The Interdisciplinary Center, Herzliya
Efi Arazi School of Computer Science
M.Sc. Program - Research Track

Algorithmic Induction of Physiological States: First Steps

by
Keren-Or Berkers

M.Sc. dissertation, submitted in partial fulfillment of the requirements
for the M.Sc. degree, research track, School of Computer Science
The Interdisciplinary Center, Herzliya

May, 2019

I wish to express my sincere thanks to Prof. Doron Friedman who gave me the freedom to explore the first steps of an intelligent human machine interaction based on human physiological signals and which this work was carried out under his supervision. My grateful thanks goes to Jonathan Giron for valuable support and commitment to help this research succeed, including but not limited to lab, equipment and experiment set-up. Also, I would like to extend my thanks to Asaf Shemla and Debbie Chetrit for helping me run the experiment. Special thanks to my spouse Ori Pelleg, for his utmost support, patience and belief in my work. Lastly, my thank you goes to Irit Schlezinger-Berkers and Carla Pelleg, my mother and mother-in-law, for their help and playtime with my son during the more challenging parts of this research. Although they told me they are the ones that were lucky, I am also lucky to have such a supportive grandma-power.

Abstract

In traditional biofeedback, the participant is required to learn to regulate his or her physiological state, based on continuous feedback about this state. Here we suggest a complimentary scheme, whereby the machine has to learn how to modulate the physiological state of the participant, by ongoing selection of the most effective stimuli sequence. Since the physiological state of the participant is dynamic and might also adapt to the attempts at being modulated, we suggest a framework based on on-line reinforcement learning (RL); in this framework the RL is responsible for online adaptive control. Specifically, we present the first steps towards developing an intelligent system that learns to elicit high (or low) levels of arousal in human participants in a virtual reality (VR) setting.

The combined dynamics of an RL agent and human physiology is complex. In order to study this dynamic systematically we selected a simple physiological signal – skin conductance – of participants experiencing fear inducing stimuli with multiple repetitions. Based on the data from this experiment, as well as findings reported in the psychophysiological literature, we developed a simulator of the physiological responses to the VR stimuli. This simulator was then used to train a machine learning system, whose goal is to learn optimal policies for inducing high (or low) levels of physiological arousal.

Our main challenges are learning to modulate a noisy and non stationary signal, and we specifically model both habituation – the tendency of the response to decline after repeated exposure to identical stimuli – as well as dishabituation – the tendency of the response to rebound back after some duration in which a stimulus does not appear. Noise and habituation were framed as simple n-arm bandit problems, and can be solved by a state-less algorithm. Experiments were conducted with three known algorithms: $\varepsilon - greedy$, softmax and adaptive pursuit, with several parameters. For addressing the more complex challenge of dishabituation we show that a introducing states results in superior performance as compared to the state–less algorithms.

Contents

1 Introduction

The long-term goal of this research is to develop a method, based on online statistical machine learning, whereby the system learns how to systematically regulate the physiology of human participants through a virtual reality (VR) environment. This thesis is a first step in this ambitious direction.

The research was carried out in a VR environment due to its ability to make it possible for people to experience a simulated environment as if they are physically present inside it, while at the same time it provides a safe and controlled laboratory environment.

This research combines three fields: virtual reality, affective computing and reinforcement learning. The following subsections elaborate on affective computing and reinforcement learning. For a recent review of VR research and applications see Slater and Sanchez-Vives (2016).

1.1 Affective Computing

Affective computing is an interdisciplinary field of study, focused on the development of systems that can recognize, interpret, and express human affect. The term was coined by Rosalind Picard as computing that relates to, arises from, or influences emotions (Picard, 1995). Systems that can both be influenced by and influence users corporeally exhibit a quality named an affective loop (Höök, 2009). Affective computing systems aim to create an affective-loop between the system and the end user, by detecting the end-user's state and selecting an appropriate response. The response may or may not influence the end user's affective state (for example, it may simply represent it).

Much of the research in affective computing is aimed at using machine learning to decode an affective state from neurophysioloigical signals. The topic of this thesis is different, and our goal is to develop an active system that modulates the physiological signal; we are not concerned here with the reported subjective state resulting from this induction. We measure and manipulate physiological arousal using skin conductance (SC) (sometimes referred to as electrodemal activity — EDA). SC measurement is the use of the human body's continuous variation in the electrical characteristics of the skin, due to fluctuating levels of moisture, as an indicator of a person's psychological or physiological arousal. This response is associated to the sympathetic nervous system, responsible for our "fight or flight" response (Critchley, 2002; Benedek and Kaernbach, 2010)

1.2 Reinforcement Learning

RL (Sutton and Barto, 1998) is a sub-area of machine learning in which a goal driven agent interacts with a dynamic environment through trial and error. The agent's goal is achieved through a series of actions performed sequentially. After an action has been performed, the agent's sensors enable it to observe the environment. The observation is presented to the agent as a pair of state and reward calculated by the environment interpreter.

The RL problem faced by the agent can be formulated as a Markov decision process (Kaelbling et al., 1996). The reward is a scalar representing the immediate desirability of an action result. The state is derived from the information about the environment that is available to the agent. In RL we are interested in the long run desirability of taking an action in the current state, represented as the $< state, action >$ pair value. It can be calculated as a function of the maximum reward units that an agent can expect to accumulate from the current state. In order to understand what reward units an agent can expect, the agent first decides on a policy, a mapping from environment states to desired actions.

There is a trade-off between the need to obtain new knowledge and the need to use previously obtained knowledge known as the exploration – exploitation dilemma. The most generic RL problems, where this dilemma has to be considered, are single state problems called multi-armed bandit problem (Sutton and Barto, 1998, Section I.2). A realistic and challenging variant of multi-armed bandit problems is a non-stationary bandit problem where the actions' reward value can change over time. In this set of problems the importance of the exploration is not diminished over time. Multi-arm bandit problems with k actions are also referred to as k-arm bandit problems and can be solved by a state-less algorithm, as follows.

Algorithm 1 is a general algorithm for a stationary multi-armed bandit problem where a_t is the action taken at time t, $N(a_t)$ is the number of times action a was chosen, including the current time t, $Q(a_t)$ is the value of action a_t under the agent policy, r_{t+1} the immediate reward of performing a_t, β is $\frac{1}{N(a)}$ and T is the last time the agent take an action. The same algorithm can be used for non-stationary bandit problem with one change – β is a small positive constant, in order to obtain a fixed exploration and exploitation ratio. In this algorithm the probability of each possible action to be the current best action is typically estimated by several methods (Koulouriotis and Xanthopoulos, 2008). Three known methods are $\varepsilon-$greedy (Algorithm 2), $softmax$ (Algorithm 3) and $adaptive\ pursuit$ (Algorithm 4).

In Algorithm 2, $\varepsilon < 1$ is a very small positive constant. In Algorithm 3, τ is a positive parameter called temperature. The higher the temperature is, the greater the probability for exploration is. In Algorithm 4, $P_{min} < 1$ is a very small positive parameter, and prior to running the code in Algorithm 1, P is initialized to $\frac{1}{N(a)}$.

Algorithm 1 k-armed bandit

1: *Initialize* $Q(a)$ *for all actions with* 0
2: **for** t *in* $1..T$ **do**
3: $P_t \leftarrow BestActionProbabilityEstimations(Q)$
4: *randomNum chosen uniformly at random from the interval* $(0..1)$
5: $temp \leftarrow 0$
6: $i \leftarrow 0$
7: **while** $temp < randomNum$ **do**
8: $i \leftarrow i+1$
9: $temp \leftarrow temp + P_t[i]$
10: **end while**
11: $a_t \leftarrow i$
12: $Q(a_t) \leftarrow Q(a_t) + \beta(r_{t+1} - Q(a_t))$
13: **end for**

Algorithm 2 $\varepsilon - greedy$

1: **procedure** $BestActionProbabilityEstimations(Q)$
2: **for** a *in* $1..k$ **do**
3: $P[bestAction = a] = \frac{\varepsilon}{k}$
4: **end for**
5: $a_g = argmax_a Q$
6: $P[bestAction = a_g] += (1 - \varepsilon)$
7: **return** P
8: **end procedure**

Algorithm 3 *softmax*

procedure *BestActionProbabilityEstimations*(Q)
  for *a in 1..k* **do**
    $P[bestAction = a] = \frac{e^{Q(a)/\tau}}{\sum_{i=1}^{n} e^{Q(a_i)/\tau}}$
  end for
  return P
end procedure

Algorithm 4 *AdaptivePursuit*

procedure *BestActionProbabilityEstimations*(Q)
  for *a in 1..k* **do**
    $P[bestAction = a] += \alpha(P_{min} - P[bestAction = a])$
  end for
  $a_g = argmax_a Q$
  $P[bestAction = a_g] += \alpha(1 - kP_{min})$
  return P
end procedure

A well known learning algorithm for RL problems that includes states is SARSA (Sutton and Barto, 1998, Section I.6). Let s_t be the state at time t, a_t the action taken at time t, r_{t+1} the immediate reward of performing a_t at state s_t, $Q(a_t, s_t)$ the value of performing a_t at state s_t and s_{t+1} the new state resulting from this action. The algorithm selects a_{t+1}, the best action to take at time $t+1$ according to the policy driven from Q. Only then it updates $Q(a_t, s_t)$ with the following update rule (1), where α is the learning rate and γ is the discount factor.

$$Q(a_t, s_t) \leftarrow Q(a_t, s_t) + \alpha[r_{t+1} + \gamma \cdot Q(a_{t+1}, s_{t+1}) - Q(a_t, s_t)] \quad (1)$$

Because the SARSA agent updates the policy based on actions taken, this is known as an on–policy learning algorithm.

When the agent environment interaction breaks naturally into subsequences, we call each subsequence an episode.

Let S be the group of all possible states and $A(s)$ be the group of all available actions at state s. Algorithm 5 is the SARSA general algorithm. The action selection policy is derived from Q using one of the three methods: $\varepsilon - greedy$, $softmax$ or $adaptive\ pursuit$.

Algorithm 5 SARSA

Initialize $Q(s,a)$ $\forall s \in S, a \in A(s)$, *arbitrarily*
repeat *for each episode*
    Initialize s_1
    Choose a_1 *from* $A(s)$ *using policy derived from* Q
    for *t in 1..T* **do**
        Take action a_t *and observe* r_{t+1}, s_{t+1}
        Choose a_{t+1} *from* $A(s_{t+1})$ *using policy derived from* Q
        $Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[r_{t+1} + \gamma \cdot Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)]$
    end for
until *no more episodes*

We will use the reinforcement learning (RL) framework as follows: the RL agent is the VR system, the actions taken by the agent are the different stimuli, and the reward is derived from the participant's physiological signal. The goal of the RL algorithm is to learn an optimal policy that would result in driving the physiological signal in a specific direction.

Machine learning in general, and RL specifically, require large amounts of training data. Our methodology in addressing this is based on a simulator of human physiology. The simulator will attempt to capture the main characteristics of the simulated signal. We will begin by using unrealistic models for simplification. Gradually, we will introduce dynamic properties of the signal, approximating the real dynamics of psycho-physiological signals.

1.3 Previous Work

While each of the fields of VR, affective computing (specifically using SC), and RL have been extensively researched, to the extent of our knowledge a combination of all three, i.e. a system capable of real time, RL agent based affective–loop interaction with a human participant based on their physiological readings, has never been studied. In Marín-Morales et al. (2018), a virtual environment was used for emotion elicitation and classification, as opposed to previous studies using non-immersive stimuli. Four alternative virtual rooms were designed to elicit four possible arousal–valence combinations, and the electroencephalography (EEG) and electrocardiography (ECG) of 60 participants was recorded and analyzed. The results show a 75.00% accuracy recognition rate along the arousal dimension and 71.21% along the valence dimension. In Wu et al. (2010), a virtual

environment was used to create three scenarios of increasing arousal levels. The participants were driving through low threat (scenario I) and high-threat (scenario 2) zones, and were tasked with naming colors appearing randomly on the screen. The third scenario was similar to the second, except Stroop tests were used instead of color naming. The participants were tasked to name the color as fast as they could and their response time was measured. Psychophysiological measures, including SC level, were continuously recorded. The results show that for most participants, the optimal arousal levels were elicited in scenario 2. Further, results suggest high classification rates using psychophysiological responses. This research reflects progress toward the implementation of a closed-loop affective computing system. In Tijs et al. (2008), an emotionally adaptive game was created to investigate the relations between game mechanics, players' emotional state and their emotion data. Through manipulation of game speed, the study compared self reported data in terms of valence, arousal and boredom-frustration-enjoyment to (mainly) physiology-based data termed "emotional data". A correlation was found between self-reported and emotional data, and seven emotion–data features were found to distinguish between a boring, frustrating and enjoying game mode. Although this study did not use RL, it was proposed as the next step.

In Liu et al. (2009), two dynamic difficulty adjustment (DDA) approaches for computer games were compared, with the goal of enhancing user experience and reducing anxiety. The first approach was adjusted according to player performance only, and the second approach was adjusting DDA solely according to real–time affect measurements, including SC (based on player–specific, previously constructed models). The results show improved performance and reduced perceived anxiety–level for the majority of the participants, during the affect-based DDA session, compared to the performance-based session.

In Rovira and Slater (2017), RL was used to make participants move to a specific location in an immersive virtual environment and stay there as long as possible, without them being aware of the RL goal. This was achieved by rewarding the RL agent according to the position of the participants. The experiment was held in a VR game environment in which the participants are told they should avoid spacecraft hits. The RL agent then decides where the spacecraft attacks. The results show that the RL agent generally learns to guide participants towards the goal.

The studies above demonstrate growing effort to recognize valence and optimal arousal levels for human participants, and how a VR environment can be utilized for such effort. One study has combined VR and RL to create a real-time adaptive agent. However, none of the above studied a framework combining

on-line physiological signals, an immersive VR environment, and a RL agent capable of fast analysis and adaptation to stimuli over a prolonged period of time, including habituation.

1.4 Contribution

An earlier version of this thesis appeared as a paper: Keren-Or Berkers, Jonathan Giron and Friedman Doron, Algorithmic Induction of Physiological State: First Steps, in the peer reviewed workshop on Lifelong Learning: A Reinforcement Learning Approach (LLARLA) as part of the IJCAI/ICML conference in July 2018.

2 Methods

2.1 The VR experiment

2.1.1 Experimental procedure

The VR experimental scenario starts in a big empty gloomy room (Figure 1). A set of stimuli, related to known human fears, are presented in order to elicit emotional arousal in the participants. The stimuli are presented in a pseudo random order, such that every stimulus is repeated 10 times. Each stimulus duration is 10 seconds, and there is a 10 second inter-stimulus interval. There are three different stimuli: spiders (Sp), snakes (Sn), claustrophobia (Cl).

The SC of the participant is recorded throughout the experiment. The participants are instructed to continuously update a virtual scale according to their emotional state. This is done in order to collect a continuous subjective reporting of their level of arousal (this analysis is out of the scope of this thesis), and also in order to keep the participant engaged during the experiment. If the participants feel the stimulus is too intense for them, they can stop it at any time by pushing the trigger button.

Figure 1: A screenshot of the virtual room where the stimuli appear. The participant indicates their level of stress with the VR controller and this is displayed in the scale.

(a) Spiders (Sp)

(b) Snakes (Sn)

(c) Claustrophobia (Cl)

Figure 2: Screenshots of the aversive stimuli that were presented to the participant in the VR environment.

2.1.2 Participants

The experiment was conducted on 21 participants in campus (12 male, mean age 27). Basic demographic details from participants, including age, gender, and experience in VR was collected. The experiment was approved by the institutional ethical review board.

2.1.3 The VR Framework

The VR environment was developed in Unity3D (Unity Technologies, USA) using the SteamVR SDK (Valve Corporation, USA). The VR headset used in the experiment is the VIVE VR headset (HTC, USA) including earbuds and one wireless controller (Figure 3). The experimental protocol was implemented in Matlab for this project, and triggers were sent to the Unity software using the user datagram protocol (UDP). SC was recorded using the g.USBamp neurophysiological amplifier (gTec, Austria) and gTec SC sensors.

Figure 3: An illustration of a participant during the VR experiment.

2.2 The RL Simulator experiments

A wide range of experiments were carried out in order to refine the system and test it with increasingly more complex signal properties. In this thesis, four main experiments are described; each experiment description contains the physiological property it attempts to address, the algorithmic approach required, and the experimental results. In the studies described below, the agent's goal is to maximize the physiological arousal obtained using a fixed number of actions, i.e., maximizing the accumulated reward. The immediate reward is derived from the change in SC. States can be used to model additional information, such as memory about current and previous stimuli and physiological values. In our case, the RL agent's actions are performed in the VR environment and the reward is computed from the participant's physiological state by the simulator (Figure 4). The RL algorithms were implemented in Matlab for this project.

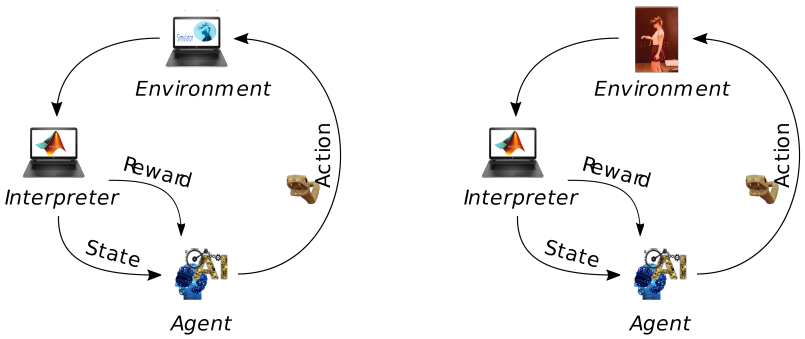

Figure 4: The RL framework: the algorithm is trained with a simulator (left), and after training the system is expected to attempt to modulate the physiological state of a human participant (right) (this is beyond the scope of this thesis).

2.2.1 Experiment 1: From experimental data to simulator

In the first stage we assume a simulated participant whereby the reward for each of the actions – spiders (Sp), snakes (Sn), and claustrophobia (Cl) – is sampled from a normal distribution. The distribution of rewards for each action has been determined by the VR experiment data analysis as follows. For a given participant, each stimulus (action) was presented 10 times. The SC signal is downsampled from 256Hz to 32Hz, and for each event (stimulus presentation) we perform local normalization by:

$$sc[0,15] = SC[0,15] - mean(SC[-5,0]) \tag{2}$$

where $sc[t1,t2]$ is the preprocessed SC level in the temporal window between $t1$ and $t2$ seconds. From this preprocessed SC signal a reward is defined for each appearance of each stimulus by the following formula:

$$R_{i,j} = max(sc_{i,j}[7,15]) - mean(sc_{i,j}[0,7]) \tag{3}$$

where i is the category of stimulus (i.e., snakes, spiders, or claustrophobia) and j is the trial number out of 10 repetitions. The rationale of Equation 3 is to take the reward to be proportional to the magnitude of the peak of the SC after the event normalized by the mean SC during the rest period just before the event. Based on the 10 repetitions we obtain a random variable R_i per participant. Based on this analysis the simulator samples rewards for action i from a normal distribution with the mean and standard variation of R_i.

Figure 5 displays the average response per category of all participants. There are large differences among the participants, and for most participants there are clear differential responses to the different categories, i.e., for a given participant there are typically one or two categories for which the response is strong and one or more categories for which the response is very low. Due to the substantial differences mentioned above, we decided to initialize the simulator based on data from a single participant. This approach aligns with our vision to create a system that can observe and learn individual differences in stimuli response rates. These differences may be quite significant. For example, individuals with anxiety show less physiological habituation (Raskin, 1975).

Figure 6 displays the SC signal of a specific participant, per category. Figure 7 displays the rewards calculated from the preprocessed signal of the same participant, the SC responses to snakes and claustrophobia were higher than the response to spiders, but a 1-way ANOVA statistical test indicates that the differences among the categories are not significant ($p = 0.15$, $F=2.03$).

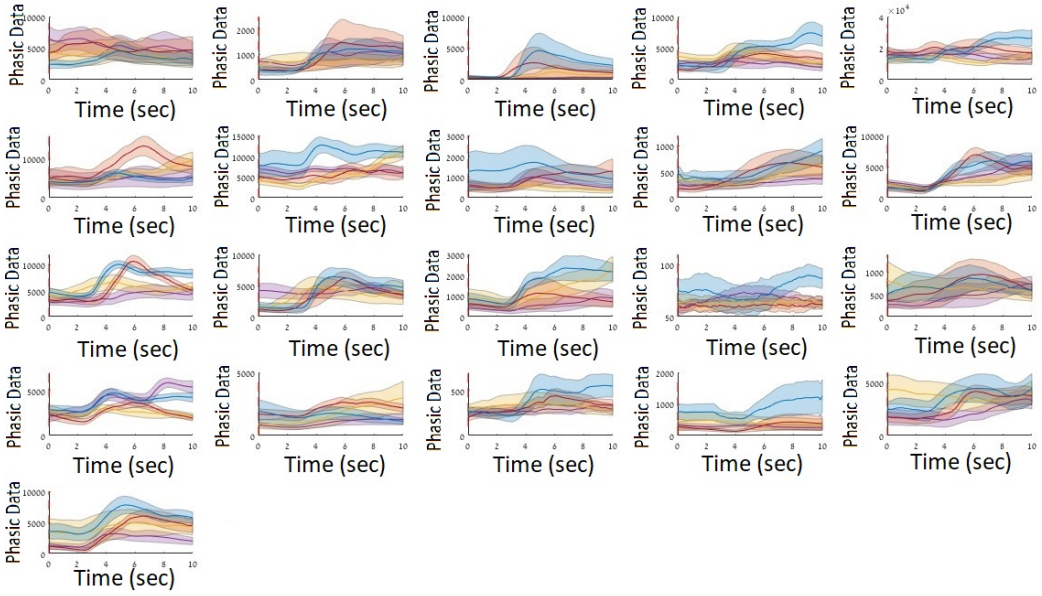

Figure 5: Event related SC response of all 21 participants to four aversive categories. In addition to the three categories discussed in this thesis, this group of participants also experienced a VR simulated fear of heights experience. The plots show the 95% confidence interval, indicating that for most participants there is a clear within-participant difference among the categories, which can be learned by the system.

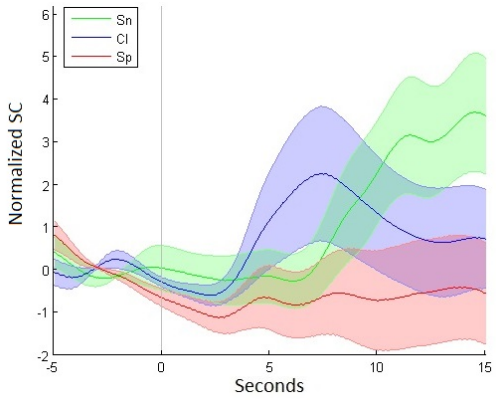

Figure 6: Event related SC response to three categories, of an example participant. Error bars indicate the variance.

The goal of the RL algorithm is to maximize the accumulated reward over time, and in this simple initial experiment this can be framed as a simple 3-arm bandit problem, which can be solved by the state-less Algorithm 1 with $\beta = \frac{1}{N(a)}$. Three methods for estimating the best action probabilities were compared; $adaptive\ pursuit$ (see Algorithm 4), $softmax$ (see Algorithm 3) and $\varepsilon - greedy$ (see Algorithm 2).

2.2.2 Experiment 2: Coping with habituation

In experiment 1 we have assumed that the reward per category is stationary. However, a well known property of most neurophysiological signals is habituation: a decay of response and a decrease in the magnitude of event related responses with repetitions of the same stimulus (e.g. Barry et al., 1993).

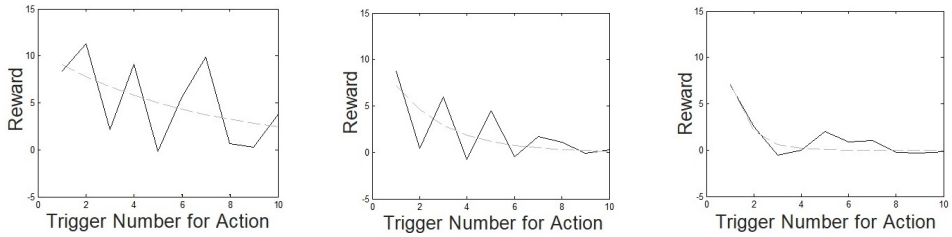

Figure 7: Rewards per category, sample participant: for each category we display the resulting reward over time, based on the category's appearance in the pseudo random presentation order. Left: snakes, middle: claustrophobia, and right: spiders.

Such a trend is also evident in the VR experiment data; in Figure 7 we can see this trend most clearly for the spiders category. The other categories seem to show some habituation but indicate that other sources of temporal variability are present. The habituation trend of each action was modeled to fit the equation $f(x) = e^b \cdot f(x-1)$ using the matlab curve fitting library with one difference – the real participant trend was stretched from 10 stimulus repeats to 50 repeats. From this analysis we can extract for each action $a \in \{Sn, Cl, Sp\}$ an initial reward value C_a and decrease factor η_a as follows.

$$C_{Sn} = 9.0140 \tag{4}$$

$$\eta_{Sn} = 0.9712 \tag{5}$$

$$C_{Cl} = 7.1954 \tag{6}$$

$$\eta_{Cl} = 0.9140 \tag{7}$$

$$C_{Sp} = 7.0967 \tag{8}$$

$$\eta_{Sp} = 0.7804 \tag{9}$$

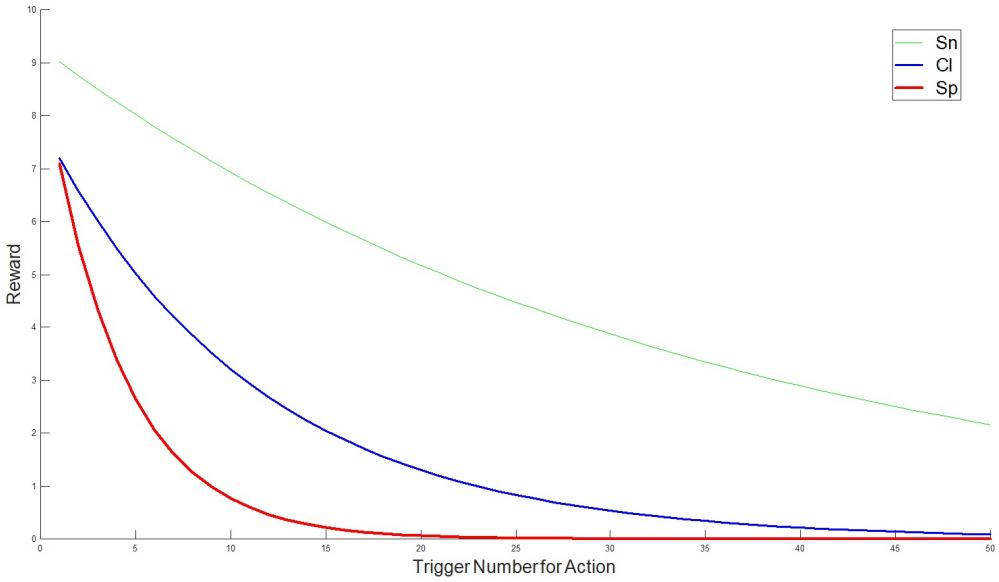

Figure 8: Reward functions extracted from the data of a real participant.

Figure 8, displays the reward functions per category for this example participant. We modeled this behaviour in our simulator as follows. For a given action a we denote the number of steps since the last time the action was taken by δ_a. If the action was never taken $\delta_a = \infty$. The reward $r(t)$ for a specific action at time t is:

$$r(t,a) = \begin{cases} C_a, & \text{if } \delta_a = \infty \\ r(t-\delta_a, a) \cdot \eta_a, & \text{otherwise} \end{cases}$$

In this experiment the reward for performing an action a_i is not affected by any other action $a_{j\neq i}$, but only by δ_a. The optimal policy is thus to start by taking the action with the highest reward, but then to switch to the next most beneficial action when the reward of the first action goes below the initial reward for the second action. Since this implies that the optimal policy can be based on maximizing immediate reward, in this experiment there is no need for the RL to include states.

The goal of the RL algorithm is to maximize the accumulated reward over time, but in contrast to the first experiment the most desirable action changes over time. This can be framed as a non-stationary 3-arm bandit problem, which can be solved by the state-less Algorithm 1 with a small constant β. In this thesis $\beta = 0.3$ was used. The reward value of an action can be reduced dramatically after a small

amount of actions. The declining reward of an action is independent of the other actions that are being performed. Therefore, there is an optimal set of actions for the task, and as long as the size of the set is constant and known, the action order is not important. The habituation problem was handled as an episodic experiment with 20 episodes; each episode was defined as 50 steps (50 actions taken) and each episode started as a new trial as if there was no habituation but with an updated Q-table.

2.2.3 Experiment 3: Leveraging dishabituation

Next, we address the issue of dishabituation (e.g. Siddle, 1985): this phenomenon is complimentary to habituation, and refers to an observed fast recovery of a stimulus response, which was previously diminished by habituation. Dishabituation may be observed when the stimulus is applied after a duration of non-exposure to the stimulus. We model this behaviour in our simulator as follows. For a given action a we denote the number of steps since the last time the action was taken by δ_a, the initial reward value by C_a, the number of presentation steps after which dishabituation occurs by Ω_a, and the decrease factor by η_a. Then the reward $r(t,a)$ for a specific action at time t is:

$$r(t,a) = \begin{cases} C_a, & \text{if } \delta_a \geq \Omega_a \\ r(t-\delta_a, a) \cdot \eta_a, & \text{otherwise} \end{cases}$$

The challenge is for the algorithm to learn **not** to take an action for enough steps, until dishabituation takes place. Unlike experiment 2, the reward reduction of an action depends on the other actions performed. As long as no other action was performed the action reward is reduced, but if other action/s have been performed for at least Ω steps the reward increases back to its initial value. Therefore, an additional state is needed, in order to represent the current dependency of an action reward on other actions that are being performed. The state representation we have used is a captured memory of the most recently presented stimuli (actions taken), and the algorithm selected was SARSA (Algorithm 5). For simplification of the problem it was assumed that $\forall a, \Omega_a = 6$. Given the habituation factor Ω we need $|A|^{\Omega}$ states, where $|A|$ is the number of actions. This representation was easy to implement and it performs better, but it is obviously not scalable because the number of states is exponential in Ω; future work will require more sophisticated mechanisms that can handle the general problem for arbitrary Ω. For comparison of the state-less approach used in experiment 2, the $\varepsilon - greedy$ with $\varepsilon = 0.1$ and

adaptive pursuit method with $Pmin = 0.1$, were also evaluated.

The SARSA algorithm is significantly slower to execute. Therefore, the stateless approach methods were executed and evaluated and only then the best method for this challenge was used as the method to derive a policy from Q in the SARSA algorithm.

In contrast to experiment 2 where the optimal policy was to choose the action with the current best reward, here this policy can yield the worst result. For example, in the case where all initial rewards and decrease factors are equal the above policy would have changed the action taken every step and therefore, would not reach dishabituation. The optimal policy in this case is unknown because the best policy depends on unknown factors such as the initial rewards and decrease factors. In this thesis an ad hoc policy with high accumulated rewards is suggested. In the ad hoc policy, unlike in the RL algorithm, the Ω is known and is being used in order to find the policy. The ad hoc policy algorithm is given in Algorithm 6.

The dishabituation problem was modeled as an episodic experiment with 100 episodes; each episode was defined as 50 steps (50 actions taken) and each episode started as a new trial as if δ_a for all actions was increased to the maximum (no habitation), but with an updated Q-table.

Algorithm 6 Ad hoc policy

1: $a_{current} \leftarrow a_1$
2: $a_{wait} \leftarrow a_1$
3: $\delta_{a_{wait}} \leftarrow 1$
4: $k \leftarrow number\ \ of\ \ actions$
5: $a_{next} \leftarrow (a_{current}\ \pmod k) + 1$
6: $policy(1) \leftarrow a_{current}$
7: **for** *t in 2..T* **do**
8: **if** $\delta_{a_{wait}} > \Omega$ **then**
9: $a_{wait} \leftarrow a_{current}$
10: $a_{current} \leftarrow a_{next}$
11: $\delta_{a_{wait}} \leftarrow 1$
12: $a_{next} \leftarrow (a_{current}\ \pmod k)) + 1$
13: **else**
14: **if** $\delta_{a_{wait}} \neq a_{next}$ **then**
15: $a_{current} \leftarrow a_{next}$
16: $a_{next} \leftarrow (a_{current}\ \pmod k)) + 1$
17: **end if**
18: **end if**
19: $policy(t) \leftarrow a_{current}$
20: **end for**
21: **return** *P*

2.2.4 Experiment 4: Individual differences in dishabituation functions

Figure 5 illustrates that different individuals physiological reaction to each stimulus differ in value and in decrease factor. It is most likely that people differ in the number of presentation steps after which dishabituation occurs, but for simplification of this experiment we assume this factor is identical for all. This experiment is an extension of the previous one, differing only by the initial reward values and decrease factors given to an action in the beginning of a run. In contrast to experiment 3 these values are not based on real participant data but are selected randomly for each run. It is important in order to make sure our results are not specific for one subject initial reward values and decrease factor.

3 Results

In all of the experiments, the results reported are always based on the mean results of 100 runs of the experiment.

3.1 Experiment 1: From experimental data to simulator

Figure 9 presents $\%-optimal$, that is the accumulated reward achieved by the an agent divided by the accumulated reward achieved by always choosing the optimal action (Sn). In this experiment we see that the algorithm can quickly converge on the best action, and the *adaptive pursuit* method with low *Pmin* (0.001) is better than the other methods. Second best is the $\varepsilon-greedy$ method with $\varepsilon = 0.1$.

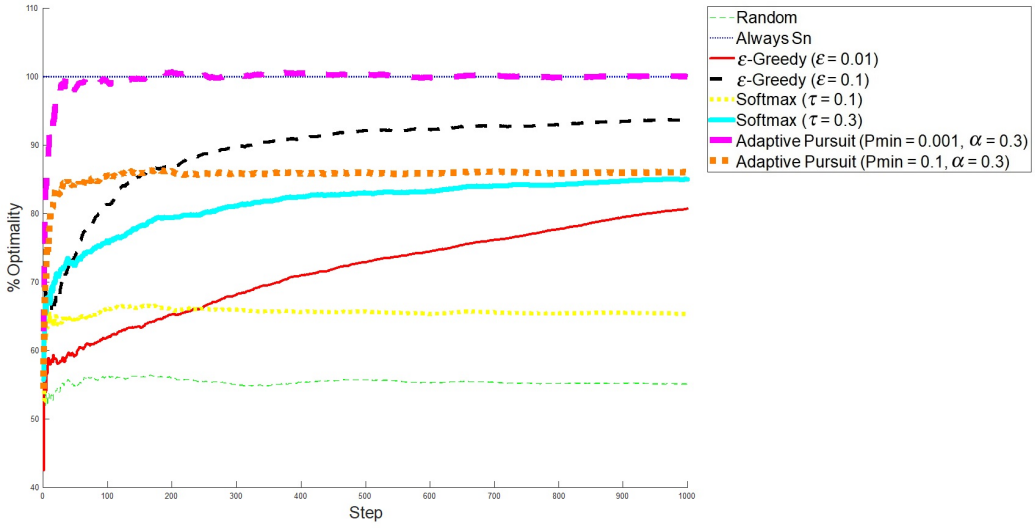

Figure 9: Learning curve of several algorithms in the first experiment. In this case the optimal policy is simply to repeat the action with maximum reward (Sn). The y axis denotes the % of the runs in which the optimal action was selected.

3.2 Experiment 2: Coping with habituation

Figure 10 presents the mean accumulated reward of a run by episodes. The $\varepsilon-greedy$ method, $softmax$ method and $adaptive\ pursuit$ method with $Pmin = 0.1$ accumulated reward value increased from the first episode towards the last

episode, while the method accumulated reward decreased from the first episode towards the last episode.

The results indicate that $\varepsilon - greedy$ with $\varepsilon = 0.1$ and the *adaptive pursuit* method with $Pmin = 0.1$ rapidly converge to near optimal results.

In this experiment there is an optimal set of actions to take in order receive the optimal accumulated reward. This set of action learned by the optimal policy is: in 36 steps choose Sn, in 10 steps choose Cl and in 4 steps choose Sp. In Figure 11 we see the mean number of times an action was performed in an episode for random, optimal and the first and last episode for the two best methods. This gives us an idea of the set of actions each agent learned to take. We can see that after the first episode the set of actions performed by the *adaptive pursuit* agent with $Pmin = 0.1$ is the most similar to the optimal agent. As for the last episode the set of actions performed by the $\varepsilon - greedy$ with $\varepsilon = 0.1$ agent is almost the same as that of the optimal agent. The information from the figure indicates that the set of actions performed by the $\varepsilon - greedy$ with $\varepsilon = 0.1$ agent is the most similar to the optimal agent given enough episodes.

Combining the results, it seems the $\varepsilon - greedy$ with $\varepsilon = 0.1$ agent and the *adaptive pursuit* agent with $Pmin = 0.1$ can both cope with habituation over a series of episodes, better than all other agents suggested.

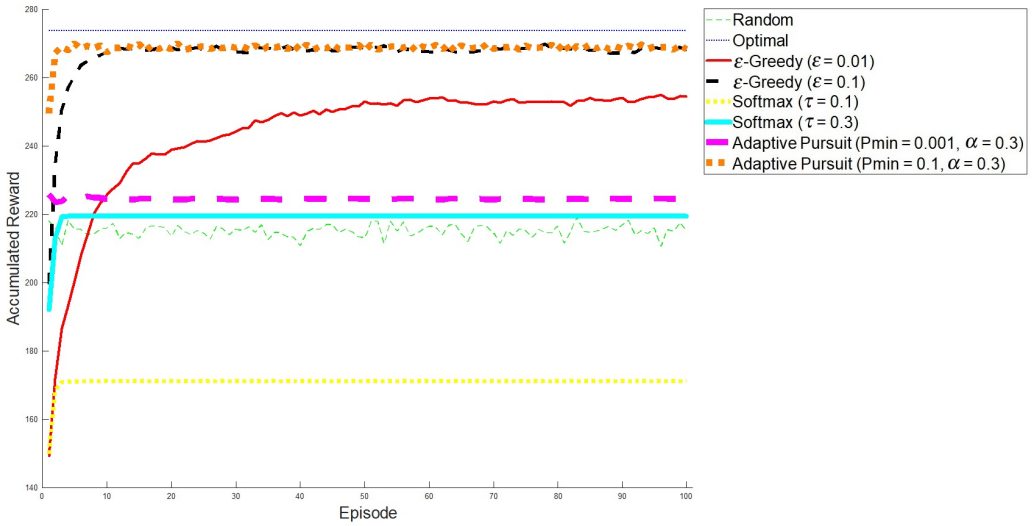

Figure 10: The learning curve (the mean accumulated reward of a run by episodes) obtained with the reward function based on real participant data, by the different algorithms.

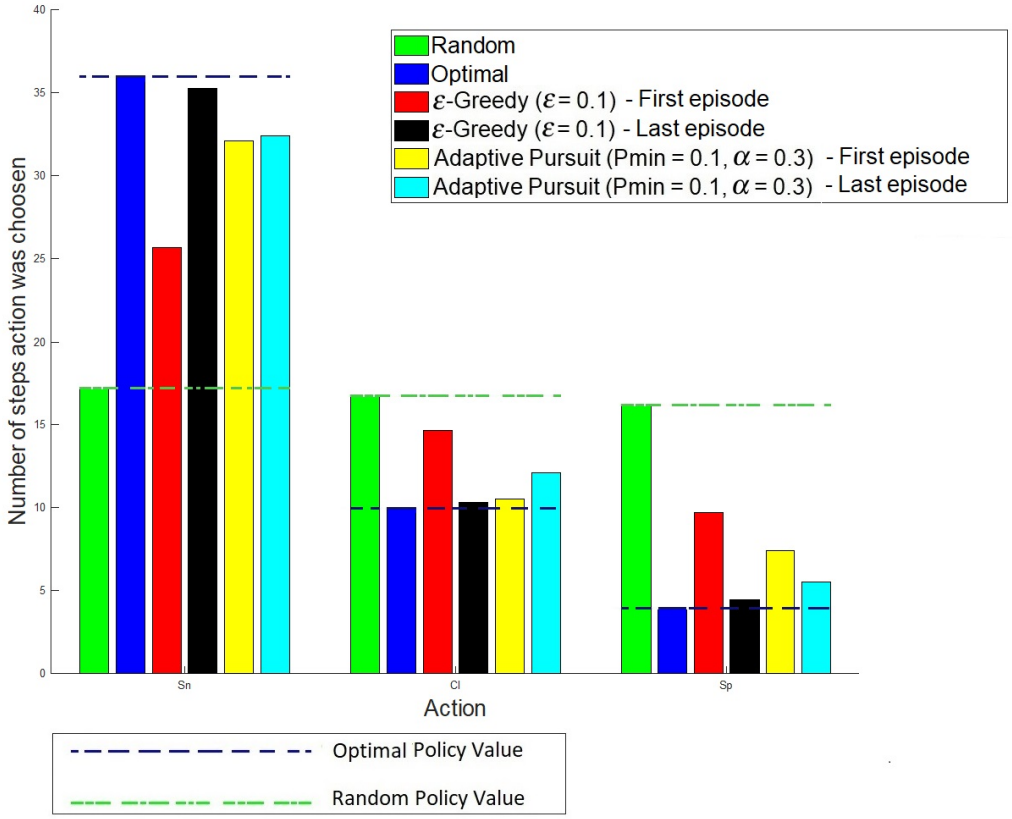

Figure 11: The mean number of steps that each action was selected in an episode. The optimal set of actions is the one of the optimal agent.

3.3 Experiment 3: Leveraging dishabituation

Figure 12 presents the mean accumulated reward of a run by episodes. The $\varepsilon - greedy$ algorithm with $\varepsilon = 0.1$ rapidly converges to a maximum value and provides better results than the random policy and the *adaptive pursuit* algorithm. Therefore, we have chosen the $\varepsilon - greedy$ algorithm for the method used for deriving a policy from Q in the SARSA algorithm. The SARSA agent needs more episodes in order to learn a better policy than random, but from approximately episode 25 obtains a better accumulated reward than the $\varepsilon - greedy$ agent. Figure 13 displays the mean of the ongoing accumulated reward during a session with dishabituation, at the last episode. The $\varepsilon - greedy$ with $\varepsilon = 0.1$ agent performs better than the random policy agent only from approximately step 30. The

SARSA agent accumulated rewards, in contrast, are more similar to the ad hoc policy results than to the random policy results.

The results indicate that the *SARSA* agent learns better than the random agent and than the $\varepsilon - greedy$ agent when given enough episodes. Additionally, it indicates that the $\varepsilon - greedy$ agent with $\varepsilon = 0.1$ rapidly converges to a maximum value and obtains better results than the *adaptive pursuit* algorithm and the random policy, but not by much.

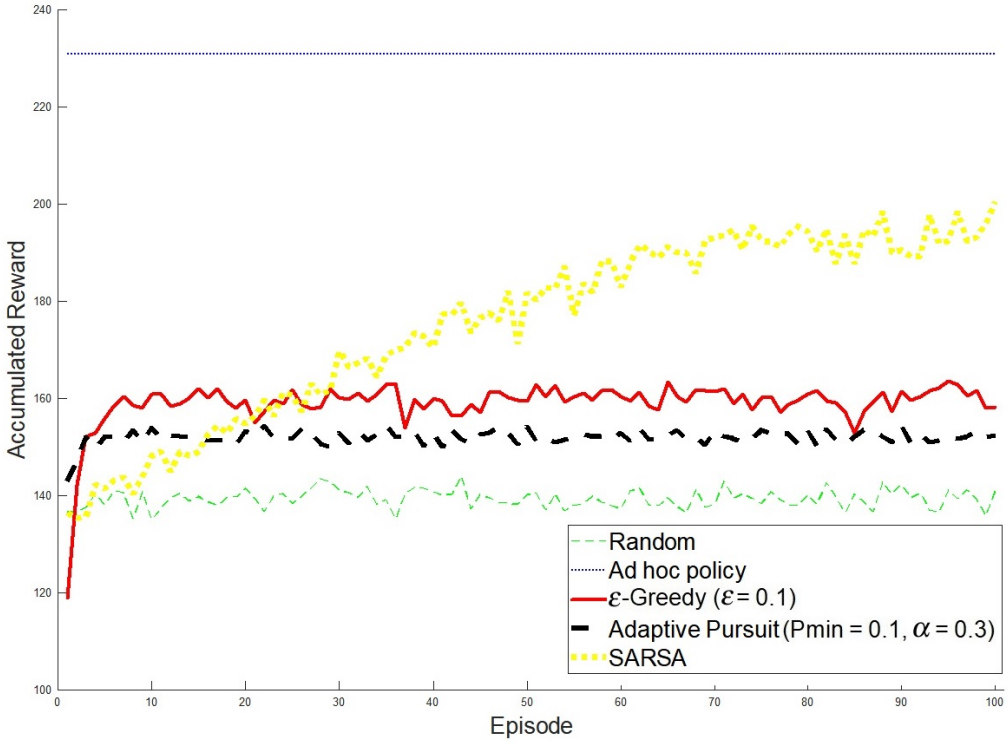

Figure 12: The learning curve obtained with the reward function with dishabituation based on real participant data.

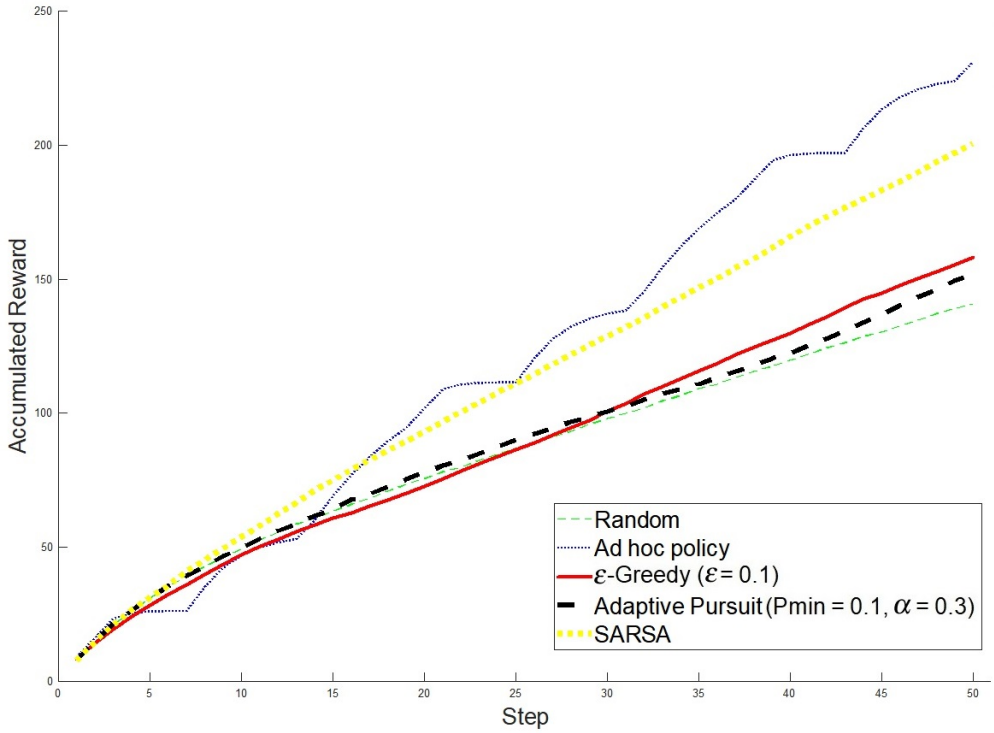

Figure 13: The mean of the ongoing accumulated reward during a session with dishabituation, at the last episode.

3.4 Experiment 4: Individual differences in dishabituation functions

Figure 14 presents the mean accumulated reward of a run by episodes. Figure 15 displays the mean of the ongoing accumulated reward during a session with dishabituation, at the last episode.

The results indicate that both RL agents learn independently to the initial reward values and decrease factors of the reward functions. Moreover, the *SARSA* agent learns better than the random agent and the $\varepsilon - greedy$ agent when given enough episodes, and the $\varepsilon - greedy$ agent with $\varepsilon = 0.1$ rapidly converges to a maximum value and obtains better results than the random policy, but not by much.

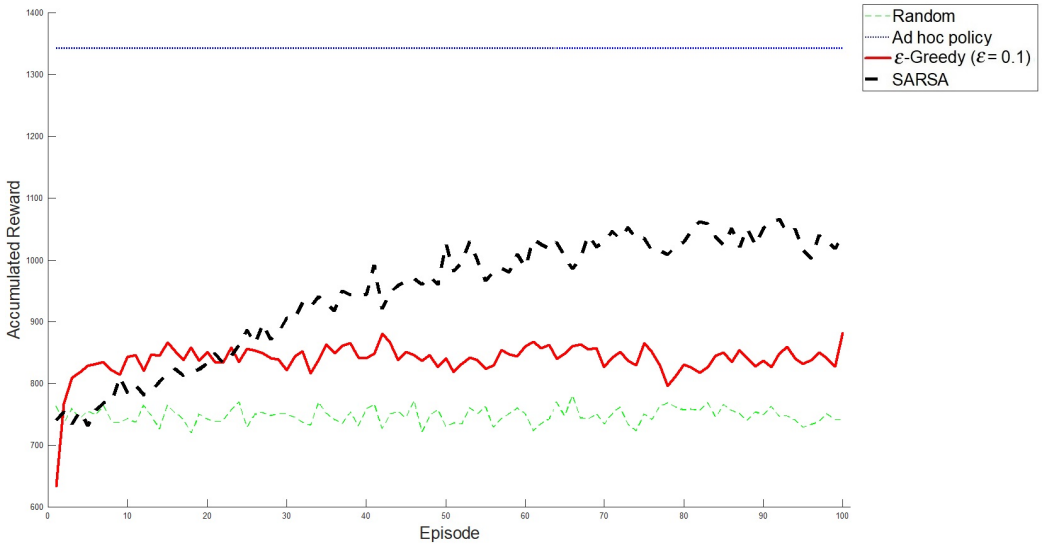

Figure 14: The learning curve obtained with the reward function with dishabituation.

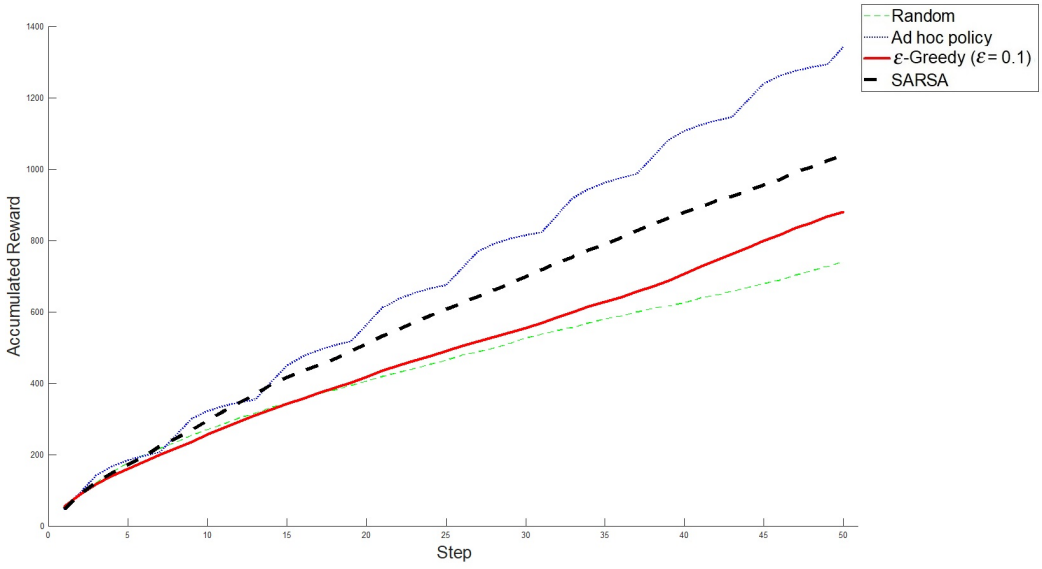

Figure 15: Ongoing reward during a session with dishabituation, after 100 episodes.

4 Discussion and Future Work

This thesis described an overall framework for physiological state induction and illustrated how physiological data from empirical studies can be captured by a simulator for RL training. A simplified example with three VR stimuli was explored, and it was shown how RL algorithms can be used to find near optimal policies, coping with some of the challenging properties of physiological signals, such as noise, habituation, and dishabituation.

Our results indicate that although the adaptive pursuit approach was the best algorithm to deal with the noise property, this was not the case with habitation and dishabitation. When dealing with habitation there was a small advantage for an $\varepsilon - greedy$ method when given enough episodes. When adding the non-stationary problem of selecting actions to maximize physiological arousal – given both habituation and dishabituation – the addition of different states gave the agent a clear advantage over the state-less algorithms. Our results indicate that a simple SARSA implementation can solve this problem. Moreover, this is still the case even if we introduce individual differences in the initial reward values and decrease factors given to an action. Nevertheless, our approach is limited to short term dishabituation (arbitrarily selected to 6 steps; $\Omega = 6$). The number of states is exponential in the number of steps, so a more general solution would probably require generalization to be applied to the Q table; i.e., instead of a tabular format the Q function would have to be represented compactly using function approximation.

Clearly, there is much more to be done from these initial steps to a concrete successful application. The next step will need to address the key question: can the overall framework be used to actually induce physiological state in VR–immersed human participants (as compared with carefully selected control conditions)? And if so, how effective is this scheme, and does it have any advantages over alternative, more simple approaches? If the results with human participants (ongoing) are promising, there are still several computational challenges to address in applying the method. One of the challenges will be to show how learning a generic simulator can be quickly transferred to different individuals with different habituation and dishabituation parameters. Obviously, scaling the method to deal with richer, more realistic, scenarios will require additional steps.

If successful, there is a wide range of long term applications for our framework: combined with various neurophysiological signals, the method can be used as part of various psychotherapeutic, entertainment or training VR scenarios, as well as to serve as a basis for new types of adaptive environments.

5 References

Barry RJ, Feldmann S, Gordon E, Cocker KI, Rennie C. 1993. Elicitation and habituation of the electrodermal orienting response in a short interstimulus interval paradigm. *International Journal of Psychophysiology*. 15:247–253.

Benedek M, Kaernbach C. 2010. A continuous measure of phasic electrodermal activity. *Journal of Neuroscience Methods*. 190:80–91.

Critchley HD. 2002. Electrodermal responses: What happens in the brain. *The Neuroscientist*. 8:132–142.

Höök K. 2009. Affective loop experiences: Designing for interactional embodiment. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*. 364:3585–3595.

Kaelbling LP, Littman ML, Moore AW. 1996. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*. 4:237–285.

Koulouriotis DE, Xanthopoulos A. 2008. Reinforcement learning and evolutionary algorithms for non-stationary multi-armed bandit problems. *Applied Mathematics and Computation*. 196:913–922.

Liu C, Agrawal P, Sarkar N, Chen S. 2009. Dynamic difficulty adjustment in computer games through real-time anxiety-based affective feedback. *International Journal of Human-Computer Interaction*. 25:506–529.

Marín-Morales J, Higuera-Trujillo JL, Greco A, Guixeres J, Llinares C, Scilingo EP, Alcañiz M, Valenza G. 2018. Affective computing in virtual reality: Emotion recognition from brain and heartbeat dynamics using wearable sensors. *Scientific reports*. 8:13657.

Picard RW. 1995. Affective computing. *MIT Technical Report*. 321.

Raskin M. 1975. Decreased skin conductance response habituation in chronically anxious patients. *Biological Psychology*. 2:309–319.

Rovira A, Slater M. 2017. Reinforcement learning as a tool to make people move to a specific location in immersive virtual reality. *International Journal of Human-Computer Studies*. 98:89–94.

Siddle DA. 1985. Effects of stimulus omission and stimulus change on dishabituation of the skin conductance response. *Journal of Experimental Psychology: Learning, Memory, and Cognition*. 11:206.

Slater M, Sanchez-Vives MV. 2016. Enhancing our lives with immersive virtual reality. *Frontiers in Robotics and AI*. 3:74.

Sutton RS, Barto AG. 1998. Reinforcement learning: An introduction. MIT press Cambridge.

Tijs T, Brokken D, IJsselsteijn W. 2008. Creating an emotionally adaptive game. Springer, pp. 122–133.

Wu D, Courtney CG, Lance BJ, Narayanan SS, Dawson ME, Oie KS, Parsons TD. 2010. Optimal arousal identification and classification for affective computing using physiological signals: Virtual reality stroop task. *IEEE Transactions on Affective Computing*. 1:109–118.

תקציר

על מנת ליצור אינטראקציות אדם-מכונה עמוקות ומשמעותיות יותר יש יתרון ליכולת המערכת להבין, ללמוד ולהגיב למצב הרגשי של משתמש הקצה ברציפות ולאורך זמן. על מנת לזהות שינוי בעוצמת התגובה הרגשית של המשתמש ניתן להיעזר בקשר בין מצב רגשי למדדים פיזיולוגים כגון מוליכות עורית, דופק לב ועוד. על מערכת כזו להיות מסוגלת להתאים את עצמה בזמן אמת לתגובות המשתמש לגירויים, כולל הסתגלות המשתמש לגירויים החוזרים על עצמם, דבר המוביל לירידה בעצמת התגובה (הביטואציה), וייתכן שגם עליה מחודשת בעצמת התגובה לגירוי (דיסהביטואציה). בביופידבק מסורתי, אנו נעזרים במדדים פיזיולוגים כדי לתת משוב מתמשך למשתמש ועל ידי כך מאפשרים לו ללמוד לווסת את המצב הפיזיולוגי שלו. בעבודה זו אנו מציגים פורמט משלים, לפיו המכונה היא זו שצריכה ללמוד כיצד לווסת את המצב הפיזיולוגי של המשתמש, על ידי בחירה מתמשכת של רצף הגירויים היעיל ביותר.
מחקר זה מהווה את הצעדים הראשונים לקראת פיתוח מערכת אינטליגנטית זו בדגש על העלאה (או הורדה) של רמת העוררות בקרב משתמשים השוהים בסביבת מציאות מדומה. ראשית, פותחה סביבת מציאות מדומה בה המשתמש נחשף לסדרת גירויים המופרדים על ידי זמני רגיעה קבועים, כאשר כל גירוי ידוע כטריגר לאחד משלושה פחדים נפוצים. בשלב הראשון של הניסוי הוקלטה המוליכות העורית (SC) של 21 נבדקים אשר נחשפו לסדרה של 30 גירויים בסביבה הנ"ל (10 גירויים זהים לכל פחד כאשר הסדר נבחר רנדומלית). בהתבסס על הנתונים מהשלב הראשון, כמו גם הממצאים המדווחים בספרות הפסיכופיזיולוגית, פותח לשלב השני של הניסוי סימולטור המבוסס על תגובות פיזיולוגיות לגירויים בסביבת המציאות המדומה. סימולטור זה שימש כדי להכשיר מערכת למידת מכונה מסוג למידת חיזוק ($Reinforcement\ Learning$), שמטרתה ללמוד מדיניות אופטימלית להפקת רמות גבוהות של עוררות פיזיולוגית. הגדרנו את הצורך להתמודד עם רעש בהקלטת מוליכות עורית ואת הסתגלות המשתמש לגירויים (הביטואציה) כאתגרים שעל המערכת לפתור.
ניתן להסתכל על שתי הבעיות הנובעות מאתגרים אלו כבעיות $3-arm\ bandit$, כאשר הראשונה היא בעלת ידית מועדפת קבועה והשניה בעלת ידית מועדפת משתנה. לצורך פתרון הבעיות ניתן להשתמש באלגוריתם פשוט של למידת חיזוק חסרת מצבים ($state-less$). שלושה סוכני למידת-חיזוק ידועים $\varepsilon-greedy, softmax$ ו- $adaptive\ pursuit$ מומשו, ונבדקה התאמתם לפתרון הבעיות. סוכנים אלו נבדלים זה מזה בגישתם לדילמת ניצול ידע קיים מול חקירה נוספת. באתגר הרעש סוכן בשיטת $adaptive\ pursuit$ הגיע לערכים הגבוהים ביותר. גם הסוכן שהשתמש בשיטת $\varepsilon-greedy$ פתר את הבעיה בתוצאות עם ערך גבוה. באתגר ההביטואציה היה יתרון לסוכן בשיטת $\varepsilon-greedy$ ו- $adaptive\ pursuit$.
אתגר נוסף שנבדק בתזה, הינו שילוב דיסהביטואציה - הופעה מחדש של התגובה המקורית לגירוי. בחנו בנוסף לסוכנים בשיטת $\varepsilon-greedy$ ו- $adaptive\ pursuit$, שהשיגו את התוצאות הטובות ביותר בהתמודדות עם הביטואציה, גם סוכן קלאסי למידת חיזוק בעל מצבים מוגדרים (states) - SARSA. מצב הסוכן הוגדר על ידי זיכרון הגירויים האחרונים שהוצגו. התוצאות מצביעות על כך ששני הסוכנים פועלים לפי פוליסה עדיפה מהאקראית, וכי סוכן SARSA השיג את התוצאות עם הערך הגבוה ביותר בטווח הארוך.

אני רוצה להביע את תודתי הכנה לפרופ' דורון פרידמן, אשר עבודה זו בוצעה בהדרכתו, על שנתן לי את החופש לחקור את הצעדים הראשונים של אינטראקצית אדם-מכונה אינטליגנטית המבוססות על אותות פיזיולוגיים אנושיים ומעבר. הכרת תודתי גם ליונתן גירון על תמיכתו והשקעתו לאורך כל הניסוי בשלב המעבדה, מסיוע בתפעול ציוד המעבדה ועד לניתוח תוצאות וכתיבת מאמרים בעקבות זאת. כמו כן, ברצוני להביע את תודתי לאסף שמלה ולדבי שטרית על שעזרו לי לנהל את הניסוי עם הנבדקים. תודה מיוחדת לבן זוגי אורי פלג, על תמיכתו הרבה, על סבלנותו ועל אמונתו בעבודתי. לבסוף, אני מודה לאירית שלזינגר-ברקרס וקרלה פלג, אמי וחמותי, על עזרתם ועל משחקם עם בני בחלקים המאתגרים יותר של מחקר זה. למרות שהן אמרו לי שהן ברות מזל, אני גם ברת מזל שיש לי כוח סבתות חזק ותומך.

בית ספר אפי ארזי
למדעי המחשב
הבינתחומי
הרצליה

המרכז הבינתחומי בהרצליה
בית-ספר אפי ארזי למדעי המחשב
התכנית לתואר שני (M.Sc.) - מסלול מחקרי

אינדוקציה אלגוריתמית של מצבים פיזיולוגים: צעדים ראשונים

מאת
ברקרס קרן-אור

עבודת תזה המוגשת כחלק מהדרישות לשם קבלת תואר מוסמך M.Sc.
במסלול המחקרי בבית ספר אפי ארזי למדעי המחשב, המרכז הבינתחומי הרצליה

מאי 2019