Dynamic Adjustment of Physical Exercises Based on Performance Using the Proficio Robotic Arm

Elham Saraee and Margrit Betke

Department of Computer Science, Boston University, USA

esaraee@bu.edu, betke@bu.edu

ABSTRACT

Physical therapy is an essential element in a comprehensive rehabilitation plan. Robot-assisted solutions have recently become more common to support the patient throughout the healing process. In this paper, we examine two techniques for assessing the movements a person performs when supported by a robotic arm. We propose DyAd, a Dynamic Adjustment system that recommends adjusted exercise configurations based on a person's performance. DyAd works with the Proficio arm, a commercial robotic arm designed for rehabilitation. The arm provides DyAd with kinematic data about the person's movement. DyAd has a graphical interface to support the user with a 3D visualization of the exercise trajectory to be followed. When a person tries to make a movement following the designed trajectory, DyAd compares this movement, the measured trajectory, to the designed trajectory. It evaluates their similarity using Dynamic Time Warping. DyAd also measures the smoothness of the movement using a modified version of the spectral arc length method. We designed DyAd to propose adjustments to the configuration of the exercise for specific users after each exercise trial. Finally, we validate the functionality of DyAd by presenting the changes in the difficulty level for users based on their performance.

CCS Concepts

•**Computing methodologies** → *Feature selection;* **Unsupervised learning; Online learning settings;** •**Human-centered computing** → **Haptic devices;**

Keywords

DyAd, Dynamic Difficulty Adjustment, Dynamic Time Warping (DTW), Spectral Arc Length, Human-Robot Interaction, Rehabilitation Technology

PETRA '16, June 29-July 01, 2016, Corfu Island, Greece

ISBN 978-1-4503-4337-4/16/06. . . $15.00

DOI: http://dx.doi.org/10.1145/2910674.2910724

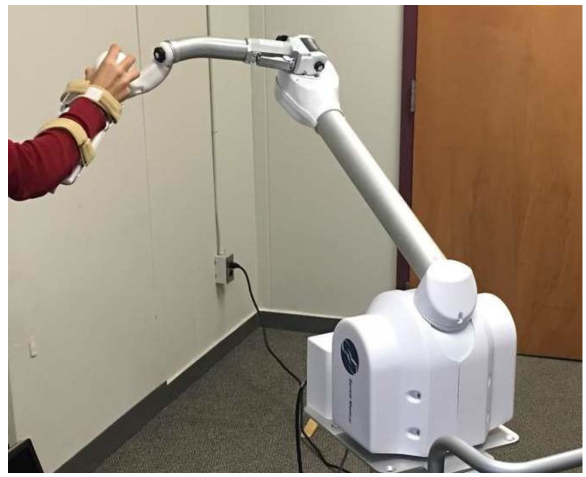


Figure 1. Exercise setup: The user's arm is strapped to the Proficio robot arm. In the back-drivable case, she controls the movement of the robot arm while grasping its end effector. In the other case, the robot guides her movement.

1. INTRODUCTION

Surgery, injury or increasing frailty due to aging are reasons one might need physical therapy. Robotic therapy is a beneficial treatment for individuals with chronic motor impairment [6]. Sport injuries also require planned programs improve the flexibility of the athlete and accelerate recovery. It has been shown that patients who have been provided with robot-aided rehabilitation have gained more from their therapy than those who were using only standard plans. Equipping therapist with robotic devices can increase productivity and quality of care as they provide quantifiable, safe and reproductive physical activity [23].

One of the crucial aspects of a rehabilitation program is to design a proper exercise for the patient to promote health while preventing any damage. The exercise should be safe while sufficiently challenging to maximize the patient's benefit and compliance [10,13]. Therapists require prior knowledge about the patient such as the level of impairment in order to design the proper exercise. Having quantitative measures of performance, provided by a robotic system, can help the therapist to track the level of recovery and changes in a patient's strength. A personalized plan capable of dynamically adjusting the level of difficulty of the exercises, based on quantitative measures, can significantly facilitate the rehabilitation process.

The Proficio, shown in Figure 1, is a commercially-available 3-degrees-of-freedom robotic arm. It is designed as a research instrument that facilitates the development of tools and methods for physical therapy. Intended users are patients recovering from accidents, stroke, and spinal-cord injury. The Proficio allows researchers to design exercises that involve reaching tasks and provide feedback using a 3D haptics force field. The Proficio robotic arm is a back-drivable manipulator, which means a user's arm strapped on the robotic arm can move the robotic arm (rather than the robotic arm moving the user's arm). The Proficio robotic arm is also capable of capturing data from the exercise to provide the therapist with valuable data about the performance of the patient.

In this work, we propose DyAd, a system that is motivated by two needs, (1) to evaluate the performance of individuals with impairments during physical therapy and (2) to provide recommendations for the therapist to reconfigure the settings to a level of difficulty appropriate for the individual. The components of our system can be summarized as follows:

- **Similarity of trajectories**: Dynamic Time Warping (DTW)[3] is a common method for aligning two trajectories and returning the error as a measure of dissimilarity between them [13]. DyAd uses DTW to compare the trajectories of the user's movement measured by the robotic arm with the trajectory the user was supposed to follow.
- **Smoothness**: DyAd uses a measure of smoothness to capture the concepts of continuity and nonintermittency of a movement. A recent study showed the importance of smoothness in the assessment of sensorimotor impairment and motor learning [2]. While a smooth motion is expected from someone without disability, individuals with poor motor control find moving smoothly quite challenging. A modified version of the spectral arc length measurement (ASL) [2] is used to evaluate this important parameter.
- **Dynamic Difficulty Adjustment**: After the assessment of an exercise trajectory using the aforementioned methods, DyAd recommends a new configuration for the difficulty level of the next exercise session.

The novel contribution of our work is the design and evaluation of a system that uses, in combination, a DTW score and a smoothness measure to analyze gesture trajectories, and then, based on this analysis, dynamically adjusts the difficulty of exercises.

The rest of the paper is organized as follows: In the next section, a brief review of related works regarding the robotic arm for rehabilitation is presented. It is then followed by a description of the performance measurement techniques that DyAd applies. In section 3, we also explain our dynamic difficulty adjustment method. The experiments are shown in section 4, and the paper ends with our conclusion and ideas for future work.

2. RELATED WORK

The goal of rehabilitation robotics is to provide effective approaches for enhancement of motor learning. A comparison of results between the use of robotics in rehabilitation and conventional techniques shows a significant difference in the recovery progress of people with motor disabilities. Volpe et al. [23] presented three studies that demonstrated the improved motor function of the paralyzed upper limb measured by clinical scales when robotic devices were utilized as part of a patients' rehabilitation plan. A systematic review of studies regarding the effect of robots in rehabilitation was given by Kwakkel et al. [12]. They have investigated various cases and suggested that there exits a positive trend toward robot-assisted therapy with regard to motor recovery when measured with common assessment scales. Another comparison was made for motor rehabilitation of the upper limb after stroke that provides encouraging evidence supporting the potentials in presence of robots to increase public healthcare burden and reduce in the expenses regarding post-stroke physical therapy [14].

Tele-rehabilitation is beneficial when the therapist cannot be present. Tele-rehabilitation can be helpful for patients in rural areas, for patients with limited mobility, or in settings where therapist involvement is considered to costly. Reinkensmeyer et al. [16], for example, introduced a web-based tele-rehabilitation system that consists of variety of games, tests and progress charts. To improve telerehabilitation, robotic devices have been proposed to be included in the therapy [8, 9].

A brain-robot interface is another approach to leverage robot-assisted technology [7]. As an integrative rehabilitation strategy, it decodes the subject's motor imaginary or movement attempt using EEG- or ECoG- based BCI shared-control strategy and enables artificial support of the sensorimotor feedback loop.

Dynamic difficulty-adjustment techniques have been extensively studied for commercial computer games to address problems such as difficulty gaps between levels or unresponsiveness of the system to player learning [19, 21]. However, these adaptive systems are not limited to computer games and have been recently the focus of research in rehabilitation as well.

An adaptive automated task-practice system was introduced by Choi et al. [5] that engages a user in realistic functional tasks based on a the user's performance. Another performance-based system [11] has been proposed that uses time, speed and EMG thresholds to trigger robot assistance in therapy. In that work, performance measures grade the patients' abilities to initiate movement, to move from starting point to the target, to aim their movement along the target axis and to reach the target position. An online component of the work tries to keep patients motivated during therapy session. Note that the approaches in measuring user performance in these two works significantly differ from ours.

A more recent work has focused on multimodal data such as speech, facial expressions, and body motion to take therapy progress into consideration and adjust subsequent exercises by formulating the problem as a Markov Decision Process [22].

Robotic arms can offer exercise support actively or passively. In the passive case, a patient does not need to exert any force, and the robot arm will move his or her arm based on the path prescribed by the therapist. In the active case, the patient moves the robotic arm himself or herself, and the robotic arm measures the kinematic and kinetics of the movement. A combined active-passive training system was proposed recently to lead a person through an exercise; a correctional force is applied when he or she deviates from the predefined trajectory [15]. In our work, we use the active approach, i.e., the user is actively moving the robotic arm. If the user deviates from the predefined trajectory because the exercise is too difficult for him or her, our system lowers the difficulty level of the subsequent exercise. In addition, our system uses haptic forces adaptively to provide the user with feedback.

3. APPROACH

In this section, we describe how DyAd combines two methods of analyzing movement trajectories. We use dynamic time warping [3] as a mean to compare the similarity of two trajectories, and the spectral arc length method to quantify the smoothness of a trajectory. We explain how we adapt Dynamic Scripting [20, 21], in particular, how we leverage the stochastic optimization idea behind Dynamic Scripting for DyAd, so that it can adjust the difficulty level of the prescribed exercise.

3.1 Dynamic Time Warping (DTW)

Dynamic Time Warping is a dynamic programming technique to measure the similarity between two temporal sequences. It aligns the signals by warping stretched or compressed signals in a non-linear fashion to find the optimal match between them. Although being originally used for speech recognition [17], DTW has been applied widely to other time-series analysis tasks, in particular, gesture recognition to compare and match temporal gesture sequences. We briefly illustrate DTW here. Assume two time series S and T represent two gesture trajectories:

$$S = s_1, s_2, ..., s_i, ..., s_n \quad (1)$$

$$T = t_1, t_2, ..., t_j, ..., t_m \quad (2)$$

The goal of DTW is to compute a warping path

$$W = w_1, w_2, ..., w_k, ..., w_p \quad (3)$$

that maps the elements of S and T such that the distance between them is minimized. W is a sequence of grid points where each point corresponds to an alignment between a point in S and another in T. The Manhattan or Euclidean distance functions are common for measuring the distance between two corresponding points:

$$\delta(w_k) = dist(s_i, t_j). \quad (4)$$

With this definition of the local distance between points, dynamic time warping can be formulated as a dynamic programming problem to minimize the following equation:

$$Error(S,T) = \min_w [\Sigma_{k=1}^{p} \delta(w_k)]. \quad (5)$$

The *Error* is then normalized by the length p of W.

An example of two joint position trajectories, measured by the robotic arm, is shown in Figure 2. DyAd matches corresponding points on the trajectories using DTW. The normalized *Error* was 0.126, small enough so that DyAd concludes that the current performance is sufficient. DyAd then adjusts the difficulty level of the exercise depending on the result of the smoothness evaluation and exercise timing, as described below. We have used the original Dynamic Time Warping code released by Salvador and Chan [18] to compare our gesture time series.

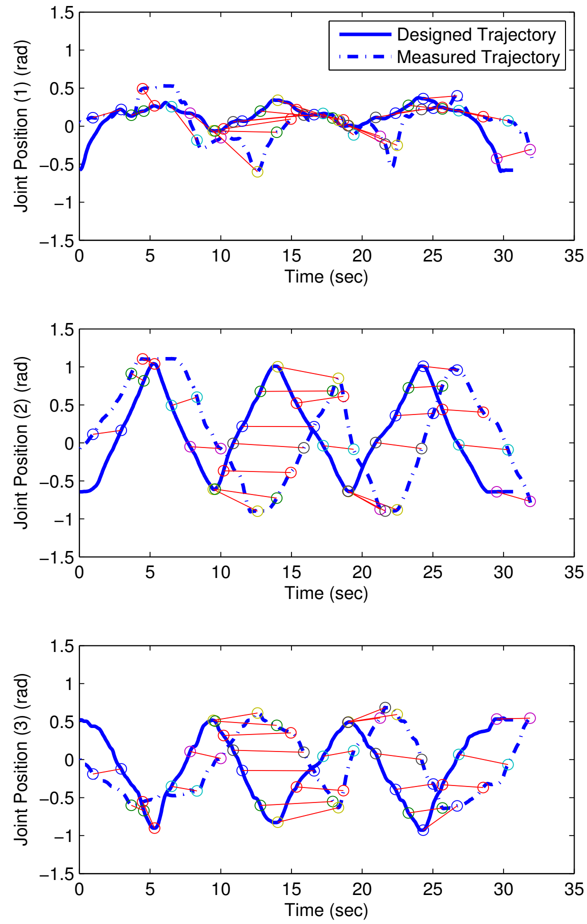


Figure 2. Matching two gesture signals using DTW. The movement trajectories that two users, a teacher and a student, created during an exercise. Three components of the two trajectories are shown as functions of time. They correspond to the angles that the three joints of the robot arm make with respect to each other, using the Denavit-Hartenberg convention. The teacher designed a desired trajectory (solid blue), the student tried to follow the same trajectory (dashed blue). DTW is used to find corresponding joint positions (red). The similarity of the trajectories is then measured by the Euclidean distance between matching positions.

3.2 Smoothness

Smoothness has been identified as a fundamental parameter to assess the scale of a motor impairment, as it has been shown to correlate significantly with a patient's ability to control the movement [4]. A smooth motion is considered to be a continual movement without any interruption. Measuring smoothness quantitatively, with the help of a robotic arm, instead of simply relying on qualitative visual observations by the therapist or comments by the patient, may yield a better understanding of the level of movement control that a patient has. It should be noted though that intermittency might be caused by poor task familiarity, and a therapist must validate the results based on experiments that eliminate this factor.

The assessment of motor control in terms of smoothness requires a range of scores to reflect the motor ability of individuals. However, it is not possible to define a specific range, because depending on the kind of task, this range will vary.

The literature discusses various methods about how to measure smoothness. The harmonic ratio, jerk-based methods, or the spectral arc length method [2] are common methods. The spectral arc length method [1] measures the arc length in the Fourier magnitude spectrum within a fixed range of a speed profile indicated as ω_c in the following equation:

$$SAL \triangleq \int_0^{\omega_c} \left[(\frac{1}{\omega_c})^2 + (\frac{d\hat{V}(\omega)}{d\omega})^2 \right]^{1/2} d\omega, \tag{6}$$

where $\hat{V}(\omega) = \frac{V(\omega)}{V(0)}$, $V(\omega)$ is the Fourier magnitude spectrum of the speed profile, $V(0)$ is the value of this spectrum at frequency 0 (also called DC magnitude), and the arc length is estimated within frequency 0 and a fixed ω_c equal to 20 Hz. A slightly modified version of this method was proposed [2] that adaptively selects the range of frequencies based on specific constraints in the movement:

$$w_c \triangleq \min\{w_c^{\max}, \min\{w, \hat{V}(r) < \bar{V},\ \forall r > w\}\}, \tag{7}$$

where $\bar{V}$ is threshold on the Fourier magnitude. The use of an adaptive method is helpful as it makes it possible to set these parameters based on the nature of the movement.

3.3 Dynamic Difficulty Adjustment (DDA)

The exercises prescribed by the therapist must not only be safe and achievable, but they should be sufficiently challenging to motivate the patient to work harder and make progress. In this part, we first explain dynamic scripting, the algorithm we are inspired by, and then propose a method for dynamic difficulty adjustment that DyAd uses to adjust an exercise according to the movement ability of an individual.

3.3.1 Dynamic Scripting

Dynamic scripting, proposed by Spronck et al. [19], is an unsupervised online learning algorithm inspired by reinforcement learning that uses stochastic optimization. Online learning must be fast, effective, robust ; these conditions exclude many regular learning algorithms because they would be too slow.

Dynamic scripting was originally introduced for complicated computer games, since scripts can be executed sequentially to overcome the complexity issue. Dynamic scripting maintains several rule bases, corresponding to separate opponents in a game. Rules are extracted from the rule base to form a script that describes the behavior of the opponent as it is generated. The probability that a rule is selected from the rule base depends on the weight assigned to it. These weights are updated based on previous performance of the rule in the script. It will be rewarded with increase in weight in case of success or punished by decrease in the weight if it has caused failure. The reward or punishment values are calculated using a fitness function that relates the success and failure of the rule to the change in the weights.

We take advantage of the stochastic optimization idea behind dynamic scripting. For DyAd, we employ a deterministic fitness function to map the performance measures to rules. We propose to obtain the parameters required for reconfiguration of the difficulty level in a stochastic manner using a random distribution function biased by the values suggested by the fitness function. A more detailed explanation of this method is discussed in the next section.

3.3.2 DyAd: Dynamic Adjustment of Difficulty

Three parameters, denoted by vector $\vec{x}$, capture the quantitative performance measures that DyAd uses to decide on recommendations for exercise adjustment. (1) DyAd compares the the measured trajectory with the designed trajectory and reports the DTW error as a measure of dissimilarity. (2) DyAd uses the measured velocity values along the trajectory of the motion to evaluate the smoothness of the motion using Equation 6. (3) DyAd uses the total recorded time to monitor the performance of a user.

The domain knowledge is now required to map the different range of $\vec{x}$ to a score between zero and one. DyAd employs two nonlinear functions, the sigmoid for parameter (1) and the logarithm for parameter (2) and (3), to serve this purpose. To illustrate more, in case of sigmoid function, DyAd first uses

$$score_{x_i} = \frac{1}{1 + exp^{-(\alpha x_i + \beta)}}, \tag{8}$$

where constants α and β control the application of domain knowledge. Different values for α and β can lead to completely different scenarios. They can bias the score function such that it becomes easy for a user to receive a high performance score. The system would then label the user's current performance acceptable and would make the exercise more challenging in the next trial. With another set of values for α and β, a motion trajectory may be given a low score and then DyAd would recommend an easier exercise until the user's performance is acceptable.

The therapist can change the values for α and β depending on the condition of patients and what is expected from them in that session of therapy.

In order to relate the quantified performance results to the level of difficulty for the next exercise, DyAd uses three variables, denoted by $\vec{y}$: (1) the path index, (2) the size of the ball, and (3) the resistance level of the haptic-feedback walls along the path. The parameters will be explained in more details in the next section. Similar to the dynamic scripting algorithm, DyAd defines a fitness function

$$score_y = A \times score_x, \tag{9}$$

where A is a 3 by 3 matrix corresponding to the contribution of each element of $\vec{x}$ to $\vec{y}$.

We here propose a method that uses the normal distribution $\mathcal{N}$ to combine domain knowledge, i.e., an understanding of the current performance of the motion of the user, with a need to have a random component for the exercise. DyAd randomly selects the value

$$result = \mathcal{N}(\mu, \sigma) \tag{10}$$

for the difficulty level in the next trial, where the inputs and output are defined as follows:

- The **mean** μ is obtained from $score_y$ by a linear function $f(score_y) = (1 - score_y) + 0.5$ that maps scores between 0 and 1 to the desired maximum and minimum of the changes in the difficulty parameters for the next trial. For example, when the score is 0.3 for the size of the ball (less than 0.5), it reflects the need

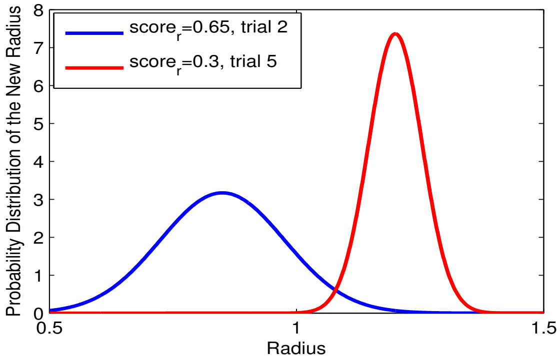


Figure 3. Two examples of normal distribution $\mathcal{N}$ applied to suggest a new size of the ball (Eq. 10). The blue curve represents the distribution for the second trial; a score of 0.65 was obtained for the radius of the ball; since the score is acceptable (above 0.5), the distribution is biased towards smaller balls on average. Similarly, the red curve is the distribution for suggested balls in the fifth trial; a score of 0.3 was obtained for the radius of the ball; larger balls are the result of lower scores (less than 0.5). The examples show how the shape of the distributions is affected by the number of previous trials: the variance becomes smaller as the users go through more trials.

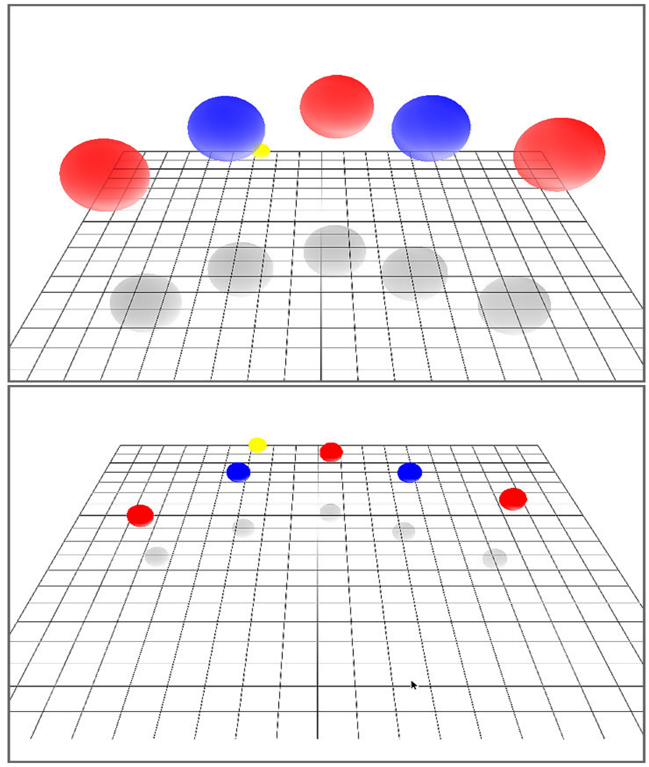


Figure 4. Two different configurations of the balls. The small yellow ball indicates the position of the handle of the arm. Users begin from the red ball on the right. They are asked to move their arm to the left to pass all five balls and move their arm back to right, repeating this motion three times. They experience the haptic force as resistant walls placed around the blue balls only. In order to help visualize the 3D space, we designed our display to show the projections of all the balls on the z-plane. When the balls are larger and closer to the users, it will be easier to move through them; as users manage to reach the required performance, the balls get smaller and farther to make the exercise more challenging for them.

for a larger ball in the next trail, and thus, the mean is mapped to $\mu = 1.2$.

- The **variance** σ of the normal distribution corresponds to how much DyAd wants to explore various cases. In our design, a wider distribution (larger σ) corresponds to lots of exploration; a narrower distribution (smaller σ) corresponds to exploitation of the current knowledge about the performance of the user. Thus, the variance is a decreasing function of number of trials: $\sigma = \sigma_0 \times (1 - \log(trialnumber))$ for a trial number between 1 and 9, and a fixed value after that.
- The **output** *result* of the random function $\mathcal{N}$ determines the changes in the $\vec{y}$ parameters. For instance, in Figure 3, the red curve recommends a value between 1 and 1.4 which is the ratio in the increase of the size of the ball.

4. EXPERIMENT

4.1 Methodology

In order to facilitate the visualization of the exercise, we have implemented a 3D virtual exercise interface (Figure 4). Users are asked to move their right arm in a semi-circular shape, moving through the virtual balls shown in the interface, while strapped to the Proficio robotic arm.

As described above, the size of the ball is one of the parameters that DyAd adjusts. Larger balls are preferred in the first trial, in order to direct the participants to the correct path. Smaller balls are suggested when participants can make smooth and accurate motions. Two examples of ball configurations, one with large and the other with small balls, are shown in Figure 4.

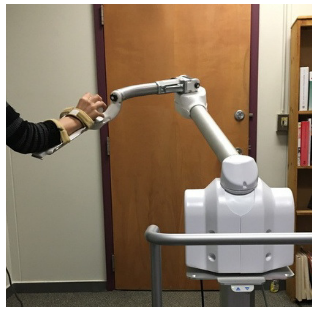
Depending on a user's previous performance, DyAd recommends new positions for the balls, which also yield different levels of difficulty. New ball positions require the user's arm to be held in a new plane during its semi-circular movement. Horizontal movements are considered to have a medium difficulty level. Easier configurations consist of balls placed so that the user's arm is directed upward during the motion. The easiest level corresponds to an arm inclination of 45 degrees and is referred to as level 1 later in the paper. A downward direction of the arm is considered difficult, since the users have to stretch more to reach the virtual balls. In the most difficult configuration, the balls are placed around 45 degrees downward (referred to as level 8).

Starting arm positions are exemplified for three different difficulty levels in Figure 5. The user begins the exercise by following the easiest path shown in Figure 5(a). The user then is asked to exert more control and follow the designed paths in Figure 5(b) and (c).

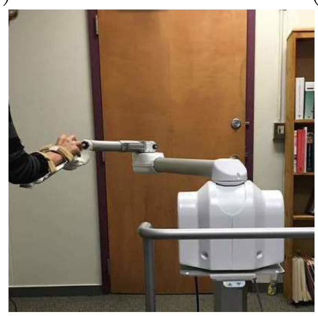
We designed two force-feedback walls along a movement path. They are the surfaces of the blue balls in the virtual world shown Figure 4. They require the user to exert a certain force in order to pass through. The strength of these walls, the haptic force, is a parameter that DyAd changes depending on the user's performance.

As stated earlier, the score function is determined based

(a)



(b)



(c)

Figure 5. Three exercise variations. The shown arm directions correspond to three difficulty levels. The user experiences the easiest in (a). As the arm points downward (b), the movement becomes harder, since the user has to stretch the arm more and keep balance. The configuration shown is (c) is the most difficult of the three shown configurations.

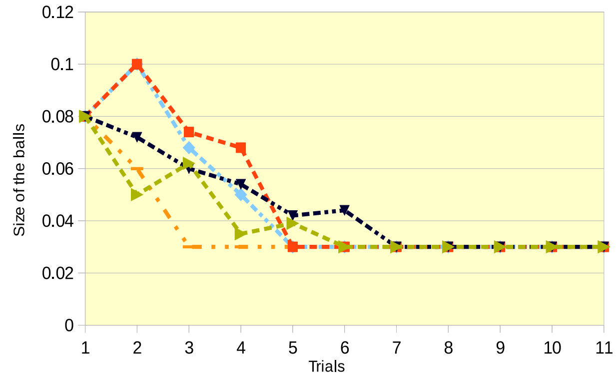


Figure 6. Adjustment of ball size during ten trials (here for visualization clarity, only shown for five participants). In the first trial, the experimental configuration is initialized with a large ball size (radius = 0.08). If a user's performance is acceptable, then the system reduces the ball size. All ten participants reached the game level with the smallest possible size of the balls in less than ten trials.

on domain knowledge. In our experiment, the parameters α and β in Equation (8) were set in such a way that on average users without movement impairments should be able to explore all levels of difficulty.

It is also worth noting that we did not apply any constraints on the width of the temporal signal in the DTW algorithm, since it is probable that an individual decides to rest and stops moving the arm, which would result in a stretched temporal signal. This signal is supposed to be matched to another relatively compressed temporal signal. Consequently, a constrained algorithm might not be able to detect the similarity of these two gesture signals.

We asked ten individuals with no physical disabilities, five males and five females in their mid and late twenties, to test DyAd. The volunteers stood besides the robotic arm and looked at the monitor with the virtual interface, which was placed in front of them. We asked them to move their arms so that they would move the yellow ball through the five red and blue balls, back and forth, three times in each trial. The experiment consisted of ten trials, each with a new parameter updated by DyAd.

4.2 Results

Our results suggest that DyAd performed as we planned with regards to adapting to the ability of users to move through the virtual space. Our idea to display virtual balls along the desired trajectory seemed to have helped the users to more clearly visualize the path they were asked to take. As the users became familiar with the movement of the yellow ball that corresponded to their arm movement and obtained a better scores, DyAd appropriately reduced the size of the red and blue balls.

We show the results of the experiment in Figures 6–8 for only five out of ten participants for visualization clarity. We chose the displayed subset of results in such a way that they can fully explain the measured patterns.

All users reached the smallest expected size of balls as shown in Figure 6. For some of them, DyAd measured a better performance in the first trials which led to smaller balls in the earlier trials. Others had difficulty finding the path and could not get higher scores for the size of the ball; thus DyAd recommended larger balls in the beginning and then moved towards the smaller balls as the users made progress in the next trial.

DyAd recommendations for ball positions has a similar pattern as its recommendations with regards to ball size. Users were able to reach the hardest level of difficulty in the position of the balls as shown in Figure 7. One of the participants reached the hardest level in three trials as he finished the exercise fast, accurate and smooth. On average, it took five trials for users to reach the farthest possible configuration of the balls.

DyAd recommendations for changing the strength of the haptic force did not have the same pattern as its recommendations for ball size or position. Here, smoothness turned out to be the most effective parameter that directly affects the strength of the walls. Some users reached the highest possible value. Others found it difficult to be smooth, as shown in Figure 8.

4.3 Discussion

Our experiment showed that DyAd is able to make adjustments dynamically. If the users were confused or found an exercise difficult, DyAd recommended a decrease in the level of difficulty to help the users find the required path. Then, as their performance improved, DyAd asked the users to try more difficult movements.

The volunteers who tested DyAd had no motor disabilities. We accounted for that when we designed the virtual world and the range of exercise difficulties. Accordingly, all

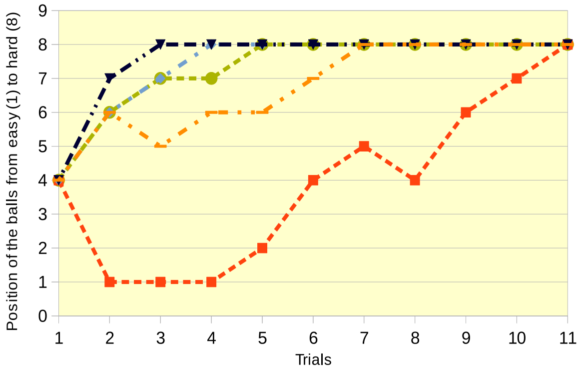


Figure 7. DyAd recommendations for changes in the difficulty level of positioning of the balls (for visualization clarity only shown for five out of ten participants). The vertical axis represents the level of difficulty from easiest (level 1) to most difficult (level 8). Every user began with the horizontal direction of the arm (level 4). For those users who could follow the designed trajectory closely and with sufficiently smooth motion, DyAd suggested a more challenging setup, where the user's arm points downward (level 5 or higher). One participant (red results) could not follow the correct path in the first two trials, and DyAd recommended the easiest path with the user's arm pointing upward (level 1) in subsequent trials.

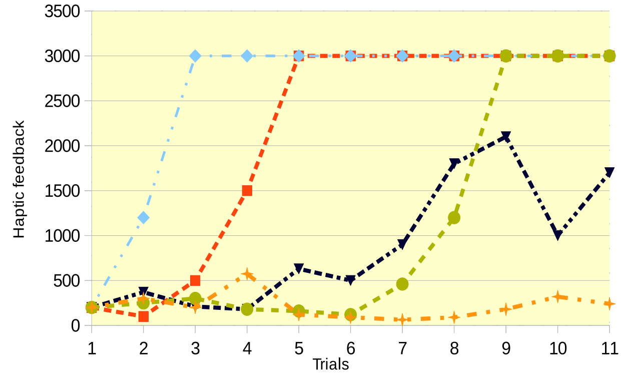


Figure 8. Change of the strength of the haptic force (walls) at each trial, as proposed by DyAd in response to users' progress (shown for five out of ten users). The strength of the walls was small in the beginning of the experiment, and it increased as users made progress. When a user's performance was not acceptable (orange), the strength of the walls did not increase.

testers were able to reach the highest level of difficulty, even if it took them a number of trials. It is worth noting that our results depended on how the score functions were initially defined for each performance measure and how they were mapped to difficulty levels.

DyAd is designed to give therapists flexibility. The score and fitness functions can be modified based on what outcomes are expected from a specific user and a specific exercise. DyAd supports different approaches: one can use a strict score function that accepts a result only if the user performs an exercise highly accurately, fast, or smoothly. Or, one can allow minimal performance and then challenge users with additional exercises. The sensitivity of DyAd to its input parameters, however, can also be considered a drawback. Prior knowledge about the possible values of smoothness or similarity of trajectories is required to label the measured values as progress or failure. In the future, we will explore ideas about how to automatically define and tune the score and fitness functions so that a therapist could take advantage of the flexibility that DyAd provides but would not need an in-depth understanding of its technical issues.

Finally, we should discuss that when using DTW and matching a stretched measured trajectory to a designed trajectory, DyAd can determine if a user is taking a rest. This analysis was not performed in our experiment. It may be beneficial to evaluate the movement of users with disabilities, since it reveals if a part of the exercise is so difficult that has made the user to stop moving the arm.

5. CONCLUSIONS

We have designed DyAd, a system that dynamically adjusts the difficulty level of physical exercises performed with the Proficio robotic system and based on quantitative performance measurements. We have shown that DyAd recommends a more difficult arm exercise if the current exercise is not challenging enough; it recommends an easier exercise if it discovers that the user has trouble following the current exercise.

The approach to use DTW to align time series of gestures has been stated before in the literature. Novel contributions of our work are (1) applying DTW in conjunction with the spectral arc length method to analyze trajectories and (2) designing a system that dynamically adjusts the difficulty of exercise based on this analysis.

In the experiment reported in this paper, users without disabilities tested the efficacy of our proposed system DyAd. The experimental outcome encourages us to prepare for and conduct a study that includes therapists and their patients. Our future work will also include the design of appropriate exercises for DyAd, as well as virtual interfaces that enable tele-rehabilitation.

6. ACKNOWLEDGMENTS

The work was supported in part by NSF (1337866). We would also like to thank the ten volunteers who tested our DyAd system.

7. REFERENCES

[1] S. Balasubramanian, A. Melendez-Calderon, and E. Burdet. A robust and sensitive metric for quantifying movement smoothness. *IEEE Transactions on Biomedical Engineering*, 59(8):2126–2136, 2012.

[2] S. Balasubramanian, A. Melendez-Calderon, A. Roby-Brami, and E. Burdet. On the analysis of movement smoothness. *Journal of neuroengineering and rehabilitation*, 12(1):1, 2015.

[3] D. J. Berndt and J. Clifford. Using Dynamic Time Warping to Find Patterns in Time Series. In *Knowledge Discovery and Data Mining (KDD) Workshop*, pages 359–370, 1994.

[4] C. Bosecker, L. Dipietro, B. Volpe, and H. I. Krebs. Kinematic robot-based evaluation scales and clinical counterparts to measure upper limb motor performance in patients with chronic stroke. *Neurorehabilitation and neural repair*, 2009.

[5] Y. Choi, J. Gordon, and N. Schweighofer. ADAPT - adaptive automated robotic task practice system for stroke rehabilitation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2471–2476, 2008.

[6] S. E. Fasoli, H. I. Krebs, J. Stein, W. R. Frontera, and N. Hogan. Effects of robotic therapy on motor impairment and recovery in chronic stroke. *Archives of physical medicine and rehabilitation*, 84(4):477–482, 2003.

[7] M. Gomez-Rodriguez, M. Grosse-Wentrup, J. Hill, A. Gharabaghi, B. Schölkopf, and J. Peters. Towards brain-robot interfaces in stroke rehabilitation. In *IEEE International Conference on Rehabilitation Robotics (ICORR)*, pages 1–6, 2011.

[8] M. J. Johnson, R. C. Loureiro, and W. S. Harwin. Collaborative tele-rehabilitation and robot-mediated therapy for stroke rehabilitation at home or clinic. *Intelligent Service Robotics*, 1(2):109–121, 2008.

[9] M. J. Johnson and H. Schmidt. Robot assisted neurological rehabilitation at home: Motivational aspects and concepts for tele-rehabilitation. *Public Health Forum*, 17(4):8.e1 – 8.e4, 2009.

[10] L. Kahn, W. Rymer, and D. Reinkensmeyer. Adaptive assistance for guided force training in chronic stroke. In *26th Annual International Conference of the IEEE in Medicine and Biology Society (IEMBS)*, volume 1, pages 2722–2725, 2004.

[11] H. I. Krebs, J. J. Palazzolo, L. Dipietro, M. Ferraro, J. Krol, K. Rannekleiv, B. T. Volpe, and N. Hogan. Rehabilitation robotics: Performance-based progressive robot-assisted therapy. *Autonomous robots*, 15(1):7–20, 2003.

[12] G. Kwakkel, J. Kollen, and H. Krebs. Effects of robot-assisted therapy on upper limb recovery after stroke: A systematic review. In *Neurorehabilitation and Neural Repair*, pages 111–121. ISSN, 2008.

[13] A. Lioulemes, P. Sassaman, S. N. Gieser, V. Karkaletsis, F. Makedon, and V. Metsis. Self-managed patient-game interaction using the Barrett WAM arm for motion analysis. In *Proceedings of the 8th ACM International Conference on Pervasive Technologies Related to Assistive Environments (PETRA)*, pages 34:1–34:8, 2015.

[14] P. S. Lum, C. G. Burgar, P. C. Shor, M. Majmundar, and M. V. der Loos. Robot-assisted movement training compared with conventional therapy techniques for the rehabilitation of upper-limb motor function after stroke. *Archives of Physical Medicine and Rehabilitation*, 83(7):952 – 959, 2002.

[15] S. Phan, A. Lioulemes, C. Lutterodt, F. Makedon, and V. Metsis. Guided physical therapy through the use of the Barrett WAM robotic arm. In *IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE)*, pages 24–28, 2014.

[16] D. J. Reinkensmeyer, C. T. Pang, J. A. Nessler, and C. C. Painter. Web-based telerehabilitation for the upper extremity after stroke. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 10(2):102–108, 2002.

[17] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 26(1):43–49, 1978.

[18] S. Salvador and P. Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[19] P. Spronck, M. Ponsen, I. Sprinkhuizen-Kuyper, and E. Postma. Adaptive game AI with dynamic scripting. *Machine Learning*, 63(3):217–248, 2006.

[20] P. Spronck, I. Sprinkhuizen-Kuyper, and E. Postma. Online adaptation of game opponent AI in simulation and in practice. *Proceedings of the 4th International Conference on Intelligent Games and Simulation*, pages 93–100, 2003.

[21] P. Spronck, I. Sprinkhuizen-Kuyper, and E. Postma. Difficulty scaling of game AI. *Proceedings of the 5th International Conference on Intelligent Games and Simulation (GAME-ON)*, pages 33–37, 2004.

[22] K. Tsiakas, M. Huber, and F. Makedon. A multimodal adaptive session manager for physical rehabilitation exercising. In *Proceedings of the 8th ACM International Conference on Pervasive Technologies Related to Assistive Environments (PETRA)*, pages 33:1–33:8, 2015.

[23] B. T. Volpe, M. Ferraro, H. I. Krebs, and N. Hogan. Robotics in the rehabilitation treatment of patients with stroke. *Current Atherosclerosis Reports*, 4(4):270–276, 2002.