Improving Embodiment with Reinforcement Learning in Virtual Reality

Yawen Hou
School of Computer and Communication Sciences, EPFL, Switzerland

***Abstract*—This project explores the possiblity of implementing an online calibration algorithm that allows to find the subjective maximum distortion threshold for each subject without provoking a ”Break In Embodiment (BIE)” in order to offer an optimal VR experience.**

I. Litterature review

An ultimate VR immersive experience aims to make the subject undergo the same sensations in a virtual body as if they were in their biological body. An elementary step in reaching this goal is to provide a consistent Sense of Embodiment (SoE) to the subject. According to Kilteni et al.[1], the concept of SoE can be decomposed into three subcomponents: the sense of body ownership, agency, and self-location. Self-location is defined as one’s spatial experience of being inside the body; in other words, having a first-person view. Agency is described as having the motor activity control of the body, having the limbs move accordingly to the subject’s will. Body ownership refers to one’s self-attribution of a body, implying that the latter is the source of sensations. If one of the three conditions is disrupted, a ”Break In Embodiment” (BIE) defined in [2] will occur, giving subject to have the illusion to have lost control of their body.

To offer a seamless control of the virtual body and to avoid BIEs, previous studies have explored possible ways to calibrate the interactions of the subject with objects in the VR environment. Bovet et al.[3] and Debarba et al. [4] proposed to use distortion functions to adjust the avatar’s hand’s position regard to the virtual objects and the avatar itself. As shown by Bovet et al.[4], in the presence of haptic feedback, such as the touch of one’s body, any slight distortion would impede the self-attribution of the virtual body [3]. Subjects are quick to notice a gap between the virtual hand and the virtual body when they feel the touch of their real hand on their physical body and they feel disturbed by the gap. However, studies [2] [4] have shown that when relying solely on visual feedback, subjects can tolerate up to a certain level of distortion. They identify themselves as the owner of the avatar even if they notice that the movements of the avatar are deviated from their original physical movements, especially if the external force helps them to accomplish the task [2] [4]. In Porssut et al. [2], subjects are asked to perform a non-biological movement with their hand, which is not possible without the help of the external distortion. The results show that a high distortion value breaks the sense of embodiment, but a reasonable magnitude of deviation is generally accepted by the subjects [2].

However, a significant variance in the distortion threshold has been found between subjects in [2] [4] [3]. Therefore we need to adjust the distortion value for each subject. Previous studies have experimented with several methods, including the standard staircase and the Point of Subjective Equality (PSE) [4] [2] [5] to find this value. However, staircase requires an excessive amount of trials before converging toward a satisfactory value. Furthermore, PSE is an offline method. Thus, these algorithms might not be suited for a real-time application. There is also the issue that when the subject is progressively exposed to the distortion stimuli over a long period of time, they might become used to the discrepancy between the virtual limb and their real limb, therefore we might be able to apply a higher magnitude of distortion later on.

As the calibration of the distortion is completely subjective and needs to be adapted overtime, we need an adaptive algorithm that can quickly learn the correct threshold. Reinforcement Learning (RL) algorithms seem to be a perfect candidate to adjust in real-time the distortion value. Past studies using RL in robotics [6] [7] [8] [9] demonstrated that a single negative feedback (transmitted when the subject feels an error) is enough for certain RL algorithms to train a satisfying model. For example, Kim et al. [10] had their robot learning user-defined gestures and the association of each gesture with a predefined robot action using intrinsic RL techniques, and their results are promising [10]. Luo et al. [8] trained an RL agent to perform a binary-choice task in an online experiment using a single type of negative reward as feedback. Their average improvement in efficiency (compared to randomly making a choice) was 15.21%. In [7], the task was to have a robotic arm learn the position of the basket chosen by the user. The accuracy varies depending on the position of the chosen basket, but in general the system managed to learn the correct position.

The objective of the current study is to find the maximum magnitude of distortion that each subject can tolerate without provoking a BIE in a real-time VR application. This maximal value of distortion is defined as the self-attribution threshold [4] (also referred in the following paragraphs as distortion threshold). We will attempt to find this threshold with different algorithms. Then, the performance of each algorithms will be evaluated using the criteria specified in section IV.

After reviewing previous studies [8] [6] [7] [10] [9] and different RL methods [11], we choose to test the two implementations of Temporal Difference (TD(0)) learning: Q-learning, also used by Iturate et al. [7], and SARSA. We choose to not use the complex RL algorithms mentioned above because we only need to adjust a single parameter (the distortion gain), therefore it is overkill to use intrinsic, inverse or interactive RL algorithms for a one-dimensional problem. TD(0) is a model-free RL algorithm based on policy control, which updates the current state based on the estimate of the next state. The difference between the two implementations is that Q-learning is an off-policy method, which computes the Q-value (estimation of accumulated reward in each state) according to a greedy policy without having the agent following this greedy policy. SARSA is an on-policy method, which means that the Q-value is computed according to a predefined policy, and the agent would always follow that policy [11]. The RL agents will be trained with explicit feedback from the subject. At each trial, if the subject does not sense a BIE, the agent receives a positive reward; else, the agent would receive a negative reward.

In addition to implementing the RL algorithms, the staircase algorithm originally defined by Meese et al. [12] and used in [3] [4] will also be implemented as a baseline methods in order to measure the performance of RL based algorithms.

II. The Distortion Movement Task

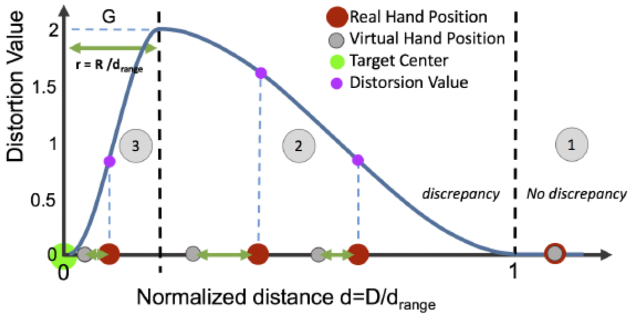

Fig. 1: Overview of the well-shaped distortion function. No distortion is applied when d is greater than 1, i.e. the virtual hand is placed exactly where the real hand is situated (region 1). When d is below 1, the virtual hand starts to get attracted to the target (its position is closer to the target than the position real hand of the subject). The attraction amplitude increases as d decreases from 1 to r (region 2), then diminishes to zero as d decreases from r to 0 (region 3).

We use the same distortion function as in Porssut et al.[2]. The design of the distortion function is to help the subject reach to a moving target while preserving their sense of agency (i.e. the subject feels that they still own the movement). The avatar's hand is first attracted towards the target until it reaches the outer boundary of the moving target. Then, once the virtual hand is inside the moving target, the attraction is progressively reduced to zero until the hand arrives at the target center. There are three input parameters to the distortion function. R is the radius of the moving target (tennis ball). d_{range} is the distance range of the attraction force centered on the moving target, which corresponds to the radius (0.35m) of the circular trajectory the target follows (sec. C.). At last, G (referred as the "distortion gain" or "gain" in the following paragraphs) is the maximum amplitude of the attraction (G = 2 in fig. 1).

We are interested in finding the optimal G for a predefined d_{range} and R. Initially, we set the range of distortion gain to [0, 2] (like in [2]) and used this range to test the RL algorithms (sec. A., sec. B.). Due to the fact that 4 subjects in [2] had a distortion threshold above 2, we increased the maximal gain to 4. However, with the pilot study's results, we realized that 4 was still not high enough. Several participants did not notice that their movement was manipulated, even though the maximal distortion gain of 4 was applied. By definition, all participants must notice the maximal distortion. Thus, we increased the magnitude of the maximal gain to 10. At last, we used the following discrete values of distortion gain for the experiment: $\{0, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, 2.75, 3, 4, 5, 7, 10\}$. The last values cover a larger range because we observed during the pilot study that as the gain increases, it becomes harder for the subject to detect a difference between two distorted movements.

For the task of this experiment, we used a modified version of the task used in Porssut et al. [2]. In their experiment, they asked the subjects to perform a non-biological movement to detect if they perceive that the movement was actually non-feasible without the help of the distortion. It consists on pursuing a sphere moving on an elliptic trajectory. The subject first put the tennis ball they are holding inside the target sphere at rest at its initial position. After a short countdown, the sphere departs and the subject needs to catch it up. The sphere accelerates in high curvature region and decelerates in low curvature regions. These changes in speed made the movement non-biological.

In our case, instead of making the subject pursue immediately the target sphere as soon as it starts moving, we let them decide when to move their hand to join the moving target. We chose a two-phase movement that uses two separate spheres (sec. C.). First, the subject needs to place the tennis ball inside a blue sphere (first target) situated at some distance (calibrated according to the length of their arm) in front of their chest. Then, the green sphere (second target) starts to move following a circular trajectory of radius 0.35m, at a constant speed (biological movement). The subject chooses the best moment to move their hand from the blue sphere to join the green sphere. We made these modifications because we wanted the distortion to be applied only when the subject decides to start moving. We want to avoid provoking a BIE caused by any other reason than the distortion. In the original experiment, the subject's hand might be attracted to the target before they start to pursue it, due to a possible delay between the moving sphere and the avatar's hand shortly when the target starts to move. This could possibly provoke a BIE.

During the pilot study, we have set a smaller radius for this circular trajectory. However, several participants reported not having enough time to observe the movement between the initial target (blue sphere) and the moving target (green sphere). Thus, we have increased the radius to 0.35m to leave more observation time.

III. Reinforcement learning

Standard RL problems can be modelled as a Markov Decision Process (MDP) defined by the tuple { S, A, T, r, γ}, where S represents the state space, A represents the action space, $T : S \times A \rightarrow S$ represents the transition probabilities from a state s to the next state s' by taking the action a. $r : S \times A \rightarrow R$ is the reward function that returns a reward to the RL agent each time it takes an action a in a state s. Finally, $\gamma \in [0, 1]$ is the discount factor. The goal is to find an optimal policy $\pi : S \times A \rightarrow R$ that maximizes the accumulated reward $R_t = \sum_{t=0}^{\infty} \gamma^t r^{t+1}$.

Normally, the reward function is unknown to the RL agent, and it should learn the best policy through its interactions with the environment. To test the performances of the algorithms, an artificial reward function has been designed as follows: when the distortion value is equal or below the subject's threshold, they will not notice the distortion in their movements, which yields a positive reward. Once the distortion value exceeds their threshold, the subject will experience a Break in Embodiment (BIE), which yields a negative reward. The absolute value of the reward is the distortion gain. For instance, if the subject performs the task with a distortion gain of 2.5 and does not notice the distortion, then the RL agent receives a reward of 2.5. On the contrary, if the subject feels a BIE, then the RL agent receives a negative reward of -2.5. In the ideal situation, the threshold of the subject stays stable, thus this reward function will entice the RL agent to learn the highest possible gain, since we want to find how much distortion can the subject tolerates.

A. *Q-learning and SARSA*

Following [7], the problem of finding the best threshold has first been modelled as a Markov Decision Process (MDP) and we attempted to solve it with Q-learning and SARSA. In the MDP scheme, a state is defined as the interval to which belongs the current distortion value. The initial range of distortion gain was defined as $[0, 2]$, taken from Porssut et al. [2] (sec. II) and was discretized into 8 intervals $[0, 0.25), [0.25, 0.5), ...[1.5, 1.75), [1.75, 2)$. The actions are defined as increasing or diminishing the gain by steps of 0.5 or 0.25. Under this design, Qlearning and SARSA took more than 1000 iterations to converge, because the combination of the action space and the state space is too large to be fully explored in a few numbers of iterations. To simplify the model design, we decided to characterise the problem as a non-stationary multi-armed bandit.

B. *Non-Stationary Multi-Armed Bandit (MAB)*

The k-armed bandit omits the state-space and the transition probabilities and focuses solely on finding the best action from a total of k choices to maximize the total rewards over a predefined period of time. Figure 2 depicts the reward distribution of a possible 10-armed bandit problem. In this case, action 3 gives the best reward, and should be the action that is chosen at each time to maximize the reward obtained over the predefined period of time.

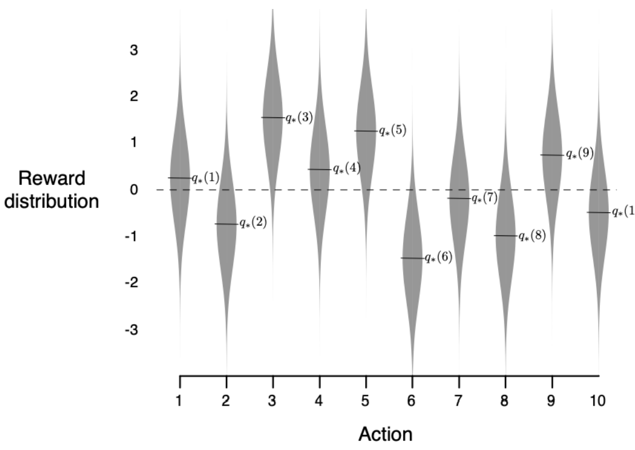

Fig. 2: Example of a stationary 10-armed bandit. The best action is 3.

For the experiment, the k actions are the distortion values are discretized as follows: {0, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, 2.75, 3, 4, 5, 7, 10}[1]. The agent receives a positive or a negative reward chosen from a non-stationary probability distribution (i.e. the reaction of the subject to the same distortion value might change over time) depending on the reaction of the subject to the selected gain. We find the threshold by maximizing the expected reward, namely the value function, conditioned by the action: $Q_*(a) = E[R_t|A_t = a]$ where R_t is the reward at time t and A_t is the action chosen at time t. The optimal distortion threshold (action) will be the value with the highest Q-value: $A_{opt} = \arg\max_a Q_*(a)$. As the Q-value of each action ($Q_*(a)$) is unknown at the beginning, we would update an estimate of $Q_*(a)$ for each action at each time step. As the problem is non-stationary, the update rule at time step t is : $Q_{t+1} = Q_t + \alpha[R_t - Q_t]$ where α is the learning rate that diminishes over time.

a. *Experimentation with 3 RL algorithms*

We tested three different algorithms to solve the non-stationary MAB: a simple ϵ-greedy algorithm; Upper-Confidence-Bound combined to ϵ-greedy algorithm; and finally the policy gradient algorithm. These 3 algorithms are originally tested on a 8-armed bandit problem, each arm corresponding to one of the following distortion gains: $\{0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2\}$ [2]. Each of them are evaluated with their cumulative rewards over 2000 steps. The higher the cumulative rewards, the better the algorithm. As we would like the algorithm to converge within the fewest steps possible, we also plotted a close-up graph (fig. 4) for the first

[1] Originally, the range of the distortion gain is [0, 2], but it has been changed to [0, 10] for the final experimentation. See section II for more details

[2] The final values are : {0, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, 2.75, 3, 4, 5, 7, 10 } (sec. II). Even though the number of arms increased, this does not change the fact that UCB's performance still dominates the two other algorithms.

150 steps for an evaluation of the short-term performance of the algorithms..

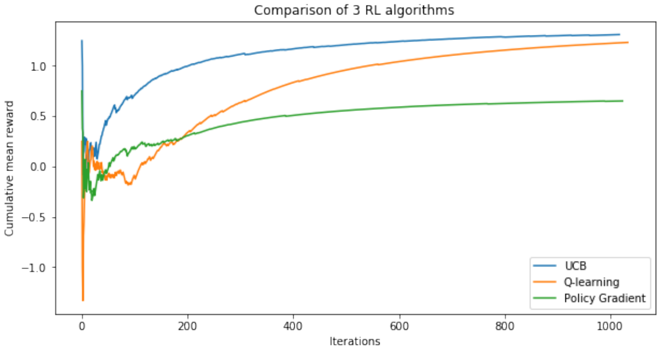

Fig. 3: Accumulated rewards of 3 algorithms (2000 steps)

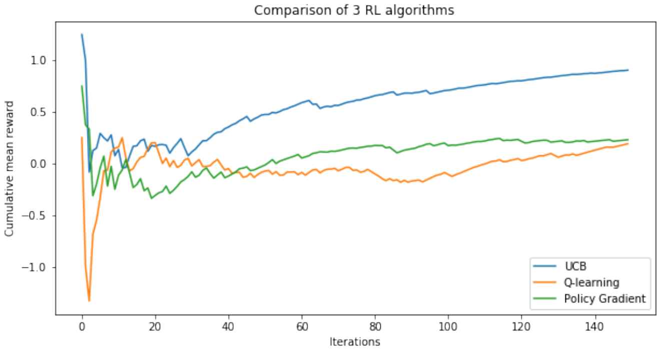

Fig. 4: Accumulated rewards of 3 algorithms (150 steps)

From the close-up graph 4 and the full graph 3, we can see that UCB outperforms the two other algorithms. We thus choose UCB as the algorithm to be implemented for this experiment.

b. Upper Confidence Bounds

As the best action is unknown at the beginning, we will select the next action to take in the next time step using a mix of the Upper-Confidence-Bound (UCB) strategy and the ϵ-greedy policy. For ϵ % of time, we choose a random action, and the $(100-\epsilon)$ % of time we choose the action according to the UCB strategy: $A_t = \arg\max_a [Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}}]$, where $c > 0$ controls the degree of exploration and $N_t(a)$ denotes the number of times action a has been chosen until time step t. c is set to 2 for this experiment.

As mentioned in section B., for the experiment, the space of actions is defined as $\{0, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, 2.75, 3, 4, 5, 7, 10\}$. The parameters mentioned below have been tested for this space of action.

The algorithm terminates when the Q-table converges to the true Q-table $Q*$, however, this generally takes a few hundred iterations. We observed that the action having the highest Q-value stops to change long before the Q-table converges. Therefore, we decided to alter the convergence conditions to the following: UCB converges when the action having the highest Q-value remains unchanged for 15 consecutive iterations. We start counting for these iterations after the 35th trial. Due to time constraints, when UCB reaches 100 iterations, we also terminate the algorithm. In this case, we choose the gain with the highest Q-value as the distortion threshold. In order to let the algorithm fully explore the actions in the starting iterations, we set ϵ, the exploration ratio to 1 in the beginning, and decay it following $\epsilon_{t+1} = \epsilon_t - \frac{\log t+1}{20}$ (t is the number of iterations), until it diminishes to the minimal exploration ratio 0.01. Similarly, the initial learning rate α for the Q-learning update $Q_{t+1} = Q_t + \alpha[R_t - Q_t]$ is set to 0.5, and decays in the same fashion: $\alpha_{t+1} = \alpha_t - \frac{\log t+1}{40}$ until it reaches the minimal learning rate 0.001. The denominators of these two decaying rules are found through grid search.

C. Multi-Armed Bandit with 2 varying parameters

We also attempted to develop a variant of the UCB algorithm that would find the best thresholds for d_{range} and gain for the same distortion function. In other words, we try to find the best numerical combination of d_{range} and G that would give a maximal distortion without provoking a BIE.

We attempted to solve this problem in two ways. We first tried to enlist all the possible combinations of the discretized gain and d_{range} values. Each combination represents an independent arm in the multi-armed bandit modelling. However, it was too difficult to define the reward function for this problem, as the maximal distortion no longer solely rely on the magnitude of the changing parameters. In fact, the maximal gain combined with the maximal d_{range} does make the subject experience the maximal distortion.

The second solution consisted on creating two independent multi-armed bandits, one for the distortion gain and another one for the d_{range}. At each trial, we update the Q-table for the gain and the Q-table for the d_{range} individually. The algorithm for the distortion gain stays the same as described in section B.2. For the d_{range}, we simplify the output of its reward function to $r : A \to \{-1, 1\}$. When a d_{range} value combined with a distortion gain gets a positive response from the subject (no BIE), the d_{range} agent receive $+1$ and the gain agent receives +gain as reward, and vice-versa. However, this algorithm failed to converge to a reasonable threshold during the first trial with a pilot. We did not have further time to investigate the reason of its failure.

IV. Experiment

As mentioned previously in section II, we will use the distortion function described in Porssut et al. [2] with R as the radius of the real tennis ball, d_{range} as the radius of the circular trajectory (0.35m) and G as the value returned by UCB / staircase. The purpose of this study is to find the distortion threshold with both algorithms, and compare their performance. Ideally, we want an adaptive algorithm that is robust and not conservative, and also fast-converging. The algorithm that satisfies the most of these conditions will be ranked higher. In order to assess this goal, we formulate different hypotheses.

First, the UCB algorithm is more robust than the staircase algorithm (H1). We quantify ”robustness” (eq.1) as the percentage of time the subject experiences a Break In Embodiment (BIE), i.e. they notice that their movement is distorted, when the distortion gain applied is under or equal to their threshold. The lower this percentage, the more robust an algorithm is. Robustness is the most important criterion for this experiment. We hypothesize that UCB is more robust, because we implemented a mechanism to correct a portion of the subject’s reaction (sec. A.). If subject has never detected a distortion at a certain level of gain and suddenly detects it, we consider their response as noisy and correct it.

Secondly, we hypothesize that the UCB algorithm is more conservative than the staircase algorithm (H2). We quantify ”conservativeness” (eq.2) as the percentage of time the subject does not experience a BIE when the gain is above their threshold. The lower this percentage, the less conservative an algorithm is. We would like the conservativeness of the algorithm to be as low as possible while still being very robust. However, we noticed that it is not possible to have an algorithm that is very robust and not conservative simultaneously. Because of how we defined the reward function ($R = \pm$gain of distortion depending on if the subject experiences BIE or not (sec. B.2)), the UCB will probably converge to a lower value than the staircase because it avoids to receive negative rewards.

Thirdly, we hypothesize that the UCB algorithm converges in less iterations than the staircase (H3). We defined the convergence condition for staircase as Bovet et al. [3] and Debarba et al. [4]: a staircase converges if there are 7 turns in direction, and terminates in maximum 20 iterations. We will run 4 staircases in parallel, so the staircase algorithm takes in the worst case 80 iterations to terminate, less otherwise. With the final distortion gain range [0, 10], the UCB will converge around 53 ± 2.5 iterations in the ideal conditions (the reaction of the subject to the same distortion gain is stable).

Finally, we hypothesize that the UCB algorithm is more adaptive than the staircase (H4). We did not perform any experimentation to verify this hypothesis. However, by definition, we modelled the problem as a non-stationary multi-armed bandit, which already includes the possibility that the reaction of the subject to the same distortion value might change over time. The staircase is not adaptive by definition.

We learnt from [3] [4] [2] that, as the experiment progresses, the training effect makes the subject getting used to the distortion, thus it is possible that the threshold of the subject at the end of the experiment becomes different than the one they had at the beginning. As we needed to test two algorithms in sequential order, we decided to start the experiment with UCB followed by staircase for half of the subjects, and staircase followed by UCB for the other half to avoid inducing bias. If we need a subject to redo this experiment another day and their threshold has changed, the adaptivity property of the UCB algorithm is valuable because it allows to find the new threshold in less iterations. We have tested that given the Q-table of a previous converged UCB, the algorithm needs less iterations to converge in the ideal conditions. In comparison, for the staircase algorithm, we would need to re-run all the 80 iterations. However, it would be better to do another experiment with real subjects to validate this hypothesis.

A pilot study of 5 subjects had been conducted to adjust the task of the experiment and the question to answer. As mentioned in section II, due to the fact that several participants did not experience any BIE at the original maximal distortion value 4, we increased it to 10. Additionally, the movement was also modified to a two-phase movement (sec. II).

A. Implementation

Equipment and software

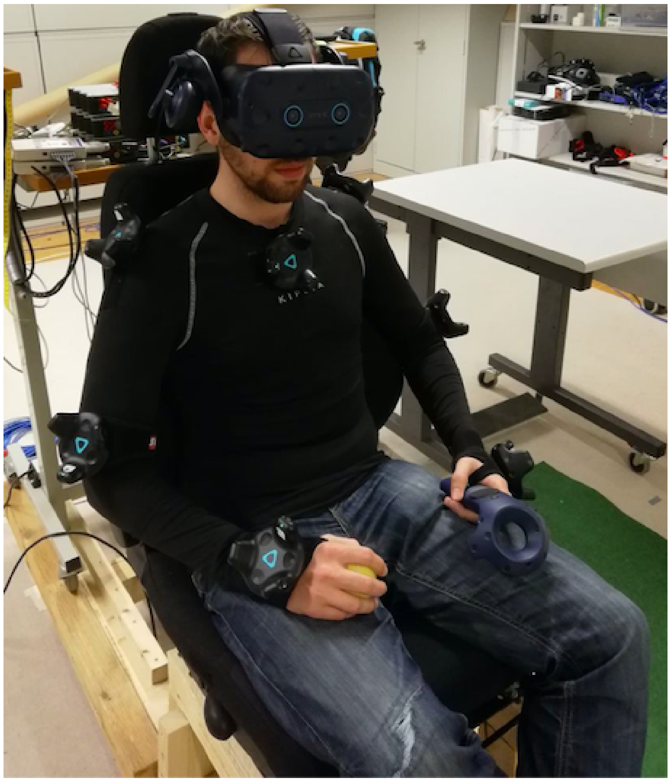

Fig. 5: Equipment: the subject has 2 trackers on the shoulder, 2 on the elbows, 2 on the hands and 1 in front of the chest. The subject holds the tennis ball in the right hand and the HTC Vive Controller in the left hand.

The HTC Vive Pro Eye, a Head Mounted Display (HMD) with 1440 x 1600 pixels per eye, 110 ° field of view and 90 Hz refresh rate, is used for display. This headset has a 120 Hz eye tracking system with 0.5-1.1 ° of accuracy and 110 ° of field of view. We use the eye tracking to ensure that the subject is always looking at their right hand in the second phase of the task. Bose QuietComfort 35 wireless headphones with active noise canceling are used to play a non-localized white noise to the subject during the experiment. The white noise is interrupted when communicating with the subject. For the motion capture we use 8 HTC Vive Trackers V2 placed at the origin of the room (in front of the chair where the subject sits) (1), on the subject's chest (1), shoulders (2), elbows (2) and hands (2). The subject also holds in the left hand an HTC Vive Controller for question answering.

The virtual environment is a square room of $6 \times 6 \times 3m^3$ with a chair in the middle of the room. An avatar holding a green tennis ball in the right hand is calibrated to collocate with the subject's body. This maintains a visuo, proprioceptive and tactile coherence between the real and virtual hands in the absence of fingers tracking. The application is implemented using Unity 3D 2019.2.0f1. The subject's movements are reproduced through animation by the avatar using the FinalIK package. The subject is seated for the whole duration of the experiment and only needs to perform simple movements with their right hand. There is no fear of marker occlusions.

In the calibration phase, the posture of the avatar is reconstructed with the analytic Inverse Kinematic solver (IK) to the posture of the subject. The avatar is globally scaled based on the subject's height. After calibrating the position of the chest and the shoulders, the subject has to remain seated with their arms forward to align the trackers on their hands with the avatar's hands (represented as red rectangles in the scene). The length of the arms and the position of the hands are calibrated during this step as well. The avatar is not visible during the calibration to prevent the subject from viewing visual penetration.

Staircase Design

Staircase	1	2	3	4
Starting Gain	0	1	7	10

TABLE I: 4 Staircases' Starting Values

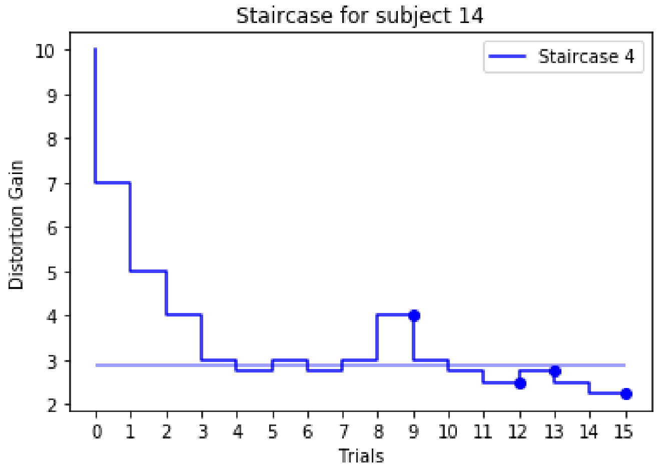

Fig. 6: Example of a converging staircase (7 turns in direction). The dots indicate the last 4 changes in direction (used to obtain the threshold). The horizontal line indicates the distortion threshold, calculated using mean of the distortion gains of the last 4 turns in direction.

The staircase implementation follows the one of Bovet et al. [3] with slight modifications. As the step size between our gain values is not constant, we decided to dismiss the step value used in [3]. The previous/next step of a gain value will be its previous/next direct neighbor. As seen in table I, the staircase block will run 4 staircases in parallel, each having a different starting gain. The staircases are presented at each iteration in a random order to avoid the subject's habituation. If the subject detects the distortion, the gain is lowered, and vice-versa. Once a staircase converges, it will return random gain until all the staircases converge or terminate. The stopping and convergence criteria remain unchanged: the staircase converges when the direction changes 7 times (7 staircase turns) or when it reaches 20 iterations. The distortion threshold is calculated using the mean of the distortion gains the last 4 turns in direction.

Perturbed rewards

The subject is not aware of their threshold, their reaction to the same distortion gain is not stable. Additionally, we might have some noises in the reaction of the subject due to their fatigue or to the training effect. Reinforcement learning algorithms in general are vulnerable to perturbed rewards. In our case, the UCB algorithm might have difficulty to converge when the reaction of the subject to the same distortion gain is very unstable. Sometimes, it might also converge to a wrong threshold (ex. 0, aka no distortion, when the subject does not experience a BIE at a higher distortion gain). Wang et al. [13] proposed a method to counter this problem. When the RL agent receives the perturbed reward from the subject's answer, it will predict the true reward based on the accumulated history of rewards for the corresponding distortion gain (majority voting). For instance, at iteration t, the subject detects the distortion having a gain of g. If in the past, the subject has encountered g 4 times and has only detected it once, then we consider their current response as noisy and we will correct it to "No detection".

B. *Participants*

A pilot study has been done with 5 participants. The experiment has been conducted with 22 subjects. Before starting the experiment, they are asked to read the information sheet and complete the informed consent form. Then they need to fill in a form with questions about their VR background (gaming experience, previous experience with VR applications, etc.) The 22 subjects were paid 20 CHF per hour for their participation. One subject's data was discarded due to technical issues.

The 21 subjects included in the analysis are aged from 18 to 25, in average 21.14 ± 1.9 years old. They are 4 females and 17 males. 20 of them are right-handed and 1 of them is left-handed. Most of the them do not have a lot of experience with VR and do not play video games on a regular basis. One of them has extensive VR experience, and two of them play 7-9 hours of video games per week.

C. Methods

The experiment is divided into 2 blocks: one using the staircase method, and the other one using the UCB method. Among the 21 participants, about half of them (10) will start with staircase followed by UCB, and the other half (11) will complete the blocks in the inverse order to avoid inducing bias.

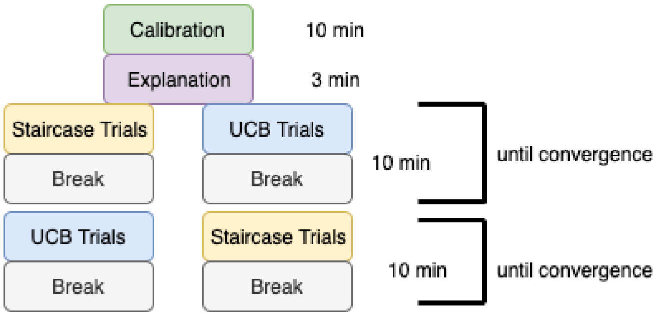

Fig. 7: Protocol Overview

The experiment proceeds in three steps. The first step is for calibration (sec. A.) (approximately 10 min). The second step is for explanation. The explanation consists of 6 trials alternating with no distortion gain (0) and maximal distortion gain (10) to show the distortion to the subject. The last step is the task, divided into two blocks (sec. C.). The number of trials for each block is not predefined; it depends on the reaction of the subject. Each block terminates when the corresponding algorithm in charge of the finding the distortion threshold converges or terminates. The subject is asked to take a break at each 10 minutes (timed with a timer) to avoid them getting too tired and rush through the experiment. The subject is asked to remain seated at all time to avoid technical issues. We resume the experiment when they want. Each block is done only once. The entire experiment lasts for about 1 hour and ends with a short debriefing with the subject.

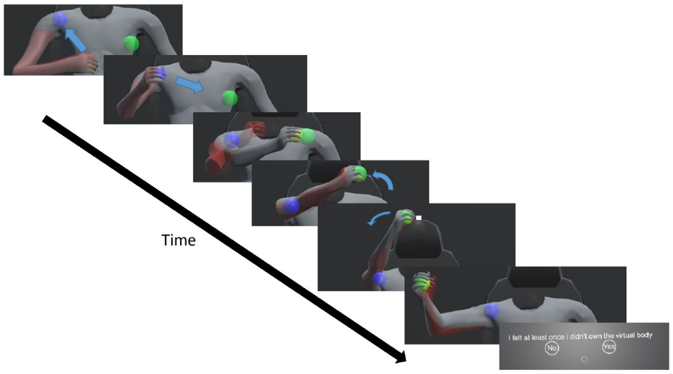

Fig. 8: Overview of a trial (front view): The subject first put the ball inside the blue sphere, then follows the green sphere, and the trial ends after the subject answers to the question. In this figure, the red arm illustrates the subject's actual movement when following the target, while the grey arm belongs to the virtual avatar. The subject is unable to see the red arm during the experiment. (see the video for a sample of the experiment)

Task

The task consists on a two-phase movement and a question. The task stays the same throughout the experiment. The subject starts with the tennis ball held in their right hand, placed in front of their chest. In the first part of the movement, the subject has to put the tennis ball inside a semitransparent blue sphere in front them and wait for a loading circle to complete. From this point in time, the eye tracking system is activated to track the eye sight of the participant. The latter needs to stay focused on the movement of their right hand, or else the trial will be restarted. We created an invisible collider around the right hand of the avatar, and if the gaze of the subject does to touch the collider for 0.5s, the trials restarts after showing a warning message to the subject. Staying focused on their right hand, they need to move the tennis ball from the blue sphere to the center of a semitransparent green sphere. Then, they need to follow the green sphere that moves along a circular trajectory having radius of 0.35m for a few seconds. This moment of transition is illustrated in fig. 9. The task is completed if the subject has followed their right hand's movement with their eyes and has maintained the tennis ball inside the green sphere for at least 4 seconds. Otherwise, the trial restarts from the beginning. Once the movement is completed, the target spheres and the avatar become invisible.

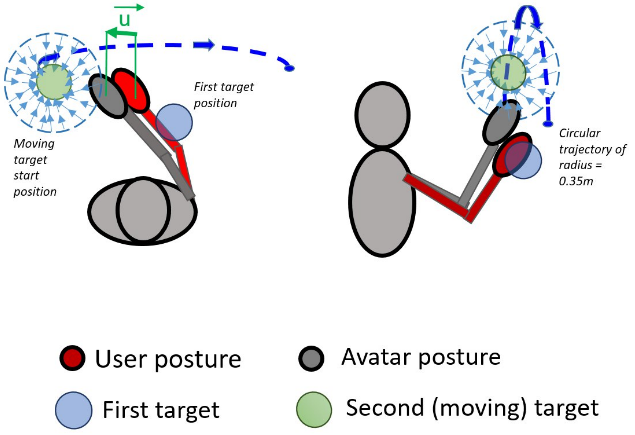

Fig. 9: This depicts the moment when the subject leaves the first target (blue sphere) and try to reach the moving target (green sphere). (Left) Top view. In dark blue: the circular trajectory of the moving target (green sphere). The vector $\overrightarrow{u}$ is the discrepancy induced by the distortion. (right) View from the right. The circular trajectory has a radius of 0.35m.

Then, a question and a cursor (white circle) will appear on the virtual wall in front of the subject. The subject needs to direct the cursor with their head orientation into the ”Yes” / ”No” circles, and press the controller's trigger for about two seconds to answer to the question. Once the question is answered, the objects in the virtual scene appear again at their initial position and the next trial starts.

Question

We gave the same description as in [14] to the subjects to explain the concept of Break In Embodiment (BIE). Kokkinara et al.[14] count the number of BIEs per session and make the subject say ”Now” every time they experience a BIE. Instead, we only want to know if the subject has experienced a BIE or not in the past round. Also, we want our subjects to be focused on the task during the trial. We therefore formulated a question for the subject to answer at the end of each trial with ”Yes” or ”No”: ”I felt at least once I didn't own the virtual body.”

V. Results

We perform comparisons within subjects. For the statistical analysis, differences are deemed statistically significant for p-values below the threshold $\alpha = 0.05$.

First, we will verify H1 and H2: UCB is more robust but more conservative than the staircase. We define the robustness (R) of an algorithm as the percentage of time the subject detected the distortion when the applied gain is equal or below the threshold found by this algorithm. The conservativeness of the algorithm is defined as the percentage of time the subject did not detect the distortion when the applied gain is above the found threshold. For each subject and each algorithm, we calculate the robustness (R) and the conservativeness (C) of this algorithm using the threshold that it has found for this subject. The staircases that did not converge are discarded as in [4].

Given distortion threshold T found by algorithm A for subject S, we compute the robustness (R) and conservativeness (C) of A as:

$$R(\%) = \frac{\text{nb trials gain} <= \text{T AND S experiences BIE}}{\text{total nb of trials during the experiment}} \quad (1)$$

$$C(\%) = \frac{\text{nb trials gain} > \text{T AND no BIE}}{\text{total nb of trials during the experiment}} \quad (2)$$

For each subject, we pick out 3 categories of 4 levels of values according to the conditions listed below. The three categories are: the value of the distortion threshold, the robustness (%) of this threshold, and its conservativeness (%).

1) Most robust staircase: The value of the staircase having the smallest R (%)
2) Least conservative staircase: The value of the staircase having the smallest C (%)
3) In-middle staircase: The value of the staircase having the smallest absolute difference between R (%) and C (%)
4) UCB: The corresponding value of the UCB

In the case that there are several staircases that met the same condition (same R, same C, or both), we take the mean. We thus have 4 levels of 21 values, for each category.

For each category, for each group of value, we carried a Shapiro-Wilk test to test the null hypothesis that the data was drawn from a normal distribution. As there was always at least one group that do not pertain to the Gaussian distribution, we cannot use one-way ANOVA analysis. Instead, we carried the Friedman test over the four levels to detect differences in value distributions across the groups. As the results were significant, we performed a post-hoc analysis with a repeated two-sided Wilcoxon signed-rank test for multiple comparisons. Bonferroni method was used to correct the previous multiple comparison tests.

A. Robustness

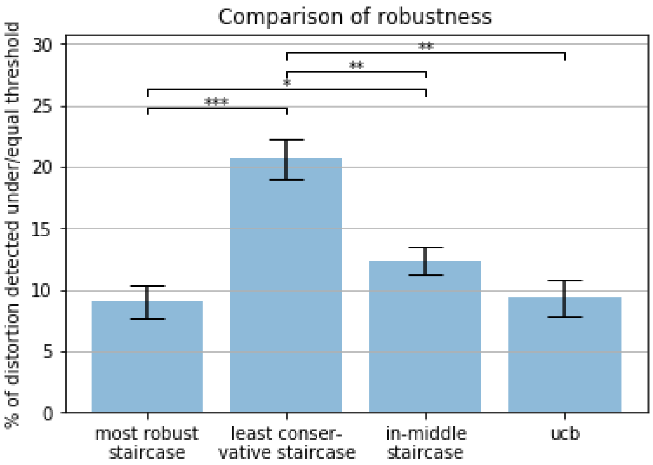

Fig. 10: Bar graph for the pairwise comparison of robustness. Error bars represent the standard error of the mean.

Shapiro shows that group 1, 3 and 4 (most robust, in-middle, ucb) do not come from a Gaussian distribution ($W_1 = 0.875$, $p = 0.012$; $W_3 = 0.885$, $p = 0.018$; $W_4 = 0.8974$, $p = 0.0312$). The Friedman test shows a significant difference in the distribution of the four groups of robustness percentages ($Q = 36.971, p = 4.66 \times 10^{-08}$). With the multiple comparisons Wilcoxon test and Bonferroni correction, figure 10 is built. We can see that the robustness of the most robust staircase and the UCB are almost equivalent ($p_{\text{corrected}} = 1$). Also, the Wilcoxon test failed to reject the null hypothesis between the in-middle staircase and UCB ($p_{\text{corrected}} = 0.65$). However, there is a significant difference between the robustness of in-middle staircase and the most robust staircase ($p_{\text{corrected}} = 0.03$). This suggests that the robustness of UCB is in between the most robust staircase and the in-middle staircase, and very close to the most robust staircase. We do not confirm our H1, but we do not reject it completely either, as UCB is significantly more robust than the least conservative staircase. we can conclude that UCB has a comparable robustness to the most robust staircase.

B. Conservativeness

Shapiro shows that only group 1 (most robust) does not come from a Gaussian distribution ($W = 0.883$, $p = 0.017$). The Friedman test shows a significant difference in the distribution of the four groups of conservativeness percentages ($Q = 36.971, p = 4.66 \times 10^{-08}$). With the multiple comparisons Wilcoxon test and Bonferroni correction, figure 11 is built. As expected, we find the inverse trend as the one found for robustness (fig. 10). When the robustness value is maximal

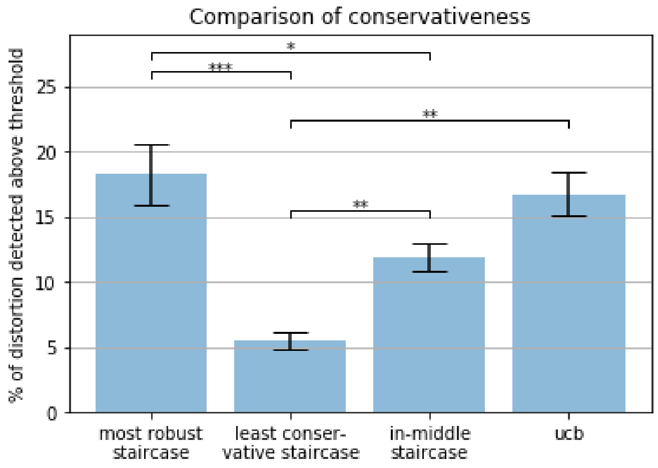

Fig. 11: Bar graph for the pairwise comparison of conservativeness. Error bars represent the standard error of the mean.

the conservativeness value is minimal. We also observe that the conservativeness of UCB is in between the most robust staircase and the in-middle staircase. As in the analysis of robustness, we do not find a significant difference in the conservativeness between the UCB and in-middle-staircase and the UCB and the most robust staircase. Thus, we do not confirm nor reject our H2. Judging on the bar plot and the statistics, we can conclude that UCB has a comparable conservativeness to the most robust staircase and to the in-middle staircase.

C. Distortion thresholds

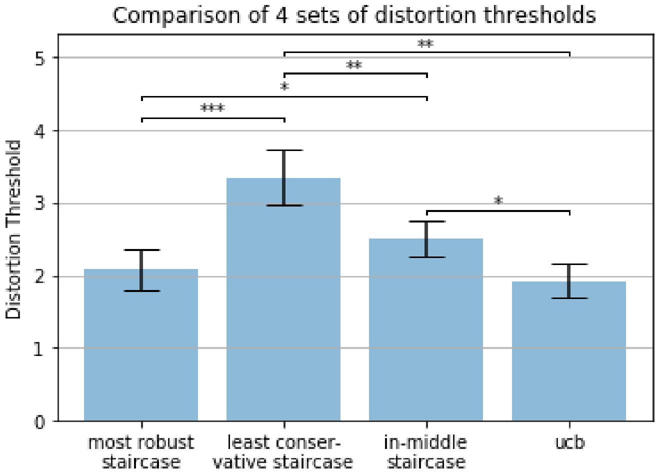

Fig. 12: Bar graph for the pairwise comparison of distortion thresholds. Error bars represent the standard error of the mean.

Shapiro shows that group 1, 2, 4 (most robust, least conservative, ucb) do not come from a Gaussian distribution ($W_1 = 0.861$, $p = 0.006$; $W_2 = 0.903$, $p = 0.039$; $W_4 = 0.879$, $p = 0.014$;). The Friedman test shows a significant difference in the distribution of the four groups of distortion thresholds ($Q = 38.262, p = 2.487e^{-08}$). With the multiple comparisons Wilcoxon test and Bonferroni correction, figure 12 is built, and we observe that the graph is very similar to the robustness graph (fig. 10). There is one slight difference: the in-middle staircase's distortion thresholds are significantly different than the thresholds found by UCB ($p_{\text{corrected}} = 0.007$). However, their percentages in robustness and in conservativeness do not differ significantly. One possible reason is that staircase and UCB do not have the same convergence criterion. Staircase terminates after 20 trials, which caused 25% of the staircases to not converge. We might needed to increase the number of trials for each staircase to get more data points, and this might lead to different results. It is also possible that the number of values tested above or under the threshold might not be enough, thus we could not obtain a significant difference. Finally, according to Porssut et al. [2], the threshold of each subject is quite different from each other. The inter-subject variance can be quite high. This variance may explain why we did not find a significant difference for robustness and conservativeness, but we found one for the distortion gains. In fact, judging solely on the graphs, we can observe a difference between the in-middle staircase and UCB's robustness and conservativeness, however, the Wilcoxon tests could not reject the null hypotheses.

In sum, we can conclude that the thresholds found by UCB performs comparably well as the most robust thresholds found by the staircase algorithm. However, considering their convergence probability, UCB seems to be a better choice. In fact, 95% UCB algorithms successfully converged, while as only 75% of staircases converged. Since we are running 4 staircases in parallel per subject, this means that in average, one staircase did not converge per person.

D. Convergence speed

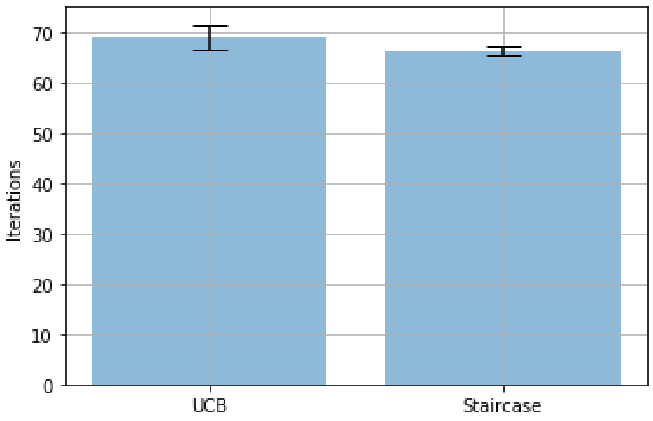

Fig. 13: Bar graph illustrating the number of iterations needed for the algorithm to converge. Error bars represent the standard error of the mean.

For 21 subjects, it takes in average 69 ± 11.5 iterations for the UCB algorithm to converge, and the staircase algorithm takes 66 ± 4 iterations. For staircases that didn't converge, we set the number of iterations needed for convergence to 20, the maximal number of iterations. We can see from 13 that staircase has a much smaller standard error and a slightly lower average than the UCB algorithm. Each trial only takes in average 6.495 ± 0.573 seconds to complete, which means that UCB might take 94.2s $((11.5+(69-66))\times 6.495)$ more to converge (roughly 1 min 35 s). According to our results, H3 is rejected, but the difference in terms of seconds is negligible. Additionally, these results might have been different if the maximal number of trials for each staircase which did not converge was above 20.

E. Comparison to previous study

For the 21 subjects, the average of the threshold found by all the converged staircases is 2.653 ± 1.635, and 1.929 ± 1.102 for UCB. The averages of the distortion gain are higher compared to Porssut et al. [2] (1.43 ± 0.41), which might explain why four of their subjects had a distortion threshold above 2. Indeed, some subjects can tolerate a distortion gain up to 5, which exceeds largely the maximal threshold (G = 2) Porssut et al.[2] had fixed for their experiment. Similarly to their results, we also found a high standard deviation emphasizing the high variability of thresholds between the subjects.

F. Correlation with video gaming

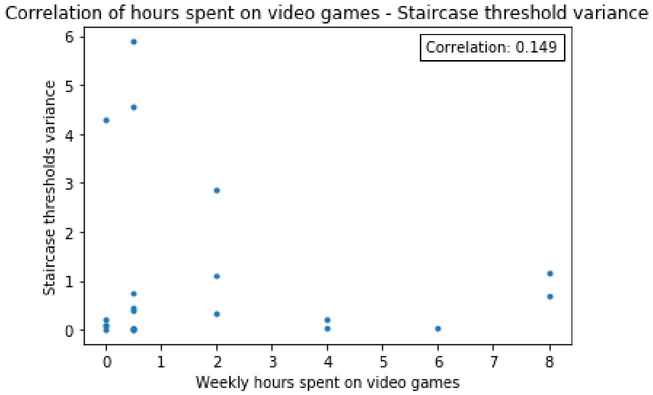

Fig. 14: Scatter plot with Spearman correlation of the number of hours each subject spends weekly on video games and the variance of the thresholds found by their converged staircases.

From Figure 16, we observe that certain box plots of the staircase distortion thresholds seem to converge around the same value (subject 0), while as some others have the thresholds spread over a large range of values (subject 8). We suspect a possible correlation between the variance of the thresholds found by the staircase and the gaming experience of the subject (the number of hours they play video games per week) (fig. 14). Similarly, we also suspect that the convergence probability of the staircase is correlated with the gaming experience (fig. 15). However, the Spearman correlation did not reveal any significant correlation (the absolute values of the correlation values were all below 0.3).

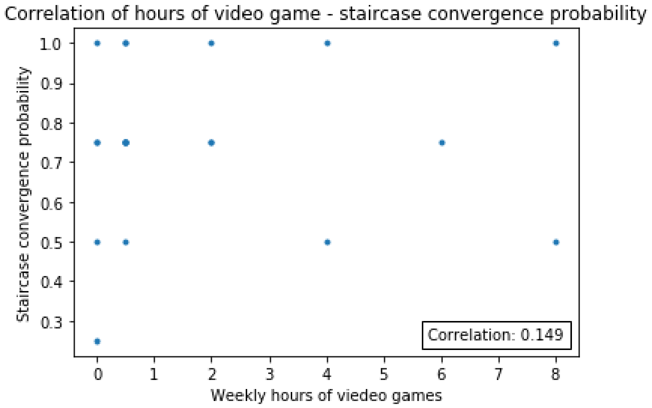

Fig. 15: Scatter plot with Spearman correlation of the number of hours each subject spends weekly on video games and the convergence probability of staircases per subject.

VI. Discussion and conclusion

This study explores the possibility of solving the problem of finding the self-attribution threshold using reinforcement learning algorithms. It extends the study of Porssut et al. [2] about finding the subjective self-attribution threshold for individual subject.

We consider two algorithms, Upper-Confidence-Bounds (UCB) and staircase, to find the thresholds. We evaluate the two algorithms with four criteria: 1) Robustness; 2) Conservativeness; 3) Convergence speed; 4) Adaptivity to the change in threshold. The ideal algorithm should be on the same time very robust and, based on that, the least conservative possible, necessitating a low number of iterations to converge. It also needs to adapt to the future variations of the subject's threshold.

Due to the design of the reward function and the correction mechanism of the reward, we hypothesized that the UCB is more robust, but also more conservative than the staircase. The result shows that UCB's robustness is comparable to the most robust staircase, and its conservativeness is comparable to the most robust and the in-middle staircases. Considering the fact we weight robustness more importantly than conservativeness because we absolutely want to avoid provoking BIEs, UCB would be a better choice.

Concerning the convergence speed, both algorithms converges in average within 70 iterations, but staircase converges slightly faster. In terms of time, UCB would need roughly 1 min 35s more to converge, which is negligible.

Concerning the adaptivity, we modelled the problem as a non-stationary multi-armed bandit, which by definition assumes that the distortion threshold of the same subject changes over time. As staircase is known as a non-adaptive algorithm, UCB is by definition more adaptive than staircase.

In conclusion, UCB is more robust and adaptive, and staircase is less conservative and converges in less iterations. However, considering that robustness is the most important evaluation criterion, UCB would be the winning algorithm. In

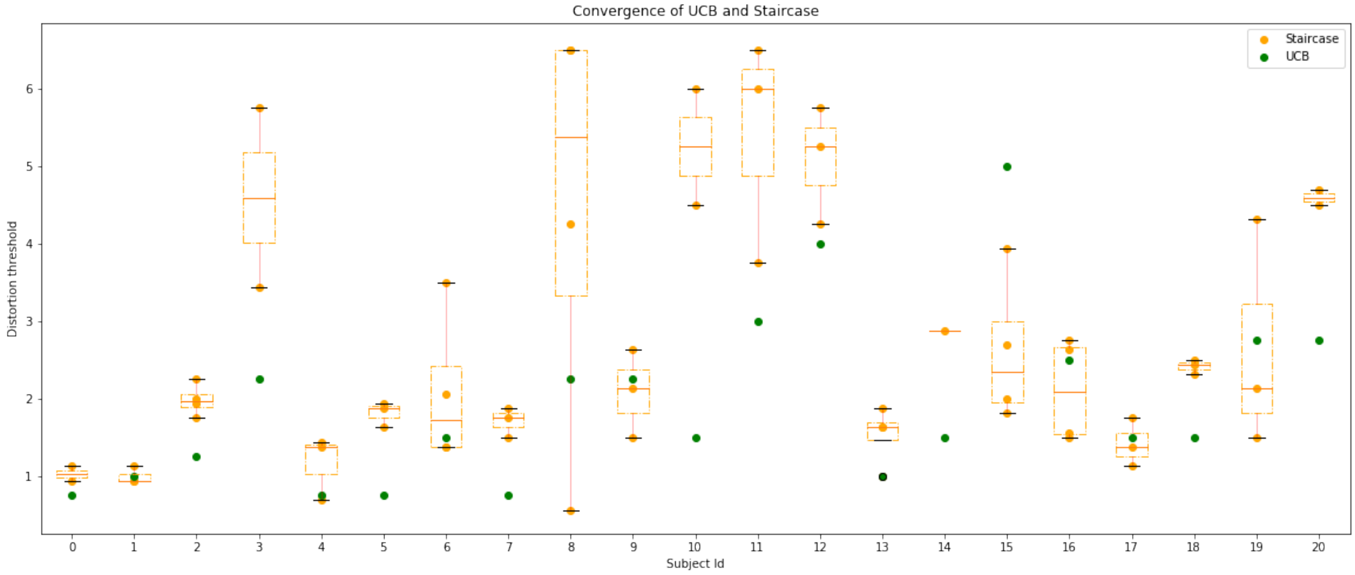

Fig. 16: Distortion thresholds found by the staircase and UCB algorithms for 21 subjects

addition, considering the high probability of non-convergence of staircase, it is also safer to use the UCB algorithm which almost always converges. In terms of convergence speed, the difference between the two algorithms is negligible. At last, UCB is an adaptive algorithm, while as staircase is not. Considering all these factors, UCB is a better algorithm to use to find the distortion threshold of the subjects.

In the debriefing with the subjects, we asked them at which moment do they experience a BIE and which factors helped them notice the distortion. Globally, the subjects report to have observed the ”jump” when they move the tennis ball from inside the blue sphere to follow the green sphere. They described as feeling helped by an external force, or feeling that the avatar’s hand is automatically attracted to the green ball. Some also report to have noticed the distortion when the movement of the avatar’s hand is too steady compared to the movement of their physical hand when they follow the green sphere on its circular trajectory. These feedbacks correspond to our expectations as they are coherent with the behaviour of the distortion function.

Porssut et al. [2] mentioned a high variability of the thresholds among subjects. This phenomenon is again found in this study. Also, the observation that their subjects had a relatively high distortion thresholds is coherent to what we have found. We can conclude that our results are consistent with the previous findings.

There are several points we could improve about this experiment. As we previously mentioned, we have a relatively high percentage of staircases that did not converge. We have set the maximal iterations of 20 of the staircase algorithm following [4] [3], but this limit can probably be increased further to allow a higher convergence probability. In fact, several non-converged staircases have reached 5 or 6 turns in direction, and they may only need 5 to 10 more iterations to reach the 7th turn needed for convergence. Otherwise, instead of determining one threshold per staircase, we could have used the Point of Subjective Equality (PSE) as in Burns et al. [15] to find one single threshold for all the staircases. In this case, we would avoid having four staircases converging to four different values.

Due to time constraints, we could not perform any experimentation with subjects to verify the adaptivity of the tested algorithms. By definition, UCB is more adaptive than staircase. We have tested when given the Q-table of a previous converged UCB, the algorithm takes less iterations to converge. However, as observed in this experiment, subjects are not aware of their true threshold, and their reaction may also be affected by the fatigue and the training effect. It would be worthwhile to conduct another study to test the adaptivity of UCB with real subjects.

In this experiment, we asked the subject to answer a question at end of each trial to indicate if they have experienced a BIE. Having explicit human feedback is very advantageous to update the RL agent’s Q-table. However, it is very demanding and tiresome for a human to simultaneously stay focused on the virtual scene while continuously generating feedback. Therefore, developing an approach to obtain implicit feedback is highly relevant. Studies from [6], [7], [8], [9], [10], and [16] demonstrated the possibility to use supervised learning models to detect the presence of a EEG signal, generated by the subject’s brain when they observe an error. In particular, Salazar-Gomez et al. [16] state that the accuracy of classifying secondary EEG - defined as the second EEG signal provoked by the misclassification the primary one - has an higher accuracy than solely using the primary EEG. If we can combine this technique with our algorithm, we could maybe adjust implicitly the distortion threshold in a VR experience. Modelling a classifier to detect the presence of EEG signals will be subject of later experiments, thus not the focus of the current project.

At last, we believe that our study will also be useful for motor rehabilitation. It has been shown by Cameirao et al. [17] that task oriented rehabilitation combined with the observation of virtual limbs facilitate the functional recovery of the arms. In this context, finding the correct distortion threshold of the subjects would help them to consider the distorted movement as their own, which could positively impact their recovery process. However, our method has only been evaluated for a predefined movement. It would be necessary to test the performance of the algorithm when the movement is only partially known or entirely unknown.

References

[1] Konstantina Kilteni, Raphaela Groten, and Mel Slater. The sense of embodiment in virtual reality. *Presence Teleoperators amp Virtual Environments*, 21, 11 2012.

[2] T. Porssut, B. Herbelin, and R. Boulic. Reconciling being in-control vs. being helped for the execution of complex movements in vr. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 529–537, March 2019.

[3] Sidney Bovet, Henrique Galvan Debarba, Bruno Herbelin, Eray Molla, and Ronan Boulic. The critical role of self-contact for embodiment in virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 24(4):1428–1436, April 2018.

[4] Henrique Galvan Debarba, Ronan Boulic, Roy Salomon, Olaf Blanke, and Bruno Herbelin. Self-attribution of distorted reaching movements in immersive virtual reality. *Computers amp; Graphics*, 76:142–152, 2018.

[5] Eric Burns, Sharif Razzaque, Abigail Panter, Mary Whitton, Matthew McCallus, and Frederick Brooks, Jr. The hand is more easily fooled than the eye: Users are more sensitive to visual interpenetration than to visual-proprioceptive discrepancy. *Presence*, 15:1–15, 02 2006.

[6] Iñaki Iturrate, Jason Omedes, and Luis Montesano. Shared control of a robot using eeg-based feedback signals. In *Proceedings of the 2Nd Workshop on Machine Learning for Interactive Systems: Bridging the Gap Between Perception, Action and Communication*, MLIS '13, pages 45–50, New York, NY, USA, 2013. ACM.

[7] Iñaki Iturrate, Luis Montesano, and Javier Minguez. Robot reinforcement learning using eeg-based reward signals. pages 4822–4829, 05 2010.

[8] Tian-jian Luo, Ya-chao Fan, Ji-tu Lv, and Changle Zhou. Deep reinforcement learning from error-related potentials via an eeg-based brain-computer interface. pages 697–701, 12 2018.

[9] Ricardo Chavarriaga, Andrea Biasiucci, Killian Forster, Daniel Roggen, Gerhard Troster, and Jose del R. Millan. Adaptation of hybrid human-computer interaction systems using eeg error-related potentials. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 2010:4226–9, 08 2010.

[10] Su-Kyoung Kim, Elsa Kirchner, Arne Stefes, and Frank Kirchner. Intrinsic interactive reinforcement learning – using error-related potentials for real world human-robot interaction. *Scientific Reports*, 7, 12 2017.

[11] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[12] Timothy Meese. Using the standard staircase to measure the point of subjective equality: A guide based on computer simulations. *Perception psychophysics*, 57:267–81, 04 1995.

[13] Jingkang Wang, Yang Liu, and Bo Li. Reinforcement learning with perturbed rewards. *CoRR*, abs/1810.01032, 2018.

[14] Elena Kokkinara and Mel Slater. Measuring the effects through time of the influence of visuomotor and visuotactile synchronous stimulation on a virtual body ownership illusion. *Perception*, 43(1):43–58, 2014. PMID: 24689131.

[15] Eric Burns, Sharif Razzaque, Mary Whitton, and Frederick Brooks, Jr. Macbeth: Management of avatar conflict by employment of a technique hybrid. *IJVR*, 6:11–20, 01 2007.

[16] Andres Salazar-Gomez, Joseph DelPreto, Stephanie Gil, Frank Guenther, and Daniela Rus. Correcting robot mistakes in real time using eeg signals. pages 6570–6577, 05 2017.

[17] Mónica Cameirão, Sergi Bermúdez i Badia, Esther Duarte, and Paul Verschure. Virtual reality based rehabilitation speeds up functional recovery of the upper extremities after stroke: A randomized controlled pilot study in the acute phase of stroke using the rehabilitation gaming system. *Restorative neurology and neuroscience*, 29:287–98, 05 2011.