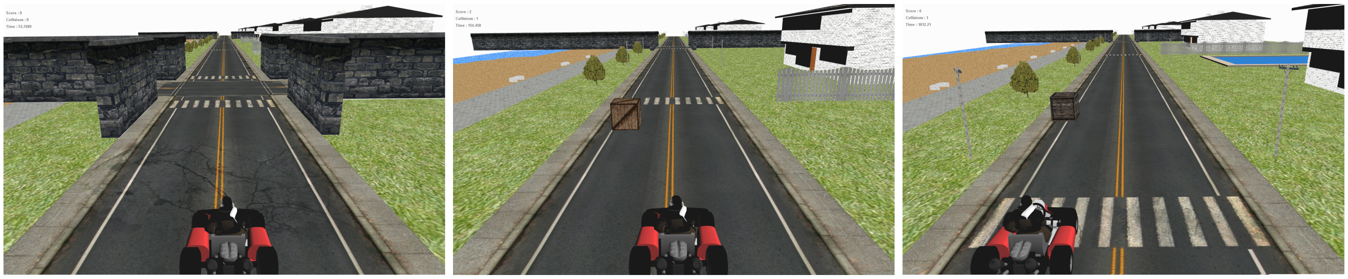

Fig. 2. A group of three snapshots of the gaming environment, with the starting position of the go-kart shown on the leftmost. The speed of the kart can be controlled through blinks, and the user must do saccades in order to deviate the go-kart from obstacles that appear unexpectedly on the road.

vertical EOG signals, as can be seen in Fig. 6. They can be detected in a straightforward fashion, but their similar impact on the horizontal and vertical EOG potentials means they can be mistaken for both vertical and horizontal saccades. Employing blinks as input commands minimizes the effects of their detection during saccades while providing a new type of input to the user.

IV. Methods

A. Signal processing and event detection

EOG signals are acquired from bipolar electrodes for the vertical and horizontal positions. Five skin-contact electrodes are applied on the user; four of them around the eyes and one in the forehead for grounding (Fig. 3). Before a game session, the user performs a short calibration session. The user is asked to perform horizontal saccades in six different amplitudes or to blink, using a calibration interface different from the game's. Recorded EOG data is then used to train the algorithm, and to extract the six horizontal and two vertical saccadic thresholds that are subsequently used in Algorithm 1 to detect saccadic onset during playing. Horizontal saccades are divided in three different amplitude classes in each direction so as to train the "weak", "medium", and "strong" saccade detectors. The EOG signals are sampled at a rate of $256Hz$ and low-pass filtered with a fourth-order Butterworth filter with cutoff frequency set to $30Hz$.

Saccades are detected through a decision rule acting upon a sliding window that is 256 samples long, comprising 1 second of EOG data, and sliding 16 samples between each run of the detection algorithms. First, the algorithm tries to detect a blink. This is done by finding the vertical signal's maximum value and the first-derivative peaks that must be present around the signal's peak value, as can be seen in Fig. 6; if they are present, and if the signal's peak is higher than a calibration threshold in relation to the averages on both its left and right sides, it is considered a blink. Otherwise, the horizontal EOG signal window is considered for detection of a horizontal saccade. For this, the first derivative of the horizontal signal is obtained. The signal window is a candidate for detection only if the first-derivative's maximum lies in the central portion of the window comprising 32 samples (samples $[112; 143]$). Afterwards, if either a saccade or a blink is detected, the algorithm is prevented from triggering again for a fixed amount of time. Onset detection of saccades and blinks is briefly summarized in Algorithm 1. If no ocular events are detected the sliding window is moved 16 samples, or $\frac{1}{16}s$, forward for the next run of Algorithm 1.

If any one of the aforementioned conditions are met, obtaining a command from EOG data is quite straightforward. The procedure in Algorithm 1 is used to calculate horizontal saccade amplitude and compare it to the calibration's thresholds. Six amplitudes are obtained from horizontal saccades executed during calibration, for three different saccades in each direction, and each of these generates an amplitude threshold; direction of online saccades can thus be directly determined. As for blinks, the algorithm cannot issue a command for a given amount of time in expectation of the possibility of a double blink, which results into a completely opposite command as that of a single blink.

Algorithm 1 Horizontal saccade and blink detection algorithm

Take Horizontal and Vertical EOG signal windows, X_H and X_V, 256 samples each at $256Hz$
Use horizontal saccadic thresholds Th_H (6 elements) and left and right thresholds for vertical peaks, Th_{Vleft} and Th_{Vright}
if $\max(X_V) - \overline{X_V}[t-100; t-50] \geq Th_{Vleft} \wedge \max(X_V) - \overline{X_V}[t+50; t+100] \geq Th_{Vright}$ **then**
  Accept window as a blink
else if $\max(X'_H)$ is located in the central $12,5\%$ of points in X'_H, its location being t **then**
  Calculate Amplitude $A_H = \overline{X_H}[t+50; t+100] - \overline{X_H}[t-100; t-50]$
  if $A_H \geq \min(Th_H)$ **then**
    Accept window as a horizontal saccade
  end if
end if

As we only intend to extract commands from out-of-center saccades, it becomes important to determine whether an ocular movement is coming to or from the gaze's center – the point right in front of the player. For such purpose we take the current saccade's direction from its amplitude and compare it to the direction of the previously detected saccade. Given that multiple saccades can be made in the same direction, only a saccade in an opposite direction of the previous one is to be

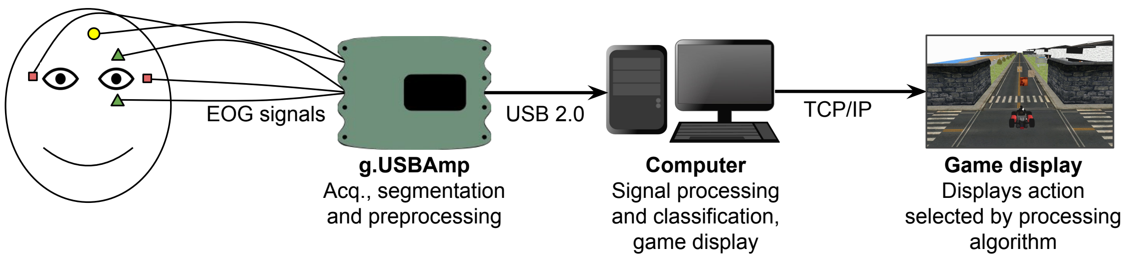

Fig. 3. A simplified diagram of the setup, with data flowing from left to right. Horizontal and vertical EOG signals are acquired from bipolar electrodes placed around the user's eyes and grouped as shown (squares – horizontal; triangular – vertical). Signal windows are pre-processed by the gUSBAmp bioamplifier, and sent to the detection algorithm for processing and classification. Any detected events are translated into game commands by a decision rule and sent to the game's display application via TCP/IP.

considered a "centering" saccade, in which the user returned its eyes to the center of the screen.

In Table I we see the events that can be extracted from user actions. Signal processing can assign each ocular movement by the user to one of six classes, whether they be "weak" or "strong" rightward or leftward saccades or blinks (single or double). The "strength" of a saccade determines, to a point, the "strength" of a lateral swipe of the go-kart in the game, which is a reflection of the kart's lateral speed applied during a fixed amount of time.

B. RL-based decision

In order to understand how RL can be applied to our game, it is important to note that RL cannot be the sole driver of decisions sent to the game for the go-kart to execute. RL intermediates between possible user actions and game actuation instructions, and tries to adapt its response to the former in order to maximize the outcomes of the latter. The actions that can be selected by Algorithm 1 are shown in Table I. From there each of them is mapped to a set of possible game commands that can be issued as a consequence, as shown in Table III (for example, a weak leftward saccade can originate a weak steering command – Actions 4 and 7 on Table III –, a normal steering command, or no steering command at all). The mode in which this choice is made can or not be influenced by the RL algorithm. The mapping is shown in Fig. 5.

The Q-learning algorithm chooses which actions to order based on a greedy choice paradigm. This is done using the Q-learning method, which involves updating the RL algorithm's Q matrix with the reward values from the last action that resulted from a user command (which are tied to that action's results in avoiding, or not, a collision with an obstacle).

TABLE I
DISCRETIZATION OF EVENTS ORIGINATED BY USER ACTION.

Event	Originated by
1	"Strong" leftward saccade
2	"Weak" leftward saccade
3	"Strong" rightward saccade
4	"Weak" rightward saccade
5	Single Blink
6	Double Blink

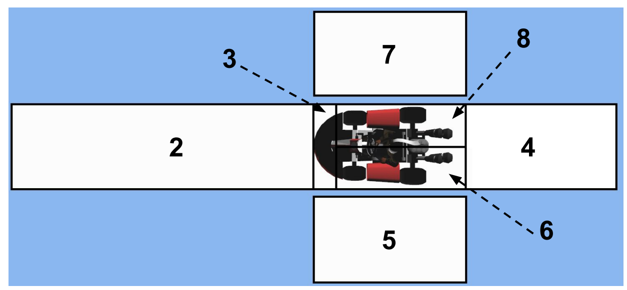

Fig. 4. The numbered regions specify where an obstacle must be in relation to the go-kart in order for state x to be declared. State 1 occurs when obstacles are in any of the shaded areas not belonging to any other state. State 9 happens when both States 6 and 8 would be observed (these, and 3, happen when a collision happens, i.e. when the kart and obstacle bounding boxes overlap). The occurrence of States 5 and 7 points to an imminent lateral collision, as well as its side.

The Q matrix is initialized with valuations of every possible Action-State pair that can happen in the game (Table II). While the possible Action is selected based on detection of user eye movements, States must be thoroughly defined before they are employed. The game state is defined by the go-kart's proximity to an obstacle, and whether it is colliding with it or not. We have defined nine states; a map of states can be seen in Fig. 4. Some states are non-obvious from the figure: State 1 occurs when obstacles are in front of the go-kart, further than those causing a State 2; States 3, 6, 8 and 9 occur when a collision happens, with the latter caused by an overlap of States 6 and 8.

TABLE II
INITIAL Q MATRIX FOR STATE-ACTION PAIRS.

		Actions								
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
States	**1**	0,5	1	0	0	-0,5	-1	0	-0,5	-1
	2	-1	-1,5	-0,5	0	0,5	1	0	0,5	1
	3	-1	-1,5	-1	0	0,5	1	0	0,5	1
	4	1	1	0	-0,5	-0,5	-0,5	-0,5	-0,5	-0,5
	5	0	0	0	0,5	1	1,5	-1	-1,5	-2
	6	-1	-1,5	-1	0,5	1	1,5	-1,5	-1,5	-2
	7	0	0	0	-1	-1,5	-2	0,5	1	1,5
	8	-1	-1,5	-1	-1,5	-1,5	-2	0,5	1	1,5
	9	0	0	-0,5	-1,5	-1,5	-1,5	-1,5	-1,5	-1,5

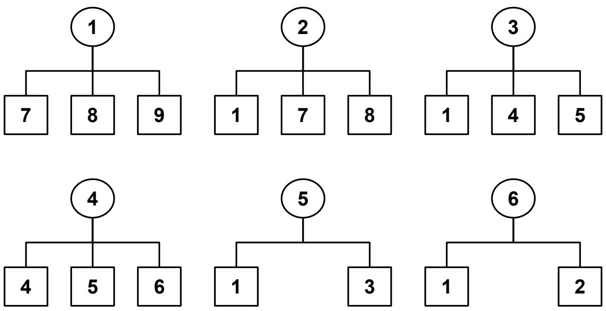

Fig. 5. Mapping of user-driven events (circles) to game commands (boxes) for the version of the game where Reinforcement Learning is employed. When RL is not used a single game command is selected outright.

The commands in Table III and the nine possible states of the kart are initialized in the Q matrix shown in Table II; this is altered by the RL procedure, the basic functioning of which is shown in Algorithm 2. This algorithm was adapted from [17]; its α and γ values were chosen for a quick adaptation of the State-Reward pairs to individual behavior patterns that will necessarily be dictated by a small number of actions.

Algorithm 2 Q-learning RL algorithm as implemented in our game, adapted from [17].

Initialize $Q(s,a), \forall s \in \mathcal{S}$, $R = -1, 1$, $\alpha = 0.8$, $\gamma = 0.9$
for Each state change caused by a command **do**
  Choose A from S using policy derived from Q (greedy)
  Take action A, observe reward R, new state S'
  Reward $R = 1$ for avoided collisions, and $R = -1$ for collisions occurred after action was taken
  Update $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma \max_a Q(S',a) - Q(S,a)]$
  $S \leftarrow S'$
end for

V. Validation and Experimental Results

In order to test what influence the Reinforcement Learning algorithm has on test participants, some of the game's parameters had to be fixed so that they would not influence the results across sessions. This meant using a single-participant trial to fine-tune the minimum detection interval and the go-kart's on-screen speed. Minimum detection intervals can be very reduced, and the single-participant trials showed that they can be shortened to as little as $187,5ms$. However, and bearing in mind that the system was intended to be used by inexperienced users, it was decided to extend this period to $500ms$. This period of time is halved if the detection is aimed at double blinks – a second blink can be detected just $250ms$ after the first. The need for a relatively long detection interval results from the need for a compromise between speed of detection and the necessity of distinguishing centering saccades from outward saccades – those originating game commands. The user must rest its eyes before doing a return saccade, otherwise the algorithm might not detect it and will not immediately allow new commands to be issued. Traces of the horizontal and vertical EOG signals for a typical horizontal saccade are shown in Fig. 7. Having set these parameters, we followed through to the game sessions.

TABLE III
Codes of command actions sent to the game.

Action	Effect
1	Stay on course, do not change speed
2	Stay on course, increase linear speed by 20%
3	Stay on course, decrease linear speed by 20%
4	Lateral swipe to right, at 80% of normal lateral speed
5	Lateral swipe to right, at 100% of normal lateral speed
6	Lateral swipe to right, at 120% of normal lateral speed
7	Lateral swipe to left, at 80% of normal lateral speed
8	Lateral swipe to left, at 100% of normal lateral speed
9	Lateral swipe to left, at 120% of normal lateral speed

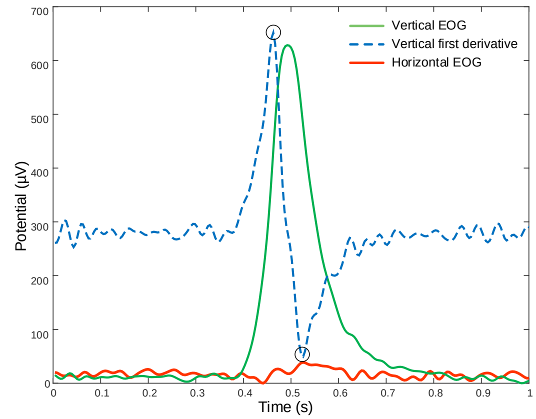

Fig. 6. The horizontal and vertical EOG $1s$ signal windows when the user blinks, with the first derivative of the vertical signal also shown. When a blink is detected by the algorithm only the vertical signal window is used for post-processing, preventing any further detection of saccades for the duration of the minimum detection interval. The vertical signal's transient amplitude peak is strong and short enough relative to its neighborhood to trigger the algorithm into detecting a blink. The dashed line is the vertical signal's first derivative, whose two peaks – shown circled – are essential for detecting a blink.

The testing phase consists of two game sessions. In the first session the RL algorithm is deactivated and commands are simply issued via a decision rule, whereas in the second session the RL algorithm active and moderates the players' decisions. The participant is not informed of the activation of the RL algorithm in each session. After the end of each session the participant is asked to fill in that session's questionnaires, which are used afterwards to compare their subjective evaluations of the RL and non-RL versions of the game.

A. Measuring game engagement

In order to assess a player's involvement with the game, we have chosen two different scales that let us know both the user's engagement with the game as well as the perceived level of effort to control the game. For game engagement we employed a scale known as the Game Engagement Questionnaire

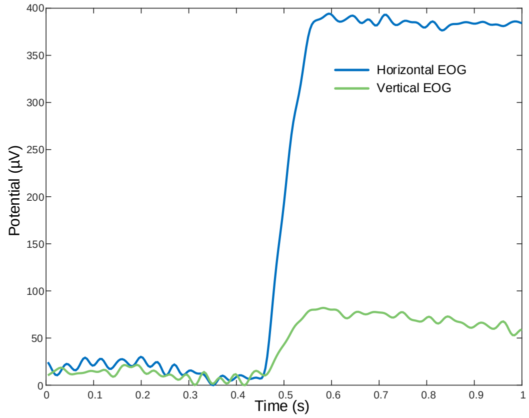

Fig. 7. Traces of a $1s$ signal window for horizontal and vertical EOG signals when a horizontal saccade is executed. When detected, this event results in a lateral command being sent to the go-kart.

or GEQ [18]. Although some of its items do not seamlessly integrate into the concept of our game, the participants were asked to respond to its entirety. In addition to the GEQ, users were also asked to respond to the NASA-Task Load Index survey (TLX) [19]. The purpose of this is to measure general acceptance of the game's working that do not directly correlate to its playability, such as the physical setup and the game interface's pacing.

Results of these questionnaires provide us with an important basis for assessing the success of the game with its players, and whether modifications to the game's behavior are subjectively perceived by them between sessions.

B. Comparing the two versions

Each participant played the game twice, each session lasting ten minutes. In the first session RL was deactivated, while in the second it was active. Each session lasted 10 minutes, which at normal kart speed allowed for a total of 120 obstacles to be met (users could modify the kart's speed while playing). We measured the percentage of successfully cleared obstacles, and what changes were made to the Q matrix when the RL algorithm was activated. Changes in the GEQ and TLX scores between sessions are reflected in the first two traces of Fig. 9. Average changes to TLX scores are shown in Table IV. It can be considered that perceived changes to the game's playability are not relevant, and that the intervention of the RL algorithm – whose existence is not disclosed to the participant – during the second session seems to have no discernible effect in those scores, and therefore no perceived subjective effect in gameplay. In the last trace of Fig. 9 we can see the changes in player accuracy (obstacle avoidance) between sessions.

In average, the RL-enabled session lowered collisions by 4.6 per participant, or 3.9 percentage points. It can be seen that for some participants the second session (RL-enabled) was more taxing than the first. Participant 2 in particular scored significantly less in the GEQ questions related to Flow and

TABLE IV
AVERAGED TLX SCORES FOR THE RL-ENABLED AND RL-DISABLED GAME SESSIONS ACROSS ALL PARTICIPANTS. SCALE IS 1-20; LOWER IS BETTER.

TLX item	Non-RL average	RL average
Mental load	6,8	7,0
Physical load	9,0	8,8
Temporal load	6,4	5,6
Performance	9,6	9,6
Effort	12,6	13,2
Frustration	6,8	7,4

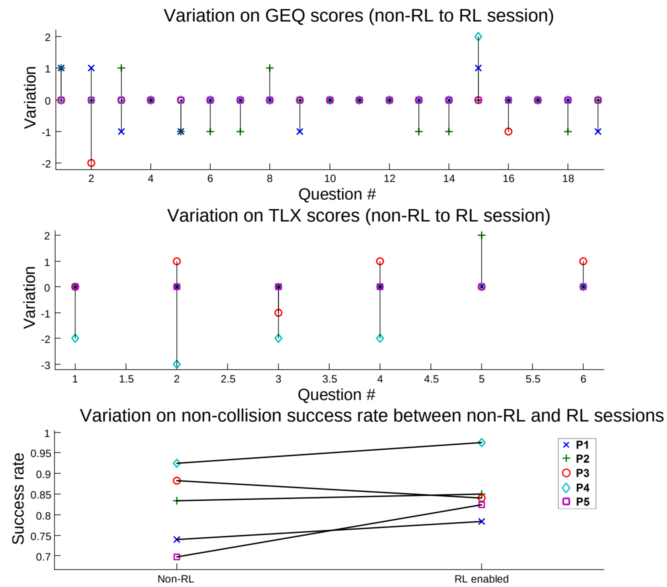

Fig. 9. Traces for changes in GEQ and TLX scores and obstacle avoidance success rate for each participant, between their normal and RL-enabled sessions. For only one of the five participants did the RL-enabled session not mean an increase in obstacle avoidance. As for effort, TLX responses for Participant 4 show significant decreases in the perceived physical and mental load (Items 1 and 2) while playing the RL-enabled version.

Immersion in the second session, which can be related to eye fatigue reported by the participant. Tiredness was also invoked by Participant 3 for the lower score on the second session.

When analyzing changes made to the Q matrix in the RL-activated session and the changes in "accuracy" rates from one session to another – the former are shown in Fig. 8 –, it was evident that RL was not performing as expected. Modifications to the Q matrix seem to be centered in State 1 for all players. This might mean that the vehicle was not being placed in pre-collision states from which the player could meaningfully deviate. As the learning only takes place if a state change has taken place as a result of an action, when the player made a saccade to deviate from an obstacle he would remain in State 1 during the whole command, and no positive reinforcement would be effected on Actions taken in other, more urgent states. We interpreted this issue as evidence that the go-kart was barely ever placed in State 2 before it was diverted from its course, because the player made the saccade before it entered State 2. It was then decided to extend State 2 to double its length in front of the kart (see Fig. 4). The temporal window in

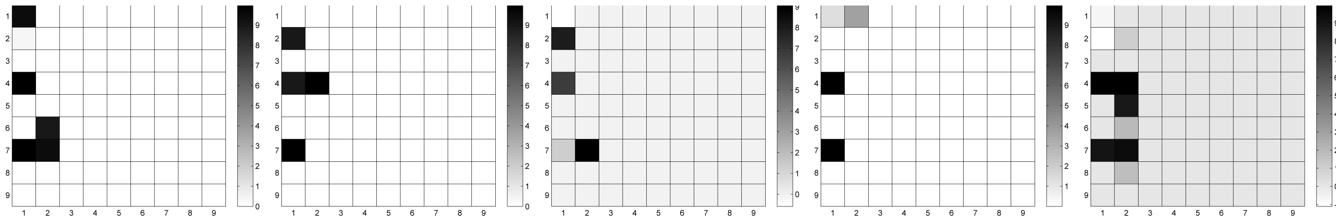

Fig. 8. Changes on each participant's Q-Matrix during the RL-moderated session, from Participant 1 on the left to Participant 5 on the right. States are on the x-axis and Actions are distributed along the y-axis of each matrix.

which an obstacle can place the kart in State 2 is then much larger, making learning much more probable. Participants 4 and 5, who played the game in this new configuration, show respectively the best collision-avoidance score and the largest improvement from Non-RL to RL sessions among all the participants.

From the graphs in Fig. 8 it can also be gleaned that the only non-neutral State the RL algorithm was interfering with was State 2. This just means that there were no collisions or intentional avoidance movements as the kart was passing by an obstacle, and that all diversions were made from a stance in which the obstacle was directly in front of the kart. This might be explained by the game's relatively low pace, which allowed for ample planning by the users in order to avoid obstacles seen beforehand. Nonetheless, it also emphasizes the point that people will look at obstacles in order to avoid them, which may make control of the game seem unnatural. This was found to be true for eye gaze related to obstacle-avoidance while driving [20], and based on our findings it also applies to the context of a game. This might make an initial approach to the game more difficult for new participants, especially if the difficulty level is increased; however, most of the participants expressed the capability of adapting to the game's paradigm rather quickly, and so this should not be as relevant a concern.

VI. Conclusions

In this paper we developed a gaming platform that could be operated using only EOG biosignals, and the adaptation of its responses to user commands through a Q-learning Reinforcement Learning algorithm. This algorithm moderates intensity of user commands given past experience. We have shown this approach can be used in the implementation of a game as a moderator of a player's decisions, and that this moderation is generally beneficial in the context of our EOG-controlled game. We expect the Reinforcement Learning algorithm to play a larger role if the game's pace is increased.

From the relative success of this implementation we conclude that the same approach can probably be taken when designing biosignal-controlled games with impaired people and Assistive Living in mind, making developments of this platform a suitable basis for this end.

References

[1] P. Ekman, R. W. Levenson, and W. V. Friesen, "Autonomic nervous system activity distinguishes among emotions," *Science*, vol. 221, no. 4616, pp. 1208–1210, 1983.

[2] E. Gu and N. I. Badler, "Visual Attention and Eye Gaze During Multiparty Conversations with Distractions," in *International Workshop on Intelligent Virtual Agents*. Springer, 2006.

[3] R. Barea, L. Boquete, M. Mazo, and E. López, "Wheelchair Guidance Strategies Using EOG," *Journal of intelligent and robotic systems*, vol. 34, no. 3, pp. 279–299, 2002.

[4] L. Y. Deng, C.-L. Hsu, T.-C. Lin, J.-S. Tuan, and S.-M. Chang, "EOG-based Human–Computer Interface system development," *Expert Systems with Applications*, vol. 37, no. 4, pp. 3337–3343, 2010.

[5] A. B. Usakli and S. Gurkan, "Design of a Novel Efficient Human–Computer Interface: An Electrooculagram Based Virtual Keyboard," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 8, pp. 2099–2108, 2010.

[6] A. Bulling, J. A. Ward, H. Gellersen, and G. Troster, "Eye Movement Analysis for Activity Recognition Using Electrooculography," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 741–753, 2011.

[7] A. Bulling, D. Roggen, and G. Tröster, "Wearable EOG goggles: Seamless sensing and context-awareness in everyday environments," *Journal of Ambient Intelligence and Smart Environments*, vol. 1, no. 2, pp. 157–171, 2009.

[8] M. R. Kim and G. Yoon, "Control Signal from EOG Analysis and its Application," *World Academy of Science, Engineering and Technology, International Journal of Electrical, Electronic Science and Engineering*, vol. 7, no. 10, pp. 830–834, 2013.

[9] B. Reuderink, "Games and Brain-Computer Interfaces: The State of the Art," *WP2 BrainGain Deliverable, HMI, University of Twente (September 2008)*, 2008.

[10] A. Y. Kaplan, S. L. Shishkin, I. P. Ganin, I. A. Basyul, and A. Y. Zhigalov, "Adapting the P300-based Brain–Computer Interface for Gaming: A Review," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 5, no. 2, pp. 141–149, 2013.

[11] R. Parafita, G. Pires, U. Nunes, and M. Castelo-Branco, "A spacecraft game controlled with a brain-computer interface using SSVEP with phase tagging," in *Serious Games and Applications for Health (SeGAH), 2013 IEEE 2nd International Conference on*, May 2013.

[12] G. Pires, M. Torres, N. Casaleiro, U. Nunes, and M. Castelo-Branco, "Playing Tetris with non-invasive BCI," in *Serious Games and Applications for Health (SeGAH), 2011 IEEE 1st International Conference on*. IEEE, 2011.

[13] K. Andrade, G. Fernandes, G. Caurin, A. Siqueira, R. Romero, and R. Pereira, "Dynamic Player Modelling in Serious Games Applied to Rehabilitation Robotics," in *Robotics: SBR-LARS Robotics Symposium and Robocontrol (SBR LARS Robocontrol), 2014 Joint Conference on*. IEEE, 2014.

[14] K. Tsiakas, M. Dagioglou, V. Karkaletsis, and F. Makedon, *Adaptive Robot Assisted Therapy Using Interactive Reinforcement Learning*. Cham: Springer International Publishing, 2016.

[15] A. Bulling, D. Roggen, and G. Tröster, "EyeMote – Towards Context-Aware Gaming Using Eye Movements Recorded from Wearable Electrooculography," *Fun and games*, pp. 33–45, 2008.

[16] C.-T. Lin, S.-L. Wu, W.-L. Jiang, J.-W. Liang, and S.-A. Chen, "A wireless Electrooculography-based human-computer interface for baseball game," in *Information, Communications and Signal Processing (ICICS) 2013 9th International Conference on*. IEEE, 2013.

[17] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 2011.

[18] J. H. Brockmyer, C. M. Fox, K. A. Curtiss, E. McBroom, K. M. Burkhart, and J. N. Pidruzny, "The development of the game engage-

ment questionnaire: A measure of engagement in video game-playing," *Journal of Experimental Social Psychology*, vol. 45, no. 4, pp. 624–634, 2009.

[19] S. G. Hart and L. E. Stavenland, "Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research," in *Human Mental Workload*, P. A. Hancock and N. Meshkati, Eds. Elsevier, 1988, ch. 7, pp. 139–183.

[20] L. Fletcher and A. Zelinsky, "Driver Inattention Detection based on Eye GazeRoad Event Correlation," *The International Journal of Robotics Research*, vol. 28, no. 6, pp. 774–801, 2009.