Automated Upper Extremity Rehabilitation for Stroke Patients Using a Partially Observable Markov Decision Process

Patricia Kan

Inst. of Biomaterials and Biomedical Engineering, University of Toronto
Toronto, Ontario, M5S 3G9
patricia.kan@utoronto.ca

Jesse Hoey

School of Computing
University of Dundee
Dundee, Scotland, DD14HN
jessehoey@computing.dundee.ac.uk

Alex Mihailidis

Dept. of Occup. Science and Occup. Therapy, University of Toronto
Toronto, Ontario, M5G 1V7
alex.mihailidis@utoronto.ca

Abstract

This paper presents a real-time system that guides stroke patients during upper extremity rehabilitation. The system automatically modifies exercise parameters to account for the specific needs and abilities of different individuals. We describe a partially observable Markov decision process (POMDP) model of a rehabilitation exercise that can capture this form of customization. The system will be evaluated in user trials during summer 2008 in Toronto, Canada.

Introduction

Stroke is the leading cause of adult disability in Canada. Every year, 60% of the 50,000 stroke sufferers are left with permanent or long-lasting disability (CNS 2007). A growing body of research shows that stroke rehabilitation can substantially reduce the limitations and disabilities that arise from stroke, and improve function, allowing stroke survivors to regain their quality of life and independence (TRI 2008). However, this long and physically demanding process is both slow and tedious, usually involving one-on-one therapist-patient repetitive therapy. A primary motivation for developing rehabilitation robotic devices is to automate interventions that are repetitive and labor-intensive. This can provide stroke patients with intensive movement training without the expense of a continuously present therapist, thus, reducing health care costs and physical strain on therapists. These devices can also provide accurate measures on patient performance and function (Hidler et al. 2005).

Stroke patients with an affected upper-limb have great difficulties performing many activities of daily living (ADLs), such as reaching to grasp objects. Although there are many robotic systems designed to assist and improve upper-limb stroke rehabilitation (Brewer, McDowell, and Worthen-Chaudhari 2007), none of them are able to autonomously learn and adapt to different users over time. This feature is especially important if the intention is to minimize therapist intervention in the clinic or to eventually use the device in the home setting. This paper presents a real-time system designed to facilitate upper-limb reaching rehabilitation for moderate level stroke survivors. It has the ability to operate autonomously (without any explicit feedback) and account for the specific needs and abilities of each individual, which will change over time. The system uses a partially observable Markov decision process (POMDP), a versatile decision-theoretic modeling technique, as the primary decision maker for our upper-limb stroke rehabilitation robotic device.

There have been several robotic devices that use AI techniques to provide assistance for upper extremity rehabilitation (Ju et al. 2005; Erol et al. 2006). However, these intelligent devices do not account for psychological factors (e.g. fatigue) that may affect rehabilitation progress. POMDPs have the ability to make decisions about unobservable parameters that cannot be measured directly, such as fatigue, and have been applied in the assistive technology area such as assisting elderly individuals with their daily activities (Pineau et al. 2003) and assisting older adults with dementia during the handwashing task (Hoey et al. 2007).

System Overview

Our system for guiding stroke patients during upper-limb stroke rehabilitation consists of three main components: the exercise (Figure 1), the robotic system (Figure 2a), and POMDP (Figure 2b). As the user performs the reaching exercise on the robot, data from the device is used as input to the POMDP, where it decides on an action for the system to take.

Exercise. Discussions with a team of experienced occupational and physical therapists at Toronto Rehabilitation Institute (TRI) have identified that early stage exercises for upper-limb stroke patients, such as the reaching motion, is an area of rehabilitation that is in need of more efficient tools. Thus, a targeted, forward reaching task was chosen as the exercise for this project. Figure 1 provides a basic overview of the reaching exercise. The exercise begins with a slight forward flexion of the shoulder, and extension of the elbow and wrist (Figure 1a).

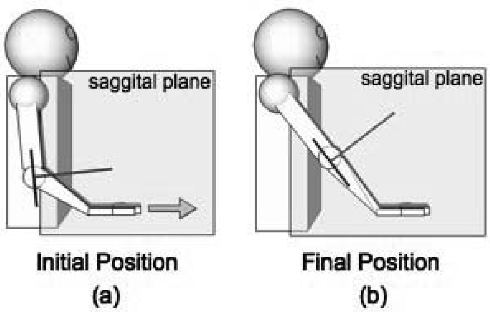

Figure 1: The reaching exercise.

Weight is translated through the heel of the hand as it is pushed forward in the direction indicated by the arrow, until it reaches the final position (Figure 1b). The reaching exercise occurs in the saggital plane (aligned with the shoulder), and the return path brings the arm back to the initial position. It is important to note that a proper reaching exercise is performed with control (e.g. no deviation from the straight path) and without compensation (e.g. trunk rotation, shoulder abduction/internal rotation). Although the reaching motion is fundamental to many ADLs, incorporating resistance into the exercise will help to improve coordination and strengthen muscle control, which will provide support and anchoring for other body movements (e.g. pushing down on a chair to stand up or using stair handrails for support).

Robotic System. The robotic system (Figure 2a) includes a haptic-robotic device, postural trunk sensors, and a virtual environment. The novel robotic device, as detailed in (Lam 2007), has been built to assist in the forward reaching exercise. Figure 3 shows an actual diagram of the rehabilitation device. The non-restraining platform has two active and two passive degrees of freedom, and allows the reaching exercise to be performed in three-dimensional (3D) space. However, for the purpose of our research, the exercise will only be performed in 2D space (in the horizontal plane parallel to the floor). Encoders in the end-effector provide data to indicate hand position and shoulder abduction/internal rotation (i.e. compensation) during the exercise. The robotic device also incorporates haptic technology, which provides feedback through sense of touch. Haptic refers to the modality of touch and the sensation of shape and texture of virtual objects (McLaughlin, Hespanha, and Sukhatme 2001). In our project, the haptic device provides resistance and boundary guidance for the user during the exercise. The unobtrusive trunk sensors (Figure 4) provide data to indicate trunk rotation compensation. The trunk sensors are comprised of three photoresistors taped to the back of a chair, each in one of three locations: the lower back, and lower left and lower right scapula. The detection of light during the exercise indicates trunk rotation, as it means a gap is present between the chair and user. Lastly, the virtual environment provides the user with visual feedback on target location and hand position during the exercise. Figure 5 shows a close up diagram of the virtual environment. The reaching exercise is disguised in the form of a 2D bull's eye game. The goal of the game is for the user to move the robot's end-effector, which

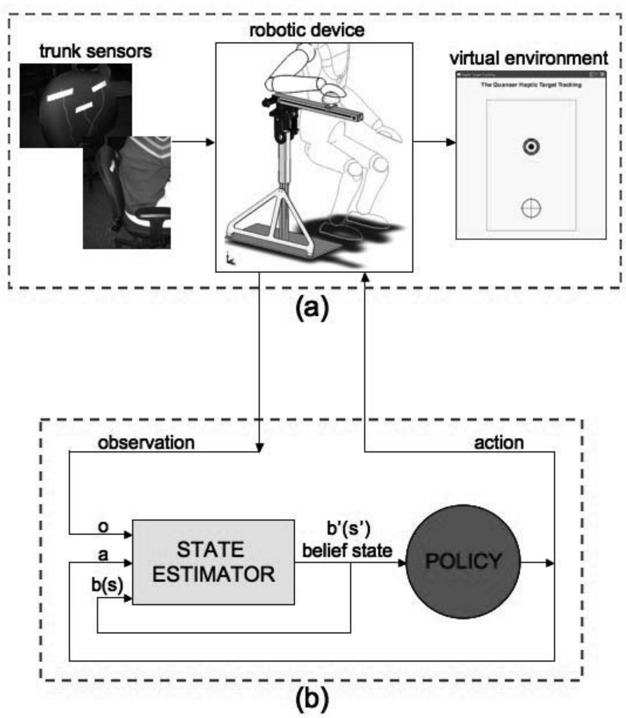

Figure 2: The robotic system (a) and POMDP (b).

Figure 3: Robotic rehabilitation device.

Figure 4: Trunk photoresistor sensors.

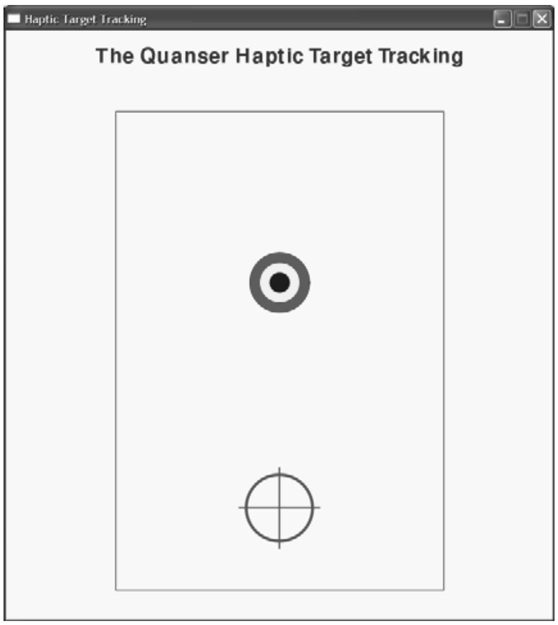

Figure 5: Virtual environment.

corresponds to the cross-tracker in the virtual environment, to the bull's eye target. The rectangular box is the virtual (haptic) boundary, which keeps the cross-tracker within those walls during the exercise.

POMDP. The POMDP (Figure 2b) is the decision-maker of the system. Observation data (e.g. the time it takes the user to reach the target) from the robotic device is passed to a state estimator that estimates the progress of the user as a belief state. A policy then maps the belief state to an action for the system to execute, which is to set a new target position and resistance level or to stop the exercise. The goal of the POMDP agent is to help patients regain their maximum reaching distance at the most difficult level of resistance, while performing the exercises with control and proper posture.

POMDP Model

We now describe the specific POMDP model for the stroke reaching rehabilitation.

A discrete-time POMDP consists of the following components: a finite set of states S of the world; a finite set of actions A; a finite set of observations O; the transition function $T : S \times A \rightarrow \prod(S)$, with $P(s'|s,a)$ denoting the probability of transition from state s to s' by performing action a; the observation function $Z : S \times A \rightarrow \prod(O)$, with $P(o|a,s')$ denoting the probability of observing o by performing action a and landing in state s', and the reward function $R : S \times A \rightarrow \mathbb{R}$, with $R(s,a)$ denoting the expected reward or cost (i.e. negative reward) incurred after performing action a in state s.

The POMDP is used to monitor beliefs about the system state in real-time, and to find a policy that maximizes the expected discounted sum of rewards attained by the system over an infinite horizon. Since knowledge of the system state is never certain, the policy must map belief states (i.e. probability distribution over S) into actions. For an overview of POMDPs, refer to (Astrom 1965; Lovejoy 1991; Kaelbling, Littman, and Cassandra 1998).

State Variables and Actions

The POMDP model of the stroke rehabilitation domain relies on two main variables: *fat* = {*yes*,*no*} describes the user's level of fatigue and *n*(*r*) = {*none*,*d1*,*d2*,*goal*} describes the range (or ability) of the user at a particular resistance level, $r \in$ {*none*,*min*,*max*}. The range is defined as the furthest target distance, $d \in$ {*d1*,*d2*,*goal*}, the user is able to reach at a particular resistance. Thus, if *r*=*min* and the furthest target the user can reach is *d*=*d2*, then the user's range is *n*(*min*)=*d2*.

There are 10 possible actions the system can take. These are comprised of 9 actions of which each is a different combination of resistance level (3 values) and target distance (3 values); and *stop*, which will terminate the exercise when the user is fatigued.

The auxiliary state variables are the user's time to reach the target, *ttt* = {*none*,*slow*,*norm*}, the amount of control they have by staying on the straight path to the target, *ctrl* = {*none*,*min*,*max*}, and if they show compensatory actions such as elbow deviation and trunk rotation, *comp* = {*yes*,*no*}.

The dynamics of a user's rehabilitation are dependent on the concept of *stretch*={*+6*,*+5*,*+4*,*+3*,*+2*,*+1*,*0*,*-1*,*-2*}. The *stretch* is the amount the system is asking the user to go beyond their current range. For example, if the user's range is *n*(*min*)=*d1*, then setting the target at *d*=*d2* at resistance *r*=*min* is a *stretch* of 1.0, while setting the target at *d*=*d1* at resistance *r*=*max* is a *stretch* of 3.0. Note that *stretch* is a direct function of both target distance and resistance level: it is a joint measure of how much a particular distance and resistance are going to push a user beyond their range.

Observations

Currently, the system's observation functions are deterministic, where the variables *ttt*, *ctrl*, and *comp* are actually the observation variables.

Dynamics

Figure 6 shows the current POMDP model as a dynamic Bayesian network (DBN) for all actions except *stop*.

Instead of explicitly using conditional probability tables (CPTs) to describe the transition probability for each variable, all variables of interest can be modeled as simple parametric functions of *stretch* and *fat*. For example, if the user is not fatigued and the system sets a target with a stretch of 0 (so at the user's range), then the user might have a 90% chance of reaching the target at normal time (*ttt*=*norm*). However, if the stretch is set to 1, then this chance might decrease to 50%. Even if the stretch is 0, but the user is fatigued, the chance of reaching the target at *ttt*=*norm* will also decrease. This idea can be applied to the other variables modeling the user's control and compensation, and even their range and fatigue levels. Certainly, a larger stretch will increase the probability of the user becoming fatigued.

We use the sigmoid function as the common parametric function, which relates stretch and fatigue levels to user performance. We call this function the *pace* function, $\phi(s,f)$, which is a function of stretch, s, and fatigue level, f:

$$\phi(s,f) = \frac{1}{1+e^{-(s-m-m(f))/\sigma_s}}$$

where m is the mean stretch (the value of stretch for which the function ϕ is 0.5 if the user is not fatigued), $m(f)$ is a shift dependent on the user's fatigue level (e.g. 0 if the user is not fatigued), and σ_s is the slope of the pace function.

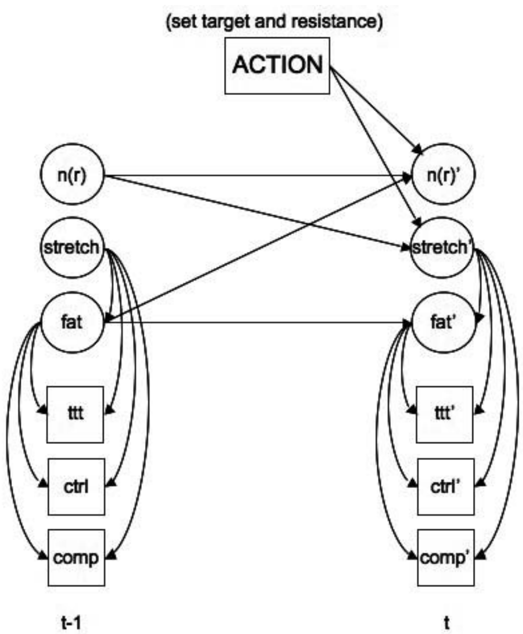

Figure 6: Current POMDP model as a DBN. The shorthand $n(r)$ is used to denote the user's ranges for all resistances. *ttt*, *ctrl*, and *comp* are the deterministic observation variables.

For each pace function, there are three parameters that need to be specified: m, σ_s, and $m(f)$ (where the latter is technically a function, but since the fatigue variable is a binary value in our model, it is a single real-valued parameter). However, it is simpler to specify the pace function in terms of upper and lower *pace limits*: the values of *stretch* where a user's performance will vary by a certain probability when the user is not fatigued ($m(f)$=0). For example, the *upper pace limit* for a user to compensate (*comp*=*yes*) when not fatigued is the *stretch* at which the user will compensate with a probability of ϕ^+. Similarly, the *lower pace limit* for *comp*=*yes* is the *stretch* at which the user will compensate with a probability of ϕ^- (so succeed in reaching the target with *comp*=*no* with a probability of 1-ϕ^-). Denoting the upper and lower pace limits by s^+ and s^-, respectively, we have the following two equations:

$$\phi^+ = \frac{1}{1+e^{-(s^+-m)/\sigma_s}},$$

$$\phi^- = \frac{1}{1+e^{-(s^--m)/\sigma_s}}$$

which can be solved for m and σ_s:

$$m = \frac{s^+\beta^- - s^-\beta^+}{(\beta^- - \beta^+)}$$

$$\sigma_s = \frac{s^+ - s^-}{(\beta^+ - \beta^-)}$$

where $\beta^+ = \ln\left(\frac{\phi^+}{1-\phi^+}\right)$ and $\beta^- = \ln\left(\frac{\phi^-}{1-\phi^-}\right)$.

Setting the pace limits for the variables *ttt*, *ctrl*, and *comp* is simple and intuitive. However, setting those for the user's fatigue level is more challenging since it is difficult to quantify how much more fatigued a user gets in a single trial. It is more intuitive to specify how many trials it takes for a user to become fatigued with a certain probability at some level of *stretch*. We can use this concept to deduce m and σ_s for *fat* (as above) in the following paragraph.

Two time intervals, T_1 and T_2, represent the number of trials it takes for a user to become fatigued (with probability q) at stretch s_1 and s_2, respectively. When specifying these numbers, we assume that all other factors remain the same (i.e. that the user performs the trials in normal time, with control, and no compensation). Although this assumption will not hold in practice, we use it to determine the time intervals and deduce the appropriate parameters for the model. Now, q is the probability a user is fatigued after some number of T time steps, which can be written as:

q = 1 – prob. user not fatigued after T steps

so that:

$$q = 1-(1-p)^T(1-p_0)$$

where p is the (unknown) probability that a user becomes fatigued in a single time step and p_0 is the probability the user is fatigued at the start of the trial (time 0). Solving for p yields:

$$p = 1 - e^{\frac{1}{T}\ln\left(\frac{1-q}{1-p_0}\right)}.$$

Since p is given by the pace function for fatigue, we can write:

$$\frac{1}{1+e^{-(s_i - m)/\sigma_s}} = 1 - e^{\frac{1}{T_i}\ln\left(\frac{1-q}{1-p_0}\right)}$$

where $i \in 1,2$. This equation can be solved for m and σ_s:

$$m = s_1 + \sigma_s \ln(\xi_1)$$

$$\sigma_s = \frac{s_1 - s_2}{\ln(\xi_2 / \xi_1)}$$

where $\xi_i = \dfrac{e^{n/T_i}}{1-e^{n/T_i}}$ and $n = \ln\left(\dfrac{1-q}{1-p_0}\right)$.

The fatigue effect, $m(f)$, is the last parameter to specify, and is a negative number that shifts the pace function downwards. The amount of shift indicates the amount the pace limits will be shifted down when the user is fatigued. Figure 7 shows an example pace function for *comp*=*yes*. Notice that both pace limits decrease when the user is fatigued (at the same probability). In other words, the user is more likely to compensate when fatigued.

For the variables with three values, such as *ttt* and *ctrl*, two pace function need to be specified, one for the lowest value and one for the highest. The middle value gets what is left, so to speak. Figure 8 shows an example pace function for *ttt*.

The ranges in the current model are modeled separately, although they could also use the concept of pace functions. They are modeled such that setting target distances at or just above the user's current range will cause their range to slowly increase. They also have constraints to ensure that ranges at higher resistances are always less than or equal to those at lower resistances.

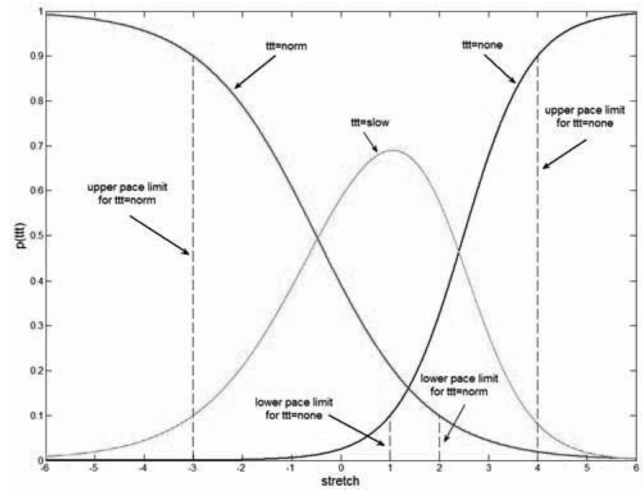

Figure 8: Example pace function for *ttt*, with $\phi^+ = 0.9$, $\phi^- = 0.1$, and $m(f{=}no) = 0.0$. Shown are the upper ($s^+ = -3$) and lower ($s^- = +2$) pace limits for *ttt*=*norm*, and the upper ($s^+ = +4$) and lower ($s^- = +1$) pace limits for *ttt*=*none*. The pace function for *ttt*=*slow* gets what is left of the probability mass.

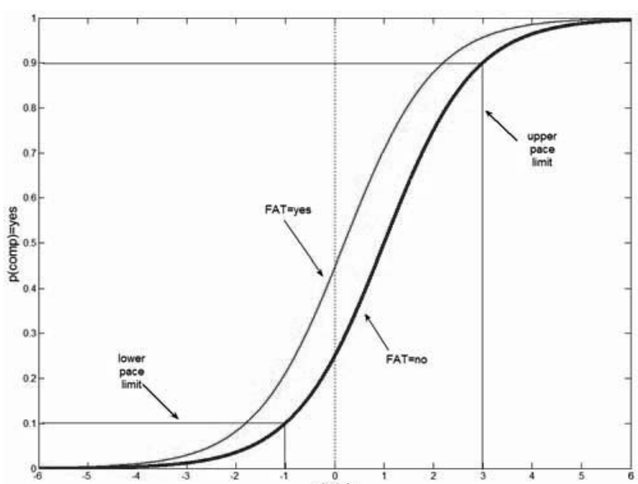

Figure 7: Example pace function for *comp*=*yes*, with $\phi^+ = 0.9$, $\phi^- = 0.1$, $s^+ = +3$, $s^- = -1$, $m(f{=}yes) = 0.8$, and $m(f{=}no) = 0.0$. Shown are the upper and lower pace limits, and the pace function for each condition of *fat*.

Rewards

The reward function is constructed to motivate the system to guide the user to exercise at maximum target distance and resistance level, with maximum control and no compensation. As such, the system gets a large reward when the user can reach the furthest target at maximum resistance. Smaller rewards are given when targets are set at or above the user's current range. However, no reward is given if the user is fatigued, cannot reach the target, has no control, or compensates during the exercise. In addition, no reward is given for negative stretches.

Computation and Simulation

Unique combinations of instantiations of the state variables represent all the different possible states of the rehabilitation exercise that the system can observe. For this model, there are 20,736 possible states.

There are several algorithmic methods for finding optimal POMDP policies as discussed in (Lovejoy 1991). However, the size of our model makes it impossible to solve optimally, thus, approximations must be used. We used a point-based approximation technique based on the Perseus algorithm (Spaan and Vlassis 2005) that exploits the structure of our system dynamics and rewards, which we represented as algebraic decision diagrams (ADDs) (Poupart 2005). We sampled a set of 3,000 belief points that was generated from 20 different initial belief states:

one for each range possibility. The POMDP was solved using 150 linear value functions and 150 iterations in approximately 12.5 hours.

Simulations performed on this model are yielding very encouraging results. During simulation, the policy slowly increases the target distance and resistance level if the user successfully reaches the target in normal time, maximum control, and with no compensation. However, once the user starts to lose control, compensates, or can no longer reach the target, the policy increases its belief that the user is fatigued and decides to stop the exercise to allow the user to take a break.

Figures 9-12 show the changes in the belief states of *n*(*r*), *stretch*, and *fat* during the first simulation example. Given the initial belief state (Figure 9), the POMDP sets the target and resistance at the lowest level (*d*=*d1*, *r*=*none*). At the end of the first trial, the user has successfully reached the target in normal time (*ttt*=*norm*), with maximum control (*ctrl*=*max*), and without compensation (*comp*=*no*). (Note that a *trial* is defined as the reaching exercise from the initial position (Figure 1a) to the final position (Figure 1b), then back to the initial position). Notice that in the updated belief state (Figure 10) *stretch* is about 60% likely to be +1. Thus, the system decides to set the target one level above the range (*d*=*d2*, *r*=*none*). Here, the user reaches the target in slow time (*ttt*=*slow*), with minimum control (*ctrl*=*min*), and again without compensation (*comp*=*no*). Figure 11 displays the updated belief state at the end of the second trial. The POMDP decides to set the same target distance and resistance level

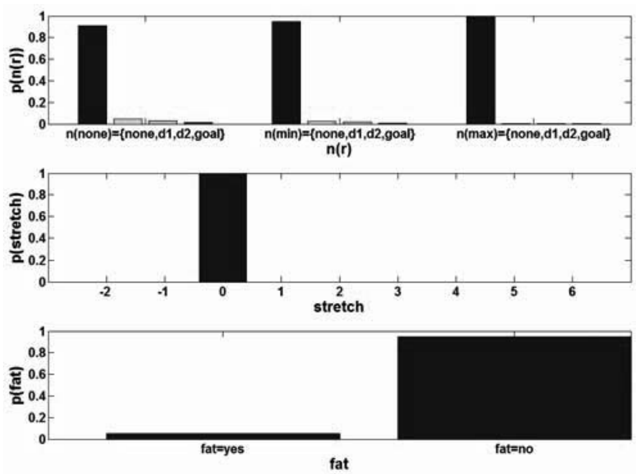

Figure 9: Initial belief state of *n*(*r*), *stretch*, and *fat*. POMDP sets target at *d*=*d1* and resistance at *r*=*none*. User reaches target with *ttt*=*norm*, *ctrl*=*max*, and *comp*=*no*.

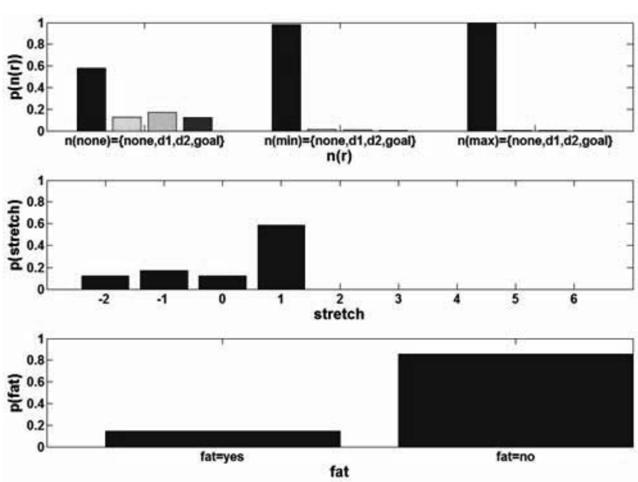

Figure 10: Updated belief state of *n*(*r*), *stretch*, and *fat* after the first trial. POMDP sets target at *d*=*d2* and resistance at *r*=*none*. User reaches target with *ttt*=*slow*, *ctrl*=*min*, and *comp*=*no*.

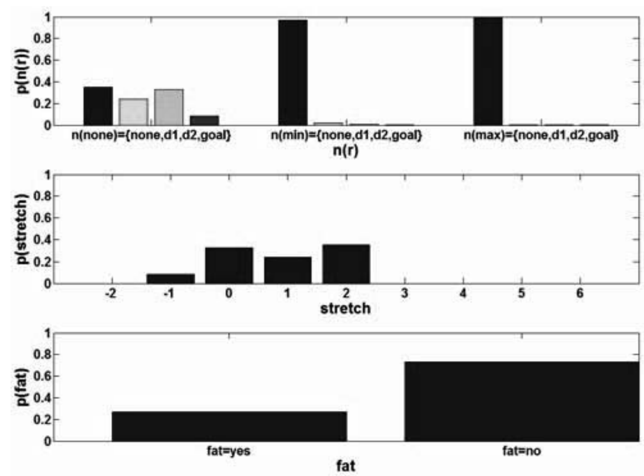

Figure 11: Updated belief state of *n*(*r*), *stretch*, and *fat* after the second trial. POMDP sets target at *d*=*d2* and resistance at *r*=*none*. User reaches target with *ttt*=*slow*, *ctrl*=*none*, and *comp*=*yes*.

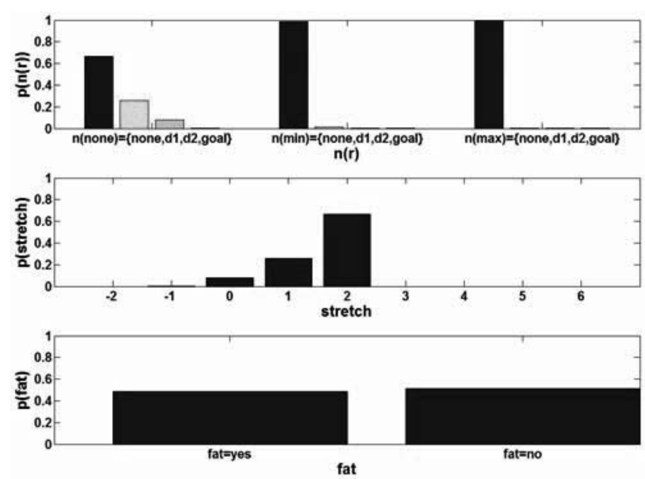

Figure 12: Updated belief state of *n*(*r*), *stretch*, and *fat* after the third trial. POMDP stops the exercise.

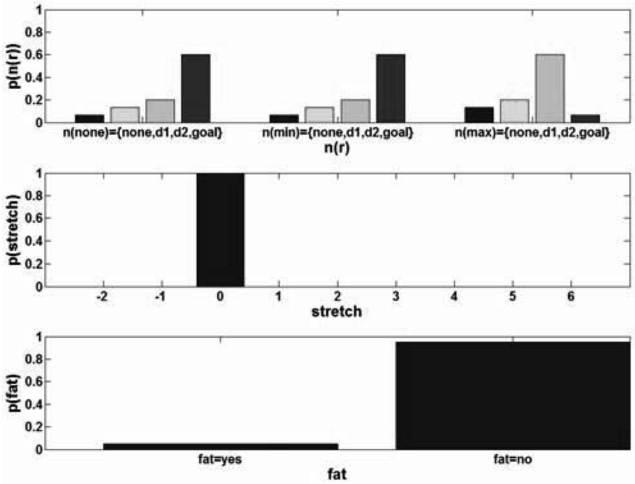

Figure 13: Initial belief state of *n*(*r*), *stretch*, and *fat*. POMDP sets target at *d*=*d2* and resistance at *r*=*max*. User reaches target with *ttt*=*norm*, *ctrl*=*max*, and *comp*=*no*.

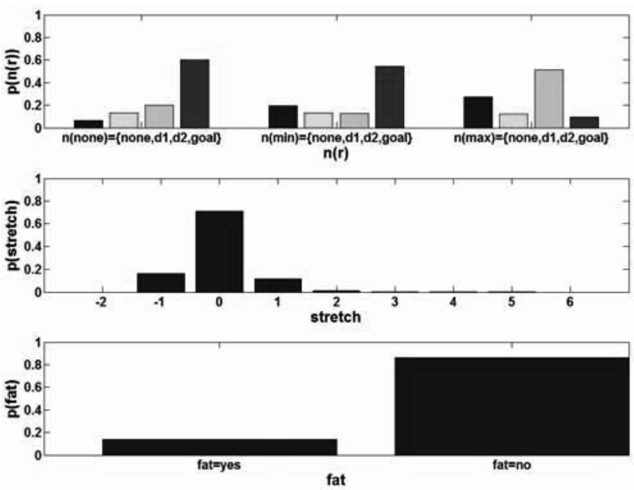

Figure 14: Updated belief state of *n*(*r*), *stretch*, and *fat* after the first trial. POMDP sets target at *d*=*goal* and resistance at *r*=*max*. User fails to reach target with *ttt*=*none*, *ctrl*=*min*, and *comp*=*yes*.

as the previous trial (*d*=*d2*, *r*=*none*). This time, however, the user reaches the target in slow time (*ttt*=*slow*), but with no control (*ctrl*=*none*) and with compensation (*comp*=*yes*). The final belief state (Figure 12) indicates that there is a 50% chance the user is fatigued, and therefore, the POMDP decides to stop the exercise for the user to take a break.

Figures 13-16 show the belief state changes of the second simulation example. The initial belief state is shown in Figure 13. Based on the initial belief state, the POMDP sets the target at *d*=*d2* and resistance at *r*=*max*. Here, the user can successfully reach the target in normal time (*ttt*=*norm*), with maximum control (*ctrl*=*max*), and without compensation (*comp*=*no*). In the updated belief state (Figure 14), notice the shift in the *n*(*max*) range towards the *goal* value and the shift in *stretch* towards +1. Thus, the system decides to increase the target distance by one (*d*=*goal*, *r*=*max*). However, the user fails to reach the target (*ttt*=*none*) with minimum control (*ctrl*=*min*) and with compensation (*comp*=*yes*). Figure 15 displays the updated belief state at the end of the second trial. The POMDP decides to set the same target and resistance as the previous trial (*d*=*goal*, *r*=*max*). Again, the user fails to reach the target (*ttt*=*none*) while compensating (*comp*=*yes*), but this time, has no control (*ctrl*=*none*). The final belief state (Figure 16) indicates that there is a 70%

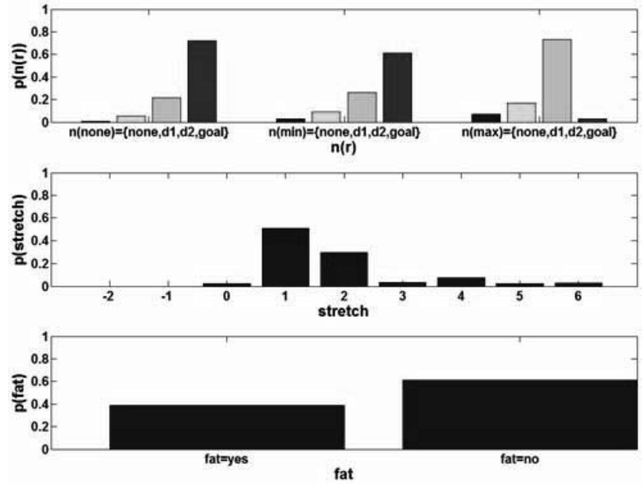

Figure 15: Updated belief state of *n*(*r*), *stretch*, and *fat* after the second trial. POMDP sets target at *d*=*goal* and resistance at *r*=*max*. User fails to reach target with *ttt*=*none*, *ctrl*=*none*, and *comp*=*yes*.

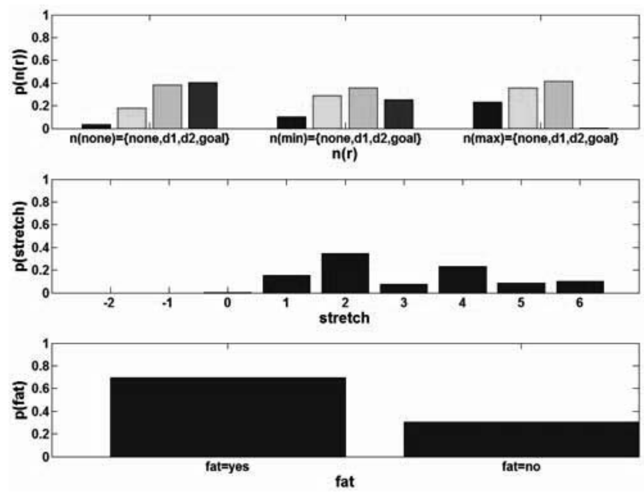

Figure 16: Updated belief state of *n*(*r*), *stretch*, and *fat* after the third trial. POMDP stops the exercise.

chance the user is fatigued and the system stops the exercise.

Conclusion and Future Work

We have presented a system that uses a partially observable Markov decision process (POMDP) to guide stroke patients through a reaching rehabilitation exercise. User trials for this system are scheduled for summer 2008 in Toronto, Canada, and are expected to be completed in the fall.

Acknowledgements

The authors gratefully acknowledge the support of CITO-Precarn and Quanser Inc., and also thank Debbie Hébert for her invaluable feedback.

References

Astrom, K. J. 1965. Optimal Control of Markov Processes with Incomplete State Information. *Journal of Mathematical Analysis and Applications*. 10: 174-205.

Brewer, B. R; McDowell, S. K.; and Worthen-Chaudhari, L. C. 2007. Poststroke Upper Extremity Rehabilitation: A Review of Robotic Systems and Clinical Results. *Topics in Stroke Rehabilitation*. 14(6): 22-44.

Canadian Stroke Network. 2007. Stroke 101. http://www.canadianstrokenetwork.ca/eng/about/stroke101.php.

Erol, D.; Mallapragada, V.; Sarkar, N.; Uswatte, G.; and Taub, E. 2006. Autonomously Adapting Robotic Assistance for Rehabilitation Therapy. In *Proceedings of the First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, 567-572. Pisa, Italy: IEEE.

Hidler, J.; Nichols, D.; Pelliccio, M.; and Brady, K. 2005. Advances in the Understanding and Treatment of Stroke Impairment using Robotic Devices. *Topics in Stroke Rehabilitation*. 12(2): 22-35.

Hoey, J.; von Bertoldi, A.; Poupart, P.; and Mihailidis, A. 2007. Assisting Persons with Dementia during Handwashing using a Partially Observable Markov Decision Process. In *Proceedings of the Fifth International Conference on Computer Vision Systems*. Bielefeld University, Germany: Applied Computer Science Group.

Ju, M. S.; Lin, C. C. K.; Lin, D. H.; Hwang, I. S.; and Chen, S. M. 2005. A Rehabilitation Robot with Force-Position Hybrid Fuzzy Controller: Hybrid Fuzzy Control of Rehabilitation Robot. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*. 13(3): 349-358.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and Acting in Partially Observable Stochastic Domains. *Artificial Intelligence*. 101: 99-134.

Lam, P. 2007. Development of a Haptic-Robotic Platform for Moderate Level Upper-Limb Stroke Rehabilitation. M.A.Sc. thesis, Inst. of Biomaterials and Biomedical Engineering, Univ. of Toronto, Toronto, Canada.

Lovejoy, W. S. 1991. A Survey of Algorithmic Methods for Partially Observed Markov Decision Processes. *Annals of Operations Research*. 28: 47-66.

McLaughlin, M. L.; Hespanha, J. P.; and Sukhatme, G. S. 2001. *Touch in Virtual Environments: Haptics and the Design of Interactive Systems*. Upper Saddle River, NJ: Prentice Hall.

Pineau, J.; Montemerlo, M.; Pollack, M.; Roy, N.; and Thrun, S. 2003. Towards Robotic Assistants in Nursing Homes: Challenges and Results. *Robotics and Autonomous Systems*. 42: 271-281.

Poupart, P. 2005. Exploiting Structure to Efficiently Solve Large Scale Partially Observable Markov Decision Processes. Ph.D. thesis, Dept. of Computer Science, Univ. of Toronto, Toronto, Canada.

Spaan, M. T. J., and Vlassis, N. 2005. Perseus: Randomized Point-based Value Iteration for POMDPs. *Journal of Artificial Intelligence Research*. 24: 195-220.

Toronto Rehabilitation Institute. 2008. Stroke Rehabilitation Fact Sheet. http://www.torontorehab.com/documents/Stroke.pdf.