A Multimodal Adaptive Session Manager for Physical Rehabilitation Exercising

Konstantinos Tsiakas
HERACLEIA Lab
Computer Science and Engineering Department
University of Texas, Arlington
konstantinos.tsiakas@mavs.uta.edu

Manfred Huber
Computer Science and Engineering Department
University of Texas, Arlington
huber@uta.edu

Fillia Makedon
HERACLEIA Lab
Computer Science and Engineering Department
University of Texas, Arlington
makedon@uta.edu

ABSTRACT

Physical exercising is an essential part of any rehabilitation plan. The subject must be committed to a daily exercising routine, as well as to a frequent contact with the therapist. Rehabilitation plans can be quite expensive and time-consuming. On the other hand, tele-rehabilitation systems can be really helpful and efficient for both subjects and therapists. In this paper, we present ReAdapt, an adaptive module for a tele-rehabilitation system that takes into consideration the progress and performance of the exercising utilizing multisensing data and adjusts the session difficulty resulting to a personalized session. Multimodal data such as speech, facial expressions and body motion are being collected during the exercising and feed the system to decide on the exercise and session difficulty. We formulate the problem as a Markov Decision Process and apply a Reinforcement Learning algorithm to train and evaluate the system on simulated data.

Categories and Subject Descriptors

I.2.6. [**Artificial Intelligence**]: Learning

Keywords

Multimodal Adaptive Systems, Reinforcement Learning, Markov Decision Process, Personalized Rehabilitation Systems.

1. INTRODUCTION

Physical Activity (PA) and Physical Training (PT) are essential parts of any rehabilitation plan and have been shown to promote health, slow disease progression and improve the activities in daily living (ADL) [1,2]. Improving motor abilities, including body function and physiological recovery require long rehabilitation plans and engagements from both the subject and the therapist [3,4].

Physical Exercising has been proven to also be a therapeutic intervention to prevent brain damage associated with neurodegenerative diseases and chronic diseases [5], such as Rheumatoid Arthritis (RA) and fibromyalgia (FM). A challenging point of any rehabilitation plan is to keep the subject engaged in a daily exercising routine, maximizing subject's compliance to the regimen, while keeping them safe from harmful injuries during exercising. In order to achieve such requirements, a personalized and adaptive rehabilitation plan is required [6]. Rehabilitation session exercises have to be continued at a level of difficulty that is adapted to the subject's changing physical and mental capabilities over time. The exercise regimen can vary widely from case to case, depending on the subject's physical and psychological condition, age, medication, and many other factors including knowledge and support for exercise programs.

A significant burden on rehabilitation plans is that they can be really expensive and time-consuming, especially when the subject needs a long-term treatment. Recent advances in tele-rehabilitation systems [7, 8, 9] using advanced technologies and virtual reality (VR) systems have been proposed to overcome these burdens. VR technologies combined with multimodal sensing systems are being examined to be used to transform the traditional physical exercising and give the opportunity to the subject to perform the prescribed exercises from their own environment, while giving essential feedback and data to the therapist to enhance their decision making [10,11,12]. An example of such a telerehabilitation system that uses avatars and sensors to monitor the subject's performance is shown in Fig.1.

Figure 1. A tele-presence rehabilitation session that shows the remote therapist, the subject and an avatar that represents the subject seen inside the inner frame. (A. Hagman, Swedish Health Care, RoboBusiness Leadership Summit, 2012).

Such systems collect and process data from various sensors to monitor the performance of the subject while exercising. Answers to basic research questions for such systems lie in the areas of

PETRA '15, July 01-03, 2015, Corfu, Greece
© 2015 ACM. ISBN 978-1-4503-3452-5/15/07…$15.00
DOI: http://dx.doi.org/10.1145/2769493.2769507

computer vision, data analysis, graphics, avatar design, dialogue systems and human-computer interaction.

Such a system must be dynamically adaptive, utilizing meaningful data from the subject during the rehabilitation session. Subject preferences and performance must be monitored to maintain an appropriate exercise difficulty level. Due to the repetitive nature of most of the rehabilitation exercises, the subject is not easily engaged over longer periods of time and more likely to quit. This is especially true if the difficulty level of the exercise is predefined and static. The subject should take initiative and remain motivated to intensively training at home [13].

Making the exercise regimen adaptive in real time is the biggest computational challenge. ReADAPT utilizes multisensory data such as body motion data, facial pain expressions, and speech as well as session information such as time spent on each exercise and the performance on the current exercise difficulty level. All this information is used as feedback to the system to modify and adapt the exercise category and difficulty. The purpose of such a system is to ensure that subjects are consistently and correctly performing their rehabilitation exercises according to the specified, by the therapist, protocol at home under the (remote) supervision of a therapist, while monitoring the pain levels to ensure safety and compliance.

ReADAPT provides the therapist with a general framework to define a specific exercise protocol and the various system parameters based on the subject's needs and personalized plan. It enhances the therapist's decision making by suggesting the exercise difficulty level, in order to achieve a high performance and subject compliance to the exercising plan.

2. RELATED WORK

Recent works have proposed adaptive systems to enhance the human performance or monitor the execution of exercising during rehabilitation sessions. Researchers have proposed the use of robotic platforms to assist the subject perform specific tasks. In [14] they propose ADAPT, a robotic task-practice system that enhances the recovery of upper extremity functions in post-stroke subjects. They present a task scheduler that decides which task the subject must perform based on his previous performance. They focus on the robot dynamics in order to simulate daily living functional tasks, such as opening a doorknob, opening a jar, turning a key, etc.

In [15, 16] a home-based mixed reality rehabilitation system is proposed. In [15], they presented HAAMR, a system to restore motor function for chronic post-stroke survivors by providing an engaging long-term limb-reaching task therapy at home. It allows participants to complement their therapy by training reaching movements, reach-to-touch, and reach-to-grasp tasks over multiple months of therapy at home. They utilize multimodal sensing methods for measuring hand trajectory, hand speed, target manipulation and torso movement while the user performs the task with certain objects.

Other systems have formulated the rehabilitation procedure as a fully or partially observable Markov Decision Process (POMDP). In [17], POMDP was used for the decision-making module to automatically modify the exercise parameters using a robotic system that guides stroke subjects through an upper-limb reaching task. The system uses motion range data, learn rate and time, as well as fatigue levels of the subject while performing specific reaching rehabilitation tasks. In [18], a reaching task is modeled as a trajectory in the state space of hand part features using MDP and a learning reward.

In previous work, we have proposed a theoretical framework of a system that uses a dialogue system able to interact naturally with the subject in order to ensure exercise safety and monitor the subject's performance during the session [19]. An important attribute of this system is that it allows natural communication between the user and the system, using audiovisual automatic speech recognition [20] and Natural Language Understanding (NLU). The patient subject can interact with the system to provide a self-report on comfort and pain levels in order to ensure safety and the system uses this feedback to adapt the session to make it safe.

In this paper, we present a framework for ReADAPT, the adaptation module which is responsible for the decision-making and the personalization of the rehabilitation session, based on the multisensing input that is collected during the rehab session. We define the problem as an optimization problem, using the Markov Decision Process formulation and apply Reinforcement Learning algorithms for the system training.

3. BACKGROUND

Markov Decision Process (MDP) models provide a mathematical framework for modeling decision making in systems where the outcomes are random. They are an extension of Markov chains, with the addition of *actions* (choices) and *rewards* (environmental feedback). A Markov Decision Process is a 5-tuple $\{S, A, R, T, \gamma\}$, where S is a non- empty set of states s, A is a non- empty set of actions α, R is the reward function which is expressed either in terms of state $R(s)$ or action-state pair $R(s, a)$; T is the transition model, which gives the probability $P(s' \mid s, a)$; the probability that action $a \in A$ in state $s \in S$ will lead to state $s' \in S$, and $\gamma \in [0,1]$ is the discount factor, which represents the difference in importance between future rewards and present rewards.

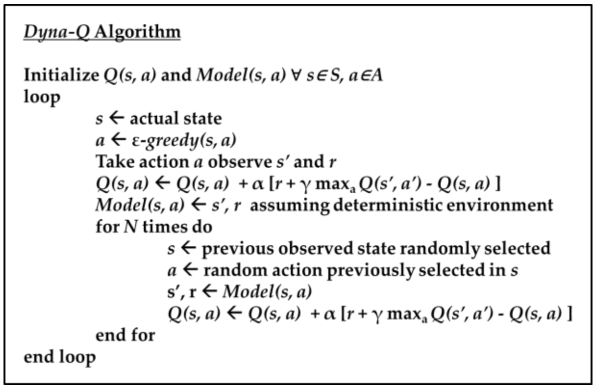
Of interest is to calculate the cumulative rewards as collected during the transitions from state to state until reaching a final state. In most problems that are modeled as an MDP, we are interested in maximizing the cumulative discounted reward or *return*. We therefore need to select an action for each state that will (in the future) maximize the return. An MDP policy π is a mapping that dictates which action to take at each state. More formally, it is a mapping of each state and action to the probability $\pi(s,\alpha)$ of taking action α in state s. The goal is to find an optimal policy, through a trial-and-error process of repeated interaction with the user. An optimal policy is a policy that maximizes the cumulative discounted expected reward.

Reinforcement Learning (RL) has been extensively applied to problems where a system or *agent* must learn behavior through trial-and-error interactions with a dynamic environment [21]. Such systems can learn the optimal policy based on the environmental feedback. One of the strongest points of this approach is that the system learns from experience. In this way, the system can adapt to its environment and find an optimal policy. During the interactions, the system updates the state values $V(s)$, that represent the total expected accumulated rewards starting at state s, or state-action values $Q(s,\alpha)$, that represent the total expected accumulated reward starting at state s and taking action a.

The agent must explore the environment to gather information about which actions can lead to the optimal policy. We need to both explore the environment for unexplored areas (*exploration*) and use the existing knowledge to make better decisions (*exploitation*). In this sense, there must be a balance between exploration and exploitation. To gain more rewards, the agent must follow the actions that it knows will lead to high immediate rewards, while it also needs to explore more states to gain knowledge about the environment.

Two basic approaches for RL solving are *off-policy* and *on-policy* strategies. An *off-policy* learner learns the value of the optimal policy independently of the agent's actions. It can update the estimated value functions using hypothetical action, which may not have actually been used. An *on-policy* learner learns the value of the policy being carried out by the agent, including the exploration steps. The most common exploration techniques are *ε-greedy* and *softmax*. In many cases the exploration strategy depends on the time the agent has interacted with the environment or the successes during the interactions [22].

Policy learning has been approached using two different forms of reinforcement learning that are referred to as *model-based* and *model-free* learning [23]. Their difference is that in model-based learning, the system uses training data to build a complete model of the environment and the optimal policy and the value function can be found off-line, whereas in model-free learning the agent is not aware of the model and learns the value function and the policy only by experience (interaction data). Model-based approaches are often referred as *planning*. In model-based learning, when the model is inaccurate, the planning process will compute a suboptimal policy. The solution is either a model-free approach or to reason explicitly about model uncertainty [22]. However, an integration of real and simulated experience can overcome these problems and the leaner can converge to an optimal policy. This architecture is called *Dyna architecture* and *Dyna-Q* is an algorithm that follows this architecture [24]. It learns a model from real experience and learns the value function and policy from real and simulated experience. Dyna-Q randomly selects previously visited state-action pairs to perform simulated steps and update the Q-values until it converges to an optimal policy.



```
Dyna-Q Algorithm

Initialize $Q(s, a)$ and $Model(s, a)$ $\forall s \in S, a \in A$
loop
    $s \leftarrow$ actual state
    $a \leftarrow \varepsilon\text{-}greedy(s, a)$
    Take action $a$ observe $s'$ and $r$
    $Q(s, a) \leftarrow Q(s, a) + \alpha\, [r + \gamma \max_a Q(s', a') - Q(s, a)]$
    $Model(s, a) \leftarrow s', r$ assuming deterministic environment
    for $N$ times do
        $s \leftarrow$ previous observed state randomly selected
        $a \leftarrow$ random action previously selected in $s$
        $s', r \leftarrow Model(s, a)$
        $Q(s, a) \leftarrow Q(s, a) + \alpha\, [r + \gamma \max_a Q(s', a') - Q(s, a)]$
    end for
end loop
```

Figure 2. The Dyna-Q algorithm

During planning, the Dyna-Q algorithm randomly samples only from the set of state-action pairs that have previously been experienced, so the model is never queried with a pair about which it has no information. Planning is achieved by applying reinforcement learning methods to the simulated experiences just as if they had really happened. Dyna-Q is a useful algorithm because it can converge much faster, since it performs simulated updates for each actual episode. It also can be efficient even if it starts with an inaccurate model, since it learns a model from real experience.

4. THE ADAPTIVE SESSION MANAGER AS A MARKOV DECISION PROBLEM

In this section, we describe the problem formulation as a Markov Decision Process (MDP). We define the state space, the action space and the reward function. We give a detailed description of the system and the MDP definition.

4.1 System Description

As already mentioned in previous sections, the goal of ReADAPT is twofold; to engage the subject to complete the whole session of the exercises while preventing him/her from injuries (*maintain safety*) and high pain levels (*maintain compliance*). In order to achieve this, the manager must learn the optimal policy *π;* which action is best at each state, through real or simulated interactions. Each interaction is called here a *session*, and describes the execution of the exercises that the system prompts to the user until it reaches a final state. A final state on this system can be either a *goal state*, where the subject completes successfully the whole set of exercises, or a *quit state*, where the subject decides to quit. We follow the strong assumption that the subject is more likely to quit or be non-compliant under high levels of pain [25].

During each session, the manager uses the multimodal input (speech, facial expressions, body motion, and session information) at time t to define the current state s_t. Based on this multimodal input, the manager performs a specific action a_t and observes the new state s_{t+1} and the reward r_t based on the environmental feedback. In other words, the manager may prompt the subject to perform the same exercise at a higher difficulty level and receive feedback, i.e., multisensory subject data, that the subject does not perform the exercise correctly. The manager receives a numerical reward that describes how good the selected action was and updates its knowledge for this specific state-action pair, by updating the Q-value. The system architecture is shown at Fig. 3.

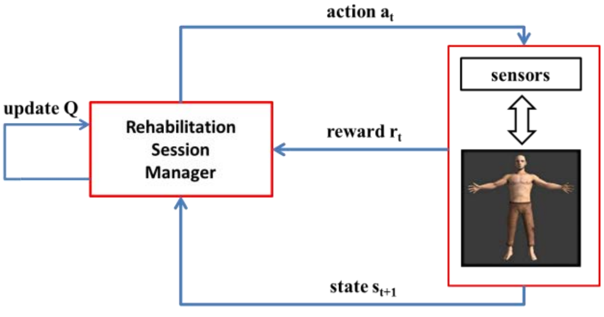


Figure 3. System Architecture

4.2 State Space

Our approach follows the following scenario: The subject must complete a set of three prescribed exercises. Each exercise has three difficulty levels (Normal- 1, Medium- 2, Easy- 3). Each exercise starts at the Normal level. The system keeps track of the time spent on each exercise using time units (1- 5). Each time the same exercise is performed at any difficulty, the time unit is increased by one. At this representation, time unit 1 is the less time that the subject will spend on this exercise, while 5 indicates that the subject has spent too much time on the same exercise. The specific mapping from actual time to time units shall be defined by a professional therapist.

During the exercise performance, the system measures the exercise performance compared to the ground truth given by the therapist, by analyzing the body motion capture of the subject (1- Good, 2- Medium, 3- Bad). Also a pain detection module is used to decide if there is a facial pain expression or not (0- No, 1- Yes). An important state feature is the *user pain report*. The system learns to prompt the subject, asking for comfort and pain levels, aiming to get an explicit answer. Responses are translated into a numeric value (0-11), according to the Numeric Rating Scale (NRS-11), an 11-point scale for subject self-reporting of pain [36]. To our knowledge, few related works combine audiovisual feedback from the user to express their pain or fatigue levels and the interest in these works has focused for example in emergency response systems [26,27].

Collecting such information during real interactions can be very useful for improving the system's pain detection, accuracy, due to the challenge of being influenced by subjectivity. Moreover, we include the age of the user pain report, because the system does not prompt the user to provide it in the self-pain report at each exercise and recent pain reports are more significant than older.

4.3 Action space

The system considers the multimodal/multisensing input in order to decide on the next action. The system takes such actions in order to:

- Engage the subject to complete the whole set of exercises correctly; if the subject does not execute well one exercise the system learns to level down the difficulty of the exercise.
- Keep the exercise difficulty level at the most efficient level (Normal). In case the subject cannot perform well or there are indicators of pain, the system learns to adjust the difficulty type or level of exercise.
- Prevent the subject from incurring an injury and/or high levels of pain; if the system detects a facial expression denoting pain, then it may ask the user to provide their own pain report level and then lower the exercise difficulty.
- Spend the appropriate amount of time on each exercise; the system keeps track of the time spent on each exercise, so if a lot of time is spent in one exercise, then the system prompts the user to move to the next exercise.
- Prevent high pain levels; if subject's pain is at a high level, then it is more likely that the subject will quit.

The system's available actions can modify and adjust the session by changing either the exercise itself or its difficulty level.

4.4 Reward Function

An important part of the problem formulation as a Markov Decision Process is the reward function definition. It is responsible for evaluating the state-action pairs. For this approach, we designed the reward function to be dependent on the state features that describe the exercise ID and level and the duration of each exercise. The reason we did not include the pain features is that they directly influence the possibility of quitting, where the manager receives a high negative reward.

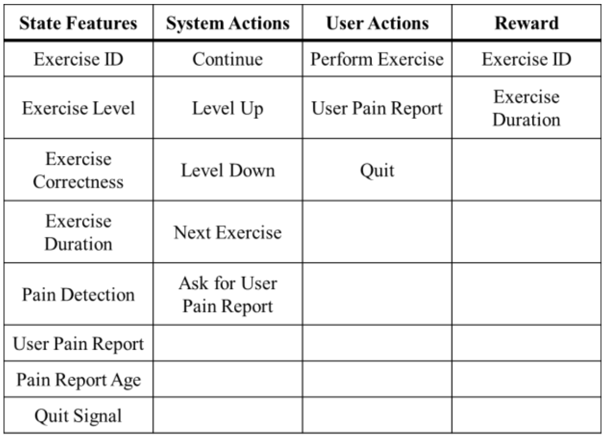
The reward function is formulated as follows:

$$\mathbf{R}(\mathbf{S}) = \begin{cases} 10 & , \; if \; success \\ -100 & , \; if \; quit \\ e^{level} \cdot \mathcal{N}(time; 3, 2) & , \; otherwise \end{cases}$$

Regarding the level of the exercise, we defined the reward as an exponential function. The manager system will learn to take the appropriate actions in order to reach the highest exercise level resulting in a successful session.

On the other hand, the subject must spend an appropriate amount of time on each of the exercises. In order to achieve this, we use a Gaussian distribution with mean $\mu = 3$. We define the appropriate time for each exercise to be three time units, where the Gaussian gives the highest density value. When the subject completes the session, the system receives a positive reward, while it receives a high negative reward in case of quitting the session. In Table 1, we summarize the state and action space, as well as the reward function features.

Table 1. Definition of the Markov Decision Process



State Features	System Actions	User Actions	Reward
Exercise ID	Continue	Perform Exercise	Exercise ID
Exercise Level	Level Up	User Pain Report	Exercise Duration
Exercise Correctness	Level Down	Quit	
Exercise Duration	Next Exercise		
Pain Detection	Ask for User Pain Report		
User Pain Report			
Pain Report Age			
Quit Signal			

5. SIMULATION AND GRAPHICAL USER INTERFACE

A significant weakness of RL approaches is the lack of data needed in order to estimate a transition model or to implement a reliable and accurate user model. In order to make an initial evaluation of our manager system, we have manually defined probability models to express the user responses to the session modifications based on the system actions. Moreover, using the Dyna architecture, as we mentioned, the system also learns a model based on experience.

Our assumption is that using a probabilistic model for the subject's reactions (e.g., pain, performance, quitting) to the system's decisions, expresses the uncertainty due to human factors, at a simulation level.

We define five user models that describe the different state features. Specifically, we define the *User Real Pain* model, the *User Pain Report* model, the *User Visual Pain* model, the *User Exercise Correctness* model and the *User Quit* model.

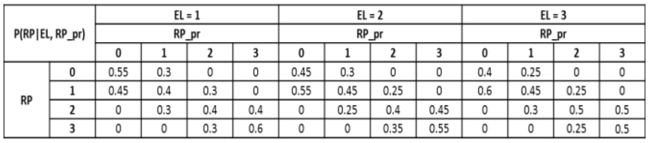
5.1 User Real Pain model

In order to simulate as accurately as possible the pain reaction of the subject while performing the exercises, we define a model for the subject/user real pain. However, the state variables that refer to the subject's pain level are the user pain report and the visual pain detection. These variables cannot be defined in a deterministic manner. A user that suffers from high level of pain may provide the system with different user pain reports due to underestimation or overestimation. Moreover, we need to take into consideration the error possibility of the visual pain detection module. We will discuss about these two state variables and their models in the corresponding subsections.

In particular, the real pain model gives the probability *P(*RP| EL, RP_pr*)*, where RP ∈ [0,1,2,3] is the variable indicating the real pain level, EL ∈ [1,2,3] is the variable indication the exercise difficulty level and RP_pr indicates the real pain level during the previous exercise execution.

We make the assumption that the level of pain depends on the exercise level and the pain level experienced previously. In other words, a more demanding exercise has a higher probability to show or lead to higher than normal levels of pain, if there was a high level pain (i.e. RP_pr) experienced previously, than if the exercise started without any previous high pain Table 2 shows the defined model that takes into account this discussion.

Table 2. User Real Pain model



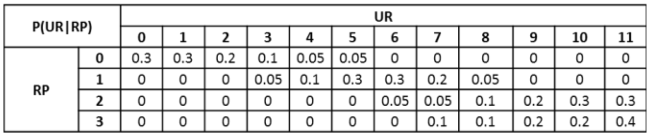
P(RP|EL, RP_pr)		EL = 1				EL = 2				EL = 3			
		RP_pr				RP_pr				RP_pr			
		0	1	2	3	0	1	2	3	0	1	2	3
RP	0	0.55	0.3	0	0	0.45	0.3	0	0	0.4	0.25	0	0
	1	0.45	0.4	0.3	0	0.55	0.45	0.25	0	0.6	0.45	0.25	0
	2	0	0.3	0.4	0.4	0	0.25	0.4	0.45	0	0.3	0.5	0.5
	3	0	0	0.3	0.6	0	0	0.35	0.55	0	0	0.25	0.5

5.2 User Pain Report model

One of the possible actions of the manager is to ask the user for their own pain report. The pain report is translated into a numerical *likert scale* [36]. As mentioned in the real pain model section above, the user pain report depends on the actual pain of the subject. Based on the real pain level, the model gives the probability *P*(UR| RP), where UR ∈ [0,11] is the variable that indicates the user report in the likert scale and RP is the real pain level defined by the real pain model.

The significance of using this model is the consideration of the human factors affecting the computation. The subject can misestimate their actual pain level. The basic reason is that pain cannot be measured and classified into a specific numeric level deterministically, since it is subjective. Moreover, other factors, such as psychological or physiological condition, led us to define the UR variable dependent on the real pain level, as shown in Table 3.

Table 3. User Pain Report model

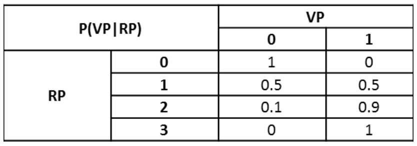


P(UR|RP)		UR											
		0	1	2	3	4	5	6	7	8	9	10	11
RP	0	0.3	0.3	0.2	0.1	0.05	0.05	0	0	0	0	0	0
	1	0	0	0	0.05	0.1	0.3	0.3	0.2	0.05	0	0	0
	2	0	0	0	0	0	0	0.05	0.05	0.1	0.2	0.3	0.3
	3	0	0	0	0	0	0	0	0.1	0.1	0.2	0.2	0.4

5.3 User Visual Pain model

An important state variable of MDP is the feedback of the pain detection module using facial features. In [28], a pain detection system is described, that uses shape and appearance facial features. The accuracy of their system reaches 90.2% using decision trees. Despite this high accuracy, it still has an error possibility. For this reason, we define a *visual pain model* that gives the probability *P*(VP |RP), where VP is a binary variable indicating the presence or absence of pain. Based on the high accuracy that the pain detection system achieves, we assume that if the real pain level is the minimum or maximum, the pain detection system will be accurate. The probabilistic model we define is suitable to handle the uncertainty of the pain detection system for the mid-levels of pain, as shown in Table 4.

Table 4. User Visual Pain model



P(VP|RP)		VP	
		0	1
RP	0	1	0
	1	0.5	0.5
	2	0.1	0.9
	3	0	1

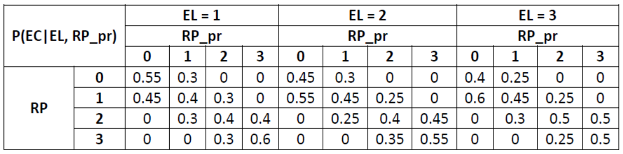
We have to mention that such models are used in order to make an initial evaluation of our proposed approach. Through the collection of real user interaction data, the system can learn a more robust learning model for our system's user. .

5.4 User Exercise Correctness model

A valuable feedback for the session manager is the exercise execution correction by the subject-user. If the subject underperforms or executes an exercise in an improper way, differently than the one prescribed, the system must adjust the difficulty level in order to enable the subject to perform the exercises correctly.

The subject's performance while executing an exercise depends on the exercise difficulty level and the pain level. There are studies that show a high association between bodily pain, pain "catastrophizing" and exercise performance [29]. Taking into consideration the association between exercise performance with pain level and difficulty level, we define the exercise correctness model which gives the probability *P*(EC| EL, RP_pr), where EC ∈ [0,1,2,3] is the variable indicating how well the subject executes the exercise and EL ∈ [1,2,3] indicates the difficulty level of the current exercise, as shown in Table 5.

Table 5. User Exercise Correctness model



P(EC|EL, RP_pr)		EL = 1				EL = 2				EL = 3			
		RP_pr				RP_pr				RP_pr			
		0	1	2	3	0	1	2	3	0	1	2	3
RP	0	0.55	0.3	0	0	0.45	0.3	0	0	0.4	0.25	0	0
	1	0.45	0.4	0.3	0	0.55	0.45	0.25	0	0.6	0.45	0.25	0
	2	0	0.3	0.4	0.4	0	0.25	0.4	0.45	0	0.3	0.5	0.5
	3	0	0	0.3	0.6	0	0	0.35	0.55	0	0	0.25	0.5

Recent work has proposed the use of computer vision in order to identify and track body motion movements and gestures. Most of this research work uses RGB-D data acquired by the Kinect sensor using the RGB-D sensor data to track body motion in order to ensure subject compliance with the prescribed physical therapy routines and activity levels [30] and to evaluate rehabilitation tools using this low-cost sensor [31]. In [32], the authors propose to apply computer vision methods. They use a popular commercial skeleton tracking software solution in a large vocabulary gesture recognition system and an RGB-D gesture dataset for gesture recognition. The exercise correctness can be defined as the similarity between the prescribed exercise trajectories and the trajectories of the motion that the subject executes during the exercising using the Dynamic Space- Time Warping algorithm [33,34].

For our implementation, during real interaction, we will classify the exercise correctness to an integer scale [0-3], based on the similarity of the two executions (subject and therapist motion).

5.5 User Quit model

One possible user action during the interaction with the system is Quit. The subject may quit for several reasons. Some of them could be non-compliance, fatigue, even loss of interest. We propose to overcome these burdens by adjusting dynamically the exercise difficulty as mentioned in previous sections. We also make the assumption that the subject is more likely to quit the session, if they are on high pain levels.

In order to express this dependence between quitting and pain, we define the *user-quit model* that depends on the subject's pain level. This model gives the probability *P*(Q |RP), where Q is a binary variable that indicates if the user wants to quit the exercise. The model is shown in Table 6.

Table 6. User Quit model

P(Q|RP)		Q	
		0	1
RP	0	1	0
	1	1	0
	2	0.95	0.05
	3	0.9	0.1

During real interactions, the system will be able to collect data in order to define a more accurate and personalized user model that will be expressing the probability of quitting the session depending on various multisensing data describing physiological and psychological factors, such as current mood, performance and progress, facial expressions and speech data, which can be used for long-term adherence.

5.6 Adaptive Session Manager GUI

Reinforcement Learning algorithms often require a large amount of real or simulated interactions with the system in order to converge to an optimal policy. For this initial implementation of this approach, we apply the model-based Dyna-Q algorithm that uses the user models to decide which will the next state be and also perform offline simulation steps to update the Q-values.

In order to get an insight of how this works, we designed a Graphical User Interface (GUI) that shows the session number, the current state of the system and the selected action of the manager. It also shows if the simulated session was successful or unsuccessful. We have integrated some simple functions to make the visualization easier and user-friendly, such as increase or decrease the simulation speed, start, stop or reset the simulation and plot a figure with the results. In Fig 4, we show two captions of the manager simulation GUI (a demo can also be viewed in https://www.youtube.com/watch?v=o46tiqwSX08).

The first capture shows Session No. 21, after a small number of interactions of the system. At this point the system does not have much knowledge of the state space and action space, so the selected action for the current state is not the one that will lead to the maximum accumulated reward. More specifically, the system chooses to move to the next exercise very early, in spite of the good performance of the subject and the absence of pain. After a few interactions, at session No. 142, where the system has gained more knowledge, the manager lowers the exercise difficulty, due to the high level of pain indicated by both visual and speech input.

For the real interactions with the subject, the system will use an avatar to show the exact exercise at the proposed difficulty to make the interaction more natural and easier for the subject.

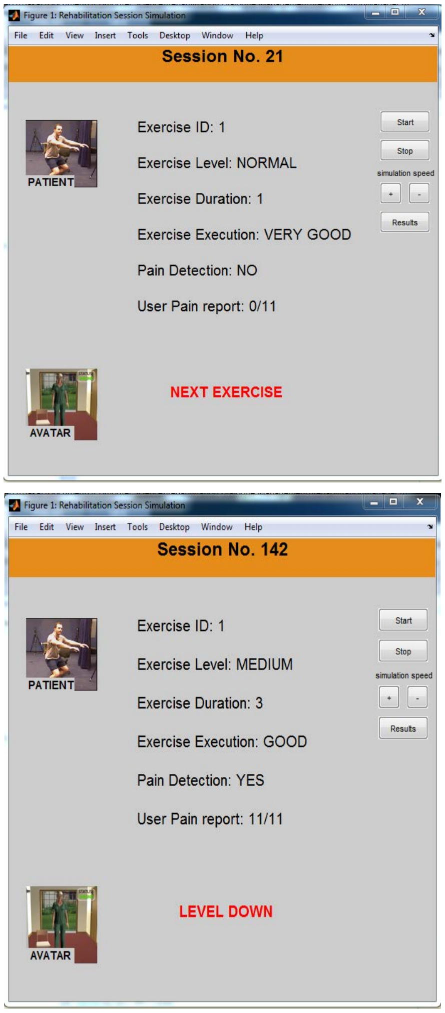


Figure 1: Rehabilitation Session Simulation
File Edit View Insert Tools Desktop Window Help
PATIENT
AVATAR
Start
Stop
simulation speed
+
-
Results

Session No. 21

Exercise ID: 1

Exercise Level: NORMAL

Exercise Duration: 1

Exercise Execution: VERY GOOD

Pain Detection: NO

User Pain report: 0/11

NEXT EXERCISE

Figure 1: Rehabilitation Session Simulation
File Edit View Insert Tools Desktop Window Help
PATIENT
AVATAR
Start
Stop
simulation speed
+
-
Results

Session No. 142

Exercise ID: 1

Exercise Level: MEDIUM

Exercise Duration: 3

Exercise Execution: GOOD

Pain Detection: YES

User Pain report: 11/11

LEVEL DOWN

Figure 4. The figure shows two captions of the GUI for the manager simulation. It shows the current state features and the selected action by the manager on different simulation numbers.

6. EXPERIMENTAL PARAMETERS AND RESULTS

For this initial implementation of the proposed system, we applied the Dyna-Q algorithm for the learning. Each session starts from the same start state, where all state features are zero. At, each iteration, the manager chooses the next action following the ε-greedy policy, where the manager chooses the action that it believes has the best long-term effect with probability 1-ε (exploitation), or chooses a random action with probability ε (exploration). The initial value is ε = 0.9. We follow an adaptive exploration-exploitation strategy, decreasing the exploration parameter ε after each successful session. Following this approach, the manager starts with high exploration probability in order to explore as much as possible its environment. After each success, the manager starts to exploit its knowledge more and more. Considering the huge state space and the uncertainty caused by the models, we decrease the ε value by 0.01%, in order to let the manager acquires the knowledge needed. After each iteration, the algorithm learns a model by 'experience'. It uses this model to perform *N* offline simulation steps, using the observed state-action pairs, to update the Q-values. For this implementation, the algorithm performs *N* = 100 simulation steps. After 30000 simulated interactions with the system, we evaluate the algorithms results. What we are interested in is to minimize the number of quits and maximize the average discounted reward ϱ. The discount factor we use is γ = 0.9 such that

$$\varrho = \sum_{\tau=0}^{\infty} \gamma^{\tau} \cdot R(s_{\tau})$$

In order to plot the results, we plot the number of quits and the average discounted reward for each 1000 sessions. The results are shown in Fig. 5.

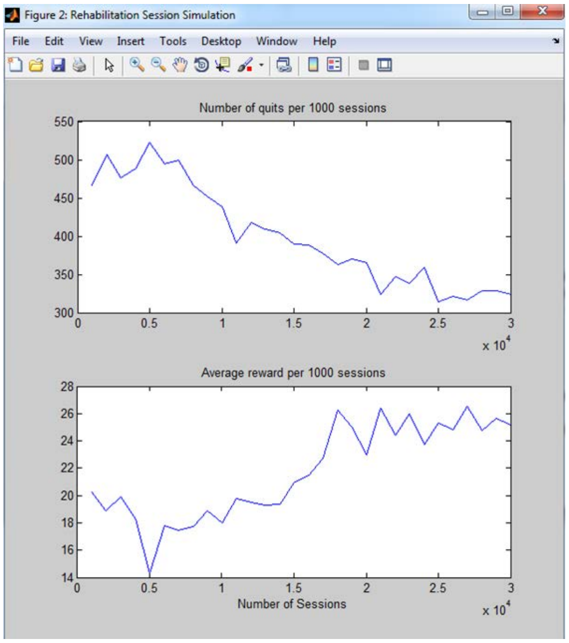


Figure 5. Results showing the number of quits and the average reward per 1000 sessions.

We observe that the number of quits decreases as the algorithm learns the optimal policy. Moreover, the average discounted reward increases as the algorithm learns. The results are promising and are evidence that the manager learns by experience which action is best for each state in order to keep the subject safe and compliant to the rehabilitation session, by adapting the session exercise and difficulty or by asking the subject to self-report.

7. DISCUSSION AND FUTURE WORK

In this paper, we proposed ReADAPT, an adaptive multimodal session manager for rehabilitation physical exercising. The manager adapts the session by taking into consideration multisensing data during the interaction with the subject. Also, it interacts with the subject asking his/her for feedback regarding comfort and pain levels, in order to make the interaction more natural for the user. We formulated the problem as a Markov Decision Process and applied the Dyna-Q reinforcement learning algorithm to learn the optimal policy. For this first implementation, we defined a user model in order to get simulated data and train the algorithm. In order to get an insight of the system learning, we followed a strong assumption that the subject is likely to quit under high pain levels. Future work includes research on ACSM principles for exercise intensity judgments of exercise progression [36]. An important functionality of the proposed system is that it utilizes multisensing data in order to adapt to the specific subject in terms of performance and pain levels.

As a future extension, we will conduct a round of Wizard of Oz experiments. A Wizard of Oz experiment is a research experiment in which subjects interact with a computer system that subjects believe to be autonomous, but which is actually being operated or partially operated by an unseen human being. The Wizard of Oz technique enables unimplemented technology to be evaluated by using a human to simulate the response of a system. Data will be recorded during these interactions and will be then used as training data for the learning algorithms. After the algorithms have been trained, we will conduct a second round of experiments in order to evaluate our system with human users who are not subjects as well as with trained therapists to gain intuition from their perspective. In future work, we will collect data during real interactions to evaluate how encouragement or breaks can assist the subject in terms of physiological and psychological state.

An important contribution of such a system could be the collection of the real interaction data that will be annotated and combined with data from different modalities. Such data can be used for modeling the user pain-based reactions during exercising. Using exercise performance data can also lead to a valuable research resource an annotated multimodal human activity repository. Finally, this work is important in a psychological setting. Audiovisual descriptors [35] can be used to associate exercise performance with psychological conditions in order to evaluate how emotional and mental states can affect compliance to physical performance, especially in the workplace.

8. AKNOWLEGDMENTS

This material is based upon work supported by the National Science Foundation under Grants No. NSF-CNS 1035913, NSF-IIS 1409897, NSF-CNS 1338118.

9. REFERENCES

[1] Legg, L. (2004). Rehabilitation therapy services for stroke patients living at home: systematic review of randomised trials. The Lancet.

[2] Bean, J. F., Vora, A., & Frontera, W. R. (2004). Benefits of exercise for community-dwelling older adults. Archives of physical medicine and rehabilitation, 85, 31-42.

[3] Maclean, N., Pound, P., Wolfe, C., & Rudd, A. (2000). Qualitative analysis of stroke patients' motivation for rehabilitation. *Bmj*, *321*(7268), 1051-1054.

[4] Lequerica, A. H., Donnell, C. S., & Tate, D. G. (2009). Patient engagement in rehabilitation therapy: Physical and occupational therapist impressions. Disability & Rehabilitation, 31(9), 753-760.

[5] Funk, J. A., Gohlke, J., Kraft, A. D., McPherson, C. A., Collins, J. B., & Jean Harry, G. (2011). Voluntary exercise protects hippocampal neurons from trimethyltin injury: possible role of interleukin-6 to modulate tumor necrosis factor receptor-mediated neurotoxicity. *Brain, behavior, and immunity*, *25*(6), 1063-1077.

[6] Garrino, L., Curto, N., Decorte, R., Felisi, N., Matta, E., Gregorino, S. & Carone, R. (2011). Towards personalized care for persons with spinal cord injury: a study on patients'

perceptions. The journal of spinal cord medicine, 34(1), 67-75.

[7] Vesmarovich, S., Walker, T., Hauber, R. P., Temkin, A., & Burns, R. (1999). Use of Telerehabilitation to Manage Pressure Ulcers in Persons with Spinal Cord Injuries: Abstract. Advances in Skin & Wound Care, 12(5), 264-269.

[8] Theodoros, D., Russell, T., & Latifi, R. (2008). Telerehabilitation: current perspectives. Studies in health technology and informatics, 131, 191-210.

[9] Torsney, K. (2003). Advantages and disadvantages of telerehabilitation for persons with neurological disabilities. NeuroRehabilitation, 18(2), 183-185.

[10] Rizzo, A. A., Strickland, D., & Bouchard, S. (2004). The challenge of using virtual reality in telerehabilitation. Telemedicine Journal & E-Health, 10(2), 184-195.

[11] Rizzo, A., & Kim, G. (2005). A SWOT analysis of the field of virtual reality rehabilitation and therapy. Presence, 14(2), 119-146.

[12] Cikajlo, I., Rudolf, M., Goljar, N., Burger, H., & Matjacic, Z. (2012). Telerehabilitation using virtual reality task can improve balance in patients with stroke. Disability and rehabilitation, 34(1), 13-18.

[13] Wijkstra, P. J., Van der Mark, T. W., Kraan, J., Van Altena, R., Koeter, G. H., & Postma, D. S. (1996). Long-term effects of home rehabilitation on physical performance in chronic obstructive pulmonary disease. American journal of respiratory and critical care medicine, 153(4), 1234-1241.

[14] Choi, Y., Gordon, J., Kim, D., & Schweighofer, N. (2009). An adaptive automated robotic task-practice system for rehabilitation of arm functions after stroke. Robotics, IEEE Transactions on, 25(3), 556-568.

[15] Baran, M., Lehrer, N., Siwiak, D., Chen, Y., Duff, M., Ingalls, T., & Rikakis, T. (2011, August). Design of a home-based adaptive mixed reality rehabilitation system for stroke survivorsAnnual International Conference of the IEEE (pp. 7602-7605). IEEE.

[16] Duff, M., Chen, Y., Attygalle, S., Herman, J., Sundaram, H., Qian, G.,& Rikakis, T. (2010). An adaptive mixed reality training system for stroke rehabilitation. Neural Systems and Rehabilitation Engineering, IEEE Transactions on, 18(5), 531-541.

[17] Kan, P., Huq, R., Hoey, J., Goetschalckx, R., & Mihailidis, A. (2011). The development of an adaptive upper-limb stroke rehabilitation robotic system. Journal of neuroengineering and rehabilitation, 8(1), 1-18.

[18] Saran, A., Kitani, K. M., & Rikakis, T. (2014, September). Automating Stroke Rehabilitation for Home-Based Therapy. In 2014 AAAI Fall Symposium Series.

[19] Papangelis, A., Galatas, G., Tsiakas, K., Lioulemes, A., Zikos, D., & Makedon, F. (2014). A Dialogue System for Ensuring Safe Rehabilitation. In Universal Access in Human-Computer Interaction. Aging and Assistive Environments (pp. 349-358). Springer International Publishing.

[20] Galatas, G., Potamianos, G., & Makedon, F. (2012, August). Audio-visual speech recognition incorporating facial depth information captured by the Kinect. In Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European (pp. 2714-2717). IEEE.

[21] Sutton, R. S., & Barto, A. G. (1998). Introduction to reinforcement learning. MIT Press.

[22] Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. arXiv preprint cs/9605103.

[23] Atkeson, C. G., & Santamaria, J. C. (1997). A comparison of direct and model-based reinforcement learning. In In International Conference on Robotics and Automation.

[24] Sutton, R. S. (1991). Dyna, an integrated architecture for learning, planning, and reacting. ACM SIGART Bulletin, 2(4), 160-163.

[25] Brewer, B. W., Cornelius, A. E., Van Raalte, J. L., Tennen, H., & Armeli, S. (2013). Predictors of adherence to home rehabilitation exercises following anterior cruciate ligament reconstruction. Rehabilitation psychology, 58(1), 64.

[26] Melinda Hamill, Vicky Young, Jennifer Boger and Alex Mihailidis. “Development of an automated speech recognition interface for personal emergency response systems”. Journal of NeuroEngineering and Rehabilitation. 2009. 6-26.

[27] Collignon O, Girard S, Gosselin F, Roy S, Saint-Amour D, Lassonde M, Lepore F. “Audio-visual integration of emotion expression”. Brain Research, Vol. 1242, 2008. 126-135.

[28] Khan, R. A., Meyer, A., Konik, H., & Bouakaz, S. (2013, July). Pain detection through shape and appearance features. In Multimedia and Expo (ICME), 2013 IEEE International Conference on (pp. 1-6). IEEE.

[29] Nijs, J., Van de Putte, K., Louckx, F., Truijen, S., & De Meirleir, K. (2008). Exercise performance and chronic pain in chronic fatigue syndrome: the role of pain catastrophizing. Pain Medicine, 9(8), 1164-1172.

[30] Metsis, V., Jangyodsuk, P., Athitsos, V., Iversen, M., & Makedon, F. (2013, January). Computer aided rehabilitation for patients with rheumatoid arthritis. In Computing, Networking and Communications (ICNC), 2013 International Conference on (pp. 97-102). IEEE.

[31] Lange, B., Chang, C. Y., Suma, E., Newman, B., Rizzo, A. S., & Bolas, M. (2011, August). Development and evaluation of low cost game-based balance rehabilitation tool using the Microsoft Kinect sensor. In Engineering in Medicine and Biology Society, EMBC, 2011

[32] Conly, C., Doliotis, P., Jangyodsuk, P., Alonzo, R., & Athitsos, V. (2013, May). Toward a 3D body part detection video dataset and hand tracking benchmark. In *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments* (p. 2). ACM.

[33] Stefan, A., Athitsos, V., Alon, J., & Sclaroff, S. (2008, July). Translation and scale-invariant gesture recognition in complex scenes. In *Proceedings of the 1st international conference on PErvasive Technologies Related to Assistive Environments* (p. 7). ACM.

[34] Alon, V. Athitsos, Q. Yuan, and S. Sclaroff. A unified framework for gesture recognition and spatiotemporal gesture segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 31(9):1685–1699, 2009.

[35] Scherer, S., Stratou, G., Mahmoud, M., Boberg, J., Gratch, J., Rizzo, A., & Morency, L. P. Automatic behavior descriptors for psychological disorder analysis. In Automatic Face and Gesture Recognition (FG), 2013

[36] American College of Sports Medicine. ACSM's guidelines for exercise testing and prescription. Lippincott Williams & Wilkins, 2013.

[37] Hartrick, C. T., Kovan, J. P., & Shapiro, S. (2003). The numeric rating scale for clinical pain measurement: a ratio measure?. Pain Practice, 3(4), 310-316