2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

ADAPT – Adaptive Automated Robotic Task Practice System for Stroke Rehabilitation

Younggeun Choi, James Gordon, and Nicolas Schweighofer

Abstract—**We propose a new robotic task practice system designed to enhance the recovery of upper extremity functions in patients with stroke. Our system, ADAPT (ADaptive and Automatic Presentation of Tasks), which was designed in accordance to current training guidelines for stroke rehabilitation, engages the patient intensively, actively, and adaptively in a variety of realistic functional tasks that require reaching and manipulation. A general-purpose robot simulates the dynamics of the functional tasks. Based on the subject's performance, a task scheduler adaptively selects a task and sets the task difficulty. The tool changer selects the tool corresponding to the selected task, a doorknob for instance. The low-level controller then implements the selected task with the desired difficulty on the robot during the robot-patient interaction. Our preliminary experimental results demonstrate the feasibility of our system.**

I. Introduction

Stroke is the leading cause of disability among American adults. Over 80% of first-time strokes (infarctions only) involve acute hemi-paresis of the upper limb [1]. Because a substantial number of activities of daily living involve use of the upper extremities [2], rehabilitation of reach and grasp skills is critical for patients in their attempts to return to a reasonable quality of life [3].

Recognizing that intensive motor practice is beneficial for recovery of upper extremity functions [4-6], and that current medical practice does not adequately allow for the required training intensity [7, 8], a growing number of investigators have been developing robotic systems for the rehabilitation of upper extremities after stroke. In these systems, such as the MIT-MANUS [9], the mirror-image motion enabler robot (MIME) [10], the ARM-guide system [11], and the Bi-Manu-Track [12], the robot assists the movements of the affected limb. These robots can enhance performance and function in patients post-stroke by providing intensive, cost-effective, rehabilitation, e.g. [13, 14]. Recent developments include using robots that allow re-training of multiple joints [15], adaptive algorithms that balance robotic assistance and patients' active movements, e.g. [16], and EMG triggered robots [17].

Although shown to be effective to some extent, these systems do not exactly parallel the role of rehabilitation therapists, who typically expend considerable effort to set up functional tasks that require a patient to actively engage in challenging reach and grasp practice [18]. To be effective, the tasks should be meaningful [19-21] and should involve the manipulation of real and functional objects. AutoCite [22], a semi-automated (non-robotic) system that allows patients to engage in the practice of such tasks, has recently been shown to be as effective as standard Constraint Induced (CI) therapy [23]. In AutoCite, however, the number of tasks cannot be increased easily, task selection and scheduling is manual, and task difficulty is not adjusted automatically.

Here we propose a new system, ADAPT (ADaptive and Automatic Presentation of Tasks), that presents functional tasks automatically, that can accommodate an expanding number of tasks, and that allows the implementation of adaptive performance-based task scheduling and adaptive modification of task difficulty. As AutoCite, but unlike most other robotic systems, ADAPT does not move the patients; instead, it adapts the tasks, such that each patient can perform doable, but constantly challenging tasks. The system is designed for patients with some volitional motor capability of the arm and hand, as those patients benefit the most from intensive rehabilitation [3].

The primary focus of this paper is to present an overall conceptual design, a control architecture, and preliminary results obtained with the current implementation of ADAPT.

II. Design of ADAPT

A. Overall conceptual design

Fig. 1 presents an overview of the ADAPT system. A general-purpose robot simulates the dynamics of functional tasks. Based on previous performance, the *task scheduler* adaptively selects a task and sets the difficulty of the task. The tool changer selects a tool corresponding to the selected task, a doorknob for instance. The low-level controller then implements the selected task with the desired difficulty on the robot during the robot-patient interaction.

B. High-level Controller: Adaptive Task Scheduler

Adaptive task schedules: Although individual tasks can be scheduled sequentially, random scheduling of several tasks

Manuscript received September 14, 2007. This work was supported in part by the Division of Biokinesiology and Physical Therapy, University of Southern California and by NIH grant R03 HD050591-02

Younggeun Choi is with the Department of Computer Science, Los Angeles, CA 90089 USA (phone: 323-442-1270; fax: 323-442-1515; e-mail: younggch@usc.edu).
James Gordon is with the Division of Biokinesiology & Physical Therapy, University of Southern California, Los Angeles, CA 90089 USA (e-mail: jamesgor@usc.edu).
Nicolas Schweighofer is with the Division of Biokinesiology & Physical Therapy and the Department of Computer Science, University of Southern California, Los Angeles, CA 90089 USA (e-mail: schweigh@usc.edu).

has been shown to enhance motor learning performance, as measured in delayed retention tests compared to sequential, or blocked, scheduling [24, 25]. Random task scheduling however does not account for differences in difficulty between tasks. Depending on nominal task difficulty and skill of the learner, the rate of performance improvement varies from task to task for each learner. Accordingly, we recently showed that an algorithm that adaptively determines the number of trials for each task, based on learner's performance, outperforms fixed random scheduling [26].

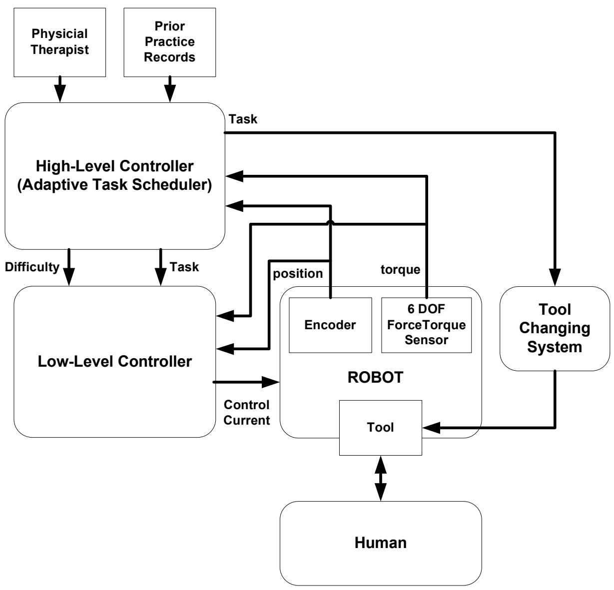


Fig. 1. The ADAPT system. At each trial, the task scheduler adaptively chooses a task and its difficulty based on previous subject's performance, prior practice records, and physical therapist's input. The tool changing system automatically selects the tool corresponding to the selected task. The low-level controller computes the control current needed to simulate the desired task dynamics during robot-patient interaction.

Because manual change would limit the ability to implement adaptive random task schedules (in which tasks should be quickly changed) and limit the cost-effectiveness of the system, ADAPT is able to switch between tasks rapidly using a tool changer (see below)

Adaptive task difficulty: The difficulty of each task is defined by parameters such as stiffness, damping, position, or range of motion. Challenging tasks, but not tasks that are too difficult or too easy, are most likely to elicit motor learning [27-30]. Challenging tasks also enhance motivation, which may in turn further enhance learning [31]. Because patients' performance will usually improve during rehabilitation, and because re-learning evolves at different rates for each task and each subject, the challenge needs to be dynamically maintained. We recently showed in a learning experiment involving multiple visuomotor tasks that adaptive difficulty algorithms outperform fixed difficulty [26]. Thus, we adopt a similar approach to adapt task difficulty as a function of performance (see result) – furthermore, at each session, initial task difficulty depends on prior performance records and possibly the therapist's input.

Once the task has been selected, the tool chosen and mounted on the robot with the tool changer, and the task difficulty determined, the low-level controller simulates the dynamics of the task on the robot (see below).

C. Low-level Controller

Detailed structure of the low level controller is shown in Fig. 2. Because we use a general-purpose robot that has low back-drivability, we used admittance control to compute the control signal for the motion response to external force. The control signal is the motor current, and is computed from the output of two feed-forward control modules (see Figure 2). The first module is a controller for motion without interaction force (motion model), and the second is for a controller for interaction force without motion (force model). We trained both modules trained off-line with Receptive Field Weighted Regression (RFWR) [32]. RFWR generates locally linear models, and combines them for the inverse dynamics prediction of the two modules

The motion model is trained with a typical sinusoidal excitation without any interaction. In case of the force model, while the subject exerts force in the position controlled state of the robot, the exerted force and the current from the robot motor are recorded to be used as training data. The motion data (angle, angular velocity, angular acceleration) and the control current are regressed by RFWR to provide inverse dynamics of model of only motion model. The force model is regressed similarly with force data (torque) and the control current for the positioned control state.

Our novel control architecture has three main advantages: First, learning the control currents directly allows accurate control even when, as in our system, the robot's dynamics are unknown and when the control current is not proportional to the motor torque. Second, our modular controller simplifies the training of the controllers (compared to a combined controller) because separation of force and motion models allows us to train each model with fewer training data points. Third, the force module can be used for the implementation of "virtual walls" (which is required for some tasks such as

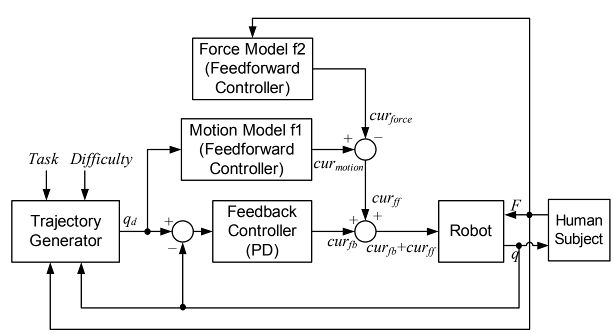


Fig. 2. Low-level controller with admittance control scheme. The controller generates the desired motion in response to the external force by the subject. The measured external force is input to the trajectory generator that defines the dynamics of the functional tasks. Based on the dynamics of the task, the trajectory generator generates the ideal desired motion. f1 computes the control current for the desired motion. and f2 computes the control current to compensate against the external force.

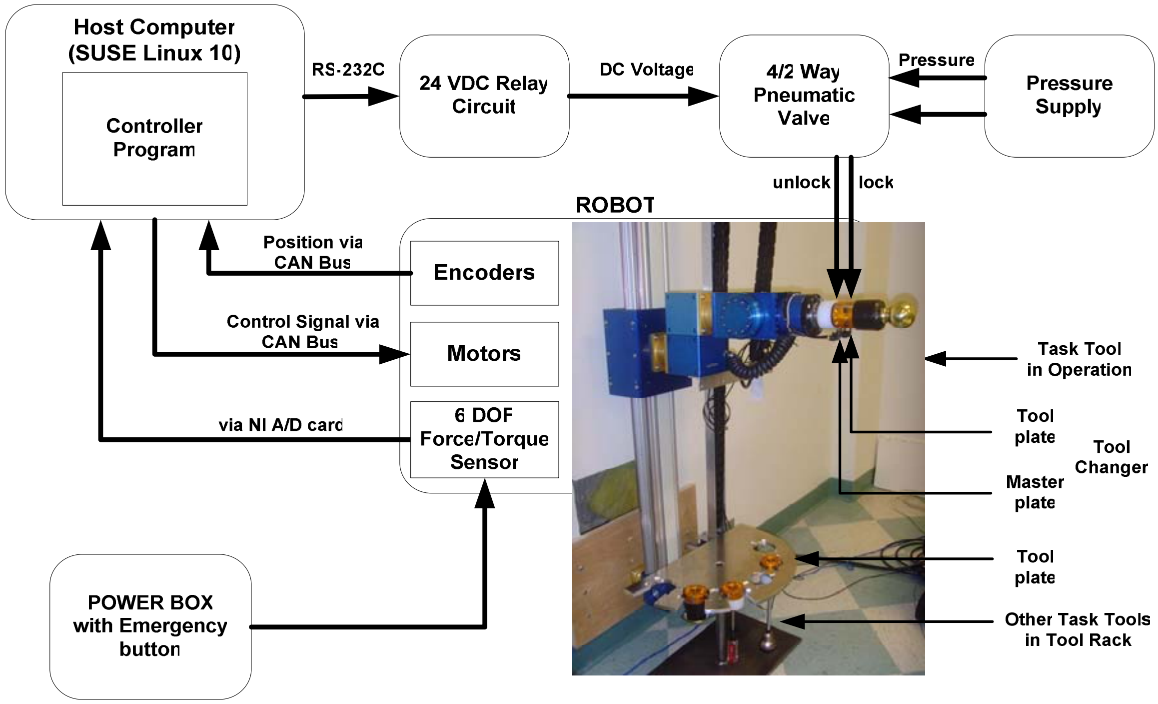


Fig. 3. The robot and the tool changer in the current implementation of ADAPT.

turning a key), because the module can simulate infinite stiffness successfully.

D. Robot and tool changer

The current robot used in ADAPT (Fig. 3) is a reconfigurable robot from AMTEC Robotics [33]. It consists of a 3-DOF wrist mounted on a 1-DOF linear actuator. This configuration allows the system to present the end-effector at different linear vertical locations and to rotate the end-effector in almost any orientation. A 6-DOF ATI Force/Torque (F/T) sensor (MINI SI-580-20) is attached to the end-effector to measure interaction forces between the subject and the robot. The tool changer in ADAPT connects a functional task tool, such as a doorknob, to the F/T sensor. Encoders in the motor module provide position data of each joint. A Pentium-4 3.4Ghz PC with a Linux operating system (SUSE 10) receives the position data via a CAN bus, receives the interaction force data via a National Instrument A/D converter, and sends control commands via the CAN bus.

Task tools such as a doorknob, a screwdriver, a jar, a faucet, keys, etc. are arranged in a tool rack, and when a task is scheduled, the robot re-positions the current tool and picks up another tool from the rack. A tool changer is locked or unlocked by a 4/2 way pneumatic valve (V5A-3341-BX1, MEAD corp), which is computer-controlled via RS-232C, a serial communication. The tool changer can switch between the tool in use and a tool in the tool rack automatically with the task scheduler in the PC.

As shown in Fig. 3 the robot's end effector is equipped with a master plate, and each tool with an interface plate from ATI corp. In the current version, four tools are arranged in the rack, and up to six additional tools can be included. A pneumatic system ensures the locking and unlocking of tools. The pneumatic system is automatically controlled via serial communication by the controller program in the PC.

The tool changing process is demonstrated in Fig. 4. After the scheduler chooses a task, the 4 DOF robot positions each joint so that it can pick up the tool in the tool rack. Then, the PC sends a lock command to a 24 VDC relay circuit, which sets the direction of the air flow in the 4/2 way pneumatic valve so that the master plate of the tool changer docks with the tool plate. After the two plates are docked, the robot repositions to present the tool to the subject. When the task scheduler decides a new task to be practiced, the robot positions the current tool in the tool rack, picks up a new tool, and presents it to the subject. The subject is seated in a chair facing the robot with an in-between table, lays his/her hands on the table, and practices the tasks simulated by the robot.

E. Safety

Safety was a crucial issue in the design of ADAPT. From the initial robot design process, we made special efforts to guarantee operational safety. Our choice of design makes our robot safer than a traditional multi-DOF robotic arm because of the small overall workspace. The linear DOF is only used for tool positioning, not for task dynamics simulation. Furthermore, the patient is not strapped to the robot.

For simplicity and safety reasons, we chose tasks that require movements around a single DOF during subject-robot interactions. Because many functional tasks in daily life (such as turning a key or door handle, screw driving, steering, opening a jar, turning a water faucet, wrist supination/pronation, etc.) need only a single rotary DOF, this configuration does not overly restrict the available number of

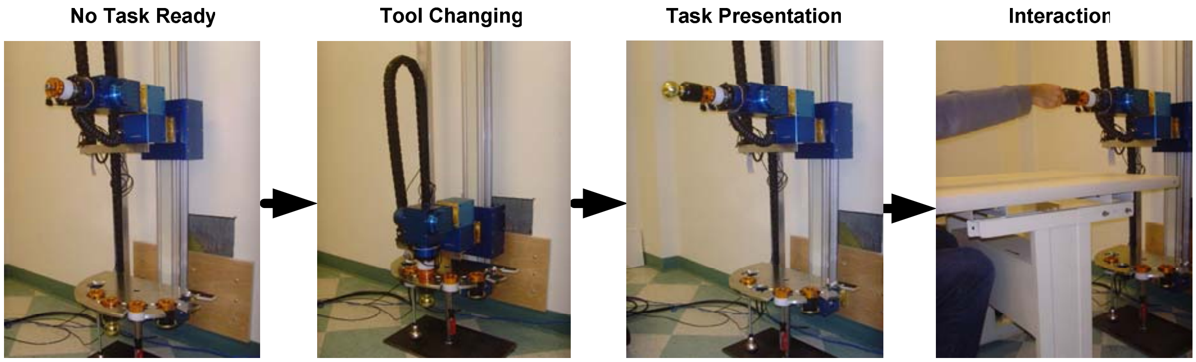


Fig. 4. The tool changing process in the current implementation of ADAPT

tasks to practice. After a task is set up for presentation to the subject, the magnetic brakes that are built into the robotic articulations are engaged on the other three DOFs during subject–robot interactions. This single DOF method simplifies kinematics and dynamics computation, and the device never falls into the wrist-singular posture, which can occur in PUMA-like manipulators [34].

Several surveillance routines are implemented to limit the maximal torque output and cap the maximum velocity of the linear and rotational motors. Watchdog routines that continuously check for failure of the position and force sensors, computer crashes, and electrical failures automatically freeze the robot by engaging the magnetic breaks in all DOFs.

An emergency red stop button can stop all robot operation and turn on magnetic brakes to disable any movement of all 4 DOF of the robot. When voltage is applied to the coil in the magnetic brake composed of a permanent magnet and spring, the magnetic field of the permanent magnet goes low and the spring opens the brake. When power is cut by the stop button, the magnetic field recovers and the brake closes. This emergency button is accessible both to the subject with his/her less affected hand and to the therapist. The magnetic brakes can also be operated when the subject is not interacting with the robot. For example, during tool changing process, the subject will be instructed to have his/her hands on an on/off switch at chair arm, which turns on the magnetic brake to stop the movement of robot if the hands are off the switch.

III. Results

We tested ADAPT01 (Version 1) with one healthy subject (we have not tested the therapeutic performance of the system at this stage). The dynamics of two tasks, doorknob turning and screw driving were simulated to demonstrate the performance of low-level controller. Then, we validated the

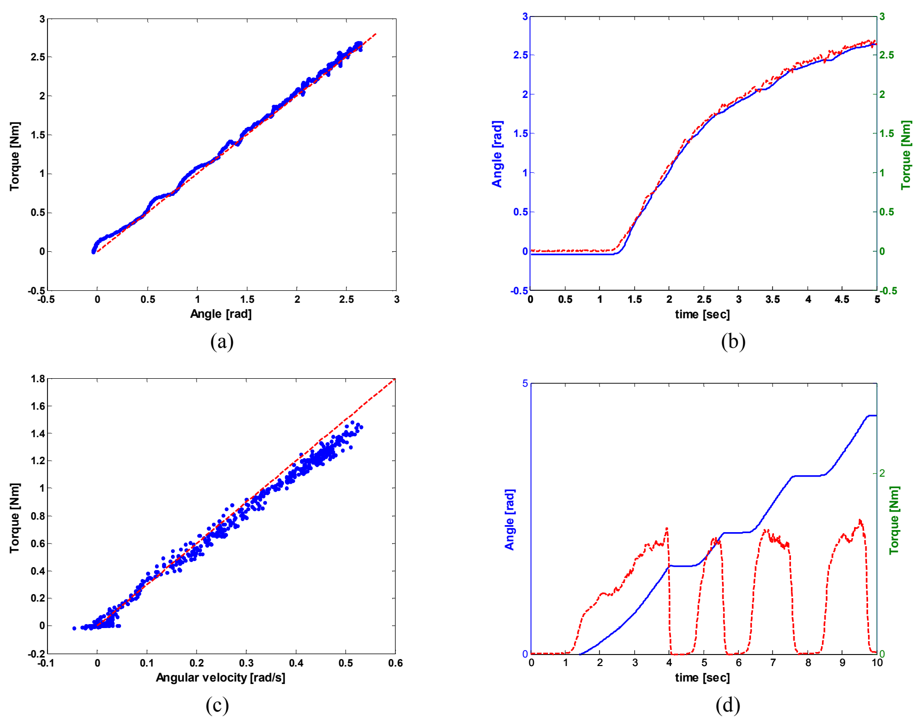


Fig. 5. Stiffness simulation (a)(b), and damping simulation (c)(d) with ADAPT. (a), torque versus angular displacement for stiffness simulation: measured data points (dot) stay near the desired stiffness (dashed, 1 Nm/rad) (b), torque (dashed) and angle trajectory (solid) of stiffness simulation for one trial. (c), torque versus angular velocity for damping simulation: measured data points (dot) stay near the desired damping (dashed, 3 Nm /(rad/sec)). (d), torque (dashed) and angle trajectory (solid) of stiffness simulation for one trial.

overall functionality of the system, including the task scheduler and the tool changing system.

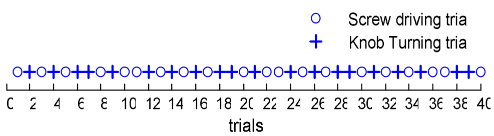
We assumed that the dynamics of the doorknob could be described as a spring with constant stiffness (Fig. 5a). The torque versus angular displacement curve (dot) shows that our low-level controller can simulate the desired stiffness (dashed line) very precisely. In Fig 5b, the smooth torque and angle trajectories for one trial of doorknob turning demonstrate the stable performance of the low-level controller. Next, we assumed that the dynamics of screw driving could be described as constant damping (Fig. 5c and 5d). The torque versus angular velocity curve shows successful damping display performance of the low-level controller.

To test the overall functionality of ADAPT01, tasks were scheduled randomly and difficulty was updated adaptively based on performance by

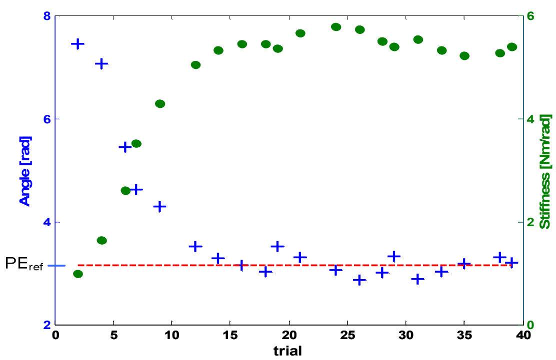
$$Dif(t+1) = Dif(t) \times (1 + \alpha\, (PE(t) - PE_{ref})) \qquad (1)$$

where $Dif(t+1)$ is next difficulty, $Dif(t)$ is current difficulty, α is learning rate, $PE(t)$ is current performance and PE_{ref} is reference performance for which the task is challenging – see [26] for details. Stiffness was controlling difficulty for the doorknob turning task and damping was controlling difficulty for the screw driving task. The range of motion, that is, how far the subject turned the knob or the screwdriver in a trial, was used as the index of performance $PE(t)$. Equation 1 ensured that $PE(t)$ was maintained near PE_{ref}, which was here arbitrarily set to be difficult enough for the healthy subject to be challenged in these tasks.

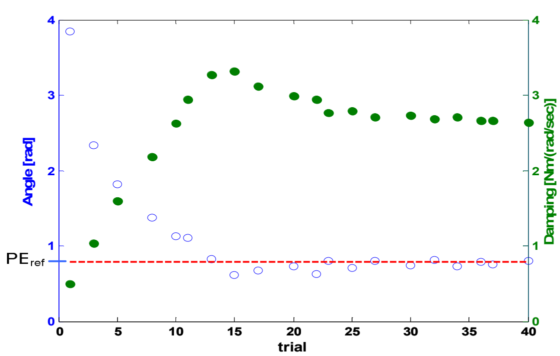
The subject practiced a total 40 trials in random schedule with 20 trials per task. Whenever a new task was selected by the scheduler, the tool changer switched to a new tool. As shown in the random schedule of Fig. 6a, the tool changer switched tools thirty two times. In this preliminary test, the system did not experience any failure in tool changing. As shown in Fig. 6, the adaptive scheduler rapidly adapted the difficulty for both tasks and the subject's performance converged to the reference performance in about 5 trials.



(a)



(b)



(c)

Fig. 6. Illustration of the function of the high-level controller. (a) Random schedule for two tasks. Details of 20 trials for each task are separately shown in (b),(c), which show how task difficulty is adapted to maintain challenging performance. (b)(c) Doorknob turning task & Screw driving task: the range of motion (cross) converges to the reference performance as stiffness (dot) or damping (dot) is increased.

IV. CONCLUSION AND FUTURE WORK

A novel robotic task practice system, ADAPT, was designed in accordance to training guidelines for stroke rehabilitation of upper extremities. Our preliminary test with a healthy subject validated the feasibility of the system. Thanks to our novel low-level controller, the robot could precisely simulate the desired task dynamics with parameters specified by the adaptive scheduler. The preliminary test further demonstrated the functionality, robustness, and safety of our adaptive scheduler combined with the tool changing system.

Although only two tasks (doorknob opening and screw driver motion) were implemented in the current version, ADAPT01, in future versions the robot will be able to present a large task repertoire by selecting from a flexible set of functional tools. These tasks, which will require active manipulation of concrete objects (e.g., turning a key or door handle, steering, opening a jar, turning a water faucet, wrist supination/pronation, etc.), will allow a large number of possible types of grasps (e.g, overhand, precision, lateral pinch, power) and individual finger motions.

In the present implementation, the tasks were modeled with simple *a priori* determined dynamics equations. In future work, in order to increase the realism of each task feel, the dynamics of the tasks will be captured, as we described in [35]. Briefly, we will first record position and force data with real tools. We will then generate the force models using RFWR. After capturing and modeling the tasks dynamics, we will replay the tasks using fixed tools mounted at the tip of the robot as described here.

While the realism of functional task is a crucial factor in task-oriented training for stroke rehabilitation, training on the robot should also be motivating. In this respect, our system can be largely improved. For instance, performing the two current tasks even with adaptive difficulty was not motivating to our subject. Moreover, the intermediate rest time was too short, causing the subject to fatigue in latter trials.

Safety concerns were strongly addressed in the initial design process of ADAPT, and the subject did not feel threatened from the robot movement. However, to guarantee the safety measures, we need long-term, intensive, and systematic tests with more subjects.

Once the current issues of the system have been solved and it has been fully tested with healthy subjects, we will begin pilot studies of efficacy and safety with patients who have had a stroke.

ACKNOWLEDGMENT

We wish to thank Matthew Sandusky for his technical support.

REFERENCES

[1] J. Desrosiers, F. Malouin, D. Bourbonnais, C. L. Richards, A. Rochette, and G. Bravo, "Arm and leg impairments and disabilities after stroke rehabilitation: relation to handicap," *Clin Rehabil*, vol. 17, pp. 666-73, 2003.

[2] W. J. Coster, S. M. Haley, P. L. Andres, L. H. Ludlow, T. L. Bond, and P. S. Ni, "Refining the conceptual basis for rehabilitation outcome measurement: personal care and instrumental activities domain," *Med Care*, vol. 42, pp. I62-72, 2004.

[3] J. H. Carr and R. B. Shepherd, *Stroke Rehabilitation - Guidelines for Exercise and Training to Optimize Motor Skill*: Butterworth-Heinemann, 2003.

[4] S. Wolf, Blanton S, Baer H, Breshears J, Butler AJ, "Repetitive Task Practice: A critical review of constraint induced therapy in stroke.," *The Neurologist*, vol. 8, pp. 325-338, 2002.

[5] G. Kwakkel, Wagenaar, RC, Twisk, JW, Lankhorst, GJ, Koetsier, JC., "Intensity of leg and arm training after primary middle-cerebral artery stroke: a randomized trial.," *Lancet*, vol. 354, pp. 191-196., 1999.

[6] E. Taub, Uswatte G, Pidikiti RD, "Contstraint-Induced (CI) movement therapy: a new family of techniques with broad appliation to physical rehabilitation - a clinical review.," *J Rehab Res Dev*, vol. 36, pp. 237-251, 1999.

[7] N. B. Lincoln, D. Willis, S. A. Philips, L. C. Juby, and P. Berman, "Comparison of rehabilitation practice on hospital wards for stroke patients," *Stroke*, vol. 27, pp. 18-23, 1996.

[8] R. A. Keith and K. S. Cowell, "Time use of stroke patients in three rehabilitation hospitals," *Soc Sci Med*, vol. 24, pp. 529-33, 1987.

[9] M. L. Aisen, H. I. Krebs, N. Hogan, F. McDowell, and B. T. Volpe, "The effect of robot-assisted therapy and rehabilitative training on motor recovery following stroke," *Arch Neurol*, vol. 54, pp. 443-6, 1997.

[10] C. G. Burgar, P. S. Lum, P. C. Shor, and H. F. Machiel Van der Loos, "Development of robots for rehabilitation therapy: the Palo Alto VA/Stanford experience," *J Rehabil Res Dev*, vol. 37, pp. 663-73, 2000.

[11] D. J. Reinkensmeyer, J. L. Emken, and S. C. Cramer, "Robotics, Motor Learning, and Neurologic Recovery," *Annu Rev Biomed Eng*, vol. 6, pp. 497-525, 2004.

[12] S. Hesse, G. Schulte-Tigges, M. Konrad, A. Bardeleben, and C. Werner, "Robot-assisted arm trainer for the passive and active practice of bilateral forearm and wrist movements in hemiparetic subjects," *Arch Phys Med Rehabil*, vol. 84, pp. 915-20, 2003.

[13] B. T. Volpe, H. I. Krebs, N. Hogan, L. Edelsteinn, C. M. Diels, and M. L. Aisen, "Robot training enhanced motor outcome in patients with stroke maintained over 3 years," *Neurology*, vol. 53, pp. 1874-6, 1999.

[14] P. Lum, D. Reinkensmeyer, R. Mahoney, W. Z. Rymer, and C. Burgar, "Robotic devices for movement therapy after stroke: current status and challenges to clinical acceptance," *Top Stroke Rehabil*, vol. 8, pp. 40-53, 2002.

[15] H. I. Krebs, M. Ferraro, S. P. Buerger, M. J. Newbery, A. Makiyama, M. Sandmann, D. Lynch, B. T. Volpe, and N. Hogan, "Rehabilitation robotics: pilot trial of a spatial extension for MIT-Manus," *J Neuroengineering Rehabil*, vol. 1, pp. 5, 2004.

[16] L. E. Kahn, W. Z. Rymer, and D. J. Reinkensmeyer, "Adaptive assistance for guided force training in chronic stroke," presented at 26th Annual International Conference of the IEEE EMBS, San Francisco, CA, 2004.

[17] L. Dipietro, M. Ferraro, J. J. Palazzolo, H. I. Krebs, B. T. Volpe, and N. Hogan, "Customized interactive robotic treatment for stroke: EMG-triggered therapy," *IEEE Trans Neural Syst Rehabil Eng*, vol. 13, pp. 325-34, 2005.

[18] J. Gordon, "Assumptions underlying physical therapy intervention: Theoretical and historical perspectives.," in *Movement science: Foundations for physical therapy in rehabilitation, 2nd edition*, C. J.H. and S. R.B., Eds. Rockville, MD.: Aspen Publishers, 2000, pp. 1-31.

[19] B. E. Fisher and K. J. Sullivan, "Activity-dependent factors affecting poststroke functional outcomes," *Top Stroke Rehabil*, vol. 8, pp. 31-44, 2001.

[20] R. P. Van Peppen, G. Kwakkel, S. Wood-Dauphinee, H. J. Hendriks, P. J. Van der Wees, and J. Dekker, "The impact of physical therapy on functional outcomes after stroke: what's the evidence," *Clin Rehabil*, vol. 18, pp. 833-62, 2004.

[21] N. A. Bayona, J. Bitensky, K. Salter, and R. Teasell, "The role of task-specific training in rehabilitation therapies," *Top Stroke Rehabil*, vol. 12, pp. 58-65, 2005.

[22] P. S. Lum, E. Taub, D. Schwandt, M. Postman, P. Hardin, and G. Uswatte, "Automated Constraint-Induced Therapy Extension (AutoCITE) for movement deficits after stroke," *J Rehabil Res Dev*, vol. 41, pp. 249-58, 2004.

[23] E. Taub, P. S. Lum, P. Hardin, V. W. Mark, and G. Uswatte, "AutoCITE: automated delivery of CI therapy with reduced effort by therapists," *Stroke*, vol. 36, pp. 1301-4, 2005.

[24] J. B. Shea and R. Morgan, "Contextual interference effects on the acquisition, retention, and transfer of motor skills.," *J. exp. psychol. [Hum. Learn.]*, vol. 3, pp. 179-187, 1979.

[25] T. D. Lee, R. A. Magill, and D. J. Weeks, " Influence of practice schedule on testing schema theory predictions in adults," *Journal of Motor Behavior*, vol. 17, pp. 283–299, 1985.

[26] Y. G. Choi, F. Qi, J. D. Gordon, and N. Schweighofer, "Performance-based adaptive schedules enhance motor learning," *Journal of Motor Behavior*, vol. in press, 2007.

[27] R. J. Nudo, B. M. Wise, F. SiFuentes, and G. W. Milliken, "Neural substrates for the effects of rehabilitative training on motor recovery after ischemic infarct," *Science*, vol. 272, pp. 1791-4, 1996.

[28] E. Plautz, Milliken GW, Nudo RJ, "Effects of repetitive motor training on movement representations in adult squirrel monkeys: role of use versus learning.," *Neurobiol Learn Mem*, vol. 74, pp. 27-55, 2000.

[29] T. D. Sanger, "Failure of motor learning for large initial errors," *Neural Comput*, vol. 16, pp. 1873-86, 2004.

[30] L. S. Vygotsky, *Mind in Society*. Cambridge, MA: Harvard University Press, 1978.

[31] E. Deci and R. M. Ryan, *Intrinsic motivation and self-determination in human behavior*. New York: Plenum, 1985.

[32] S. Schaal and C. G. Atkeson, "Constructive incremental learning from only local information," *Neural Computation*, vol. 10, pp. 2047-2084, 1998.

[33] "http://www.amtec-robotics.com/produkte_en.html."

[34] Y. Yokokohji, R. L. Hollis, and T. Kanade, "WYSIWYF display: A visual/haptic interface to virtual environment," *Presence-Teleoperators and Virtual Environments*, vol. 8, pp. 412-434, 1999.

[35] Y. G. Choi and N. Schweighofer, "Capturing and Modeling a Tool Dynamics for Adaptive Task Practice in Stroke Rehabilitation," in *Workshop on Assistive Technologies in IROS*. San Diego, USA, 2007.