2011 IEEE International Conference on Rehabilitation Robotics
Rehab Week Zurich, ETH Zurich Science City, Switzerland, June 29 - July 1, 2011

Online Learning and Adaptation of Patient Support During ADL Training

Marco Guidali, Philippe Schlink, Alexander Duschau-Wicke and Robert Riener

***Abstract*—Neurological patients with impaired upper limbs often receive arm therapy to restore or relearn lost motor functions. During the last years robotic devices were developed to assist the patient during the training. In daily life the diversity of movements is large because the human arm has many degrees of freedom and is used as a manipulandum to interact with the environment. To support a patient during the training the amount of support should be adapted in an assist-as-needed manner. We propose a method to learn the arm support needed during the training of activities of daily living (ADL) with an arm rehabilitation robot. The model learns the performance of the patient and creates an impairment space with a radial basis function network that can be used to assist the patient together with a patient-cooperative control strategy. Together with the arm robot ARMin the learning algorithm was evaluated. The results showed that the proposed model is able to learn the required arm support for different movements during ADL training.**

I. INTRODUCTION

Neuroplasticity or cortical reorganization allows the brain to adapt itself and to relearn functions lost in a brain area due to an incident (e.g. after stroke). There is evidence that therapeutic training stimulates cortical reorganization [1] and helps to restore motor function. Key factors for an effective therapy are task-specific training [2], high intensity and duration, many repetitions of the training [3]–[5], active participation of the patient [6], [7], maximal challenge for the patient [6] and execution of complex movements [8].

For stroke patients with paralyzed upper extremities robot-assisted arm rehabilitation has become a tool to address those factors. Similar to the therapist in conventional physiotherapy robotic devices can support a movement when the patient is not able to do it by himself. Additionally, the robot can measure biomechanical limb functions and give feedback about performance and progress of the therapy. Often, the devices are connected to a virtual environment, which allows a motivating and task-orientated training.

Most devices use an impedance-based control strategy to assist the patient during the training [9]. Those controllers usually have an adjustable impedance which is used to adapt the support to the patients need. One drawback of those controllers is that the support is independent of the dynamics of the human arm. In particular, patients after stroke often show abnormal movement patterns due to spasticity, pathological synergies or other pathological effects and may need different amounts of support at different joint configurations. Patient-cooperative control strategies often extend the impedance paradigm by giving the patient more freedom through no longer forcing him to a predefined trajectory. Further, to stimulate voluntary effort of the patient, assist-as-needed (AAN) strategies [10], [11] are applied to reduce the amount of support to a minimum and challenge the patient. A possible solution to learn the required arm support is to build a dynamic model of the paretic arm. Wolbrecht et al. proposed a method to learn this in real time for specific movements [12]. The impairment space of the patient is represented by a regressor matrix in task space with Gaussian radial basis functions and updated based on kinematic errors. However, this model learns the necessary arm support at different points in space for specific movements, but it does not learn a general model that can be used for a broad variety of movements, because the direction of the movement is not considered. For instance lowering the arm usually needs less effort than lifting the arm, because different muscle groups are activated to generate the force, or due to pathological synergies it might be more difficult for the patient to move his arm away from the body than towards it. Therefore, we developed a novel adaptive patient-cooperative controller that extends the idea of Wolbrecht et al. and learns the model of required arm support for different movement directions.

II. METHODS

A. Rehabilitation Robot

ARMin is an arm rehabilitation device developed at ETH Zurich, in collaboration with the University Hospital Balgrist [13], [14]. The robotic exoskeleton incorporates DC motors to support seven degrees of freedom of the arm and the hand. The range of motion (RoM) of the joints and the maximum torques provided by the actuators are listed in Table I. Mechanical end stops, redundant position sensors and many other safety features ensure a secure operation. Besides a mobilization mode and simple games, the device enables the training of activities of daily living. Training tasks are presented in an interactive virtual environment, where a broad variety of movements can be trained.

A real time system with a sampling rate of 1 kHz controls the device and communicates with the virtual world over a TCP/IP connection. The device is capable of supporting the human arm attached to it with different control strategies like impedance control or patient-cooperative control.

R. Riener is with the Sensory-Motor Systems Lab, Institute of Robotics and Intelligent Systems, ETH Zurich, Zurich, Switzerland and with the Spinal Cord Injury Center, Medical Faculty, University Hospital Balgrist, Zurich, Switzerland.
marco.guidali@mavt.ethz.ch
schlinkp@ethz.ch
riener@mavt.ethz.ch
A. Duschau-Wicke is with Hocoma AG, Volketswil, Switzerland
alexander.duschau-wicke@hocoma.com
This research was supported by the Swiss National Science Foundation (SNF) through the National Centre of Competence in Research Robotics and through grant 325200-1260621 and RANA.

TABLE I
TECHNICAL SPECIFICATION OF ARMIN III

Axis	RoM	Max. Torque
Arm elevation q_1	40°...125°	±59.5 Nm
Plane of elevation q_2	-40°...140°	±59.5 Nm
Int./ext. shoulder rotation q_3	-90°...90°	±71.4 Nm
Elbow flexion/extension q_4	0°...120°	±59.5 Nm
Forearm pro./supination q_5	-90°...90°	±7.7 Nm
Wrist flexion/extension q_6	-40°...40°	±51 Nm
Hand opening/closing q_7	0°...32°	±8 Nm

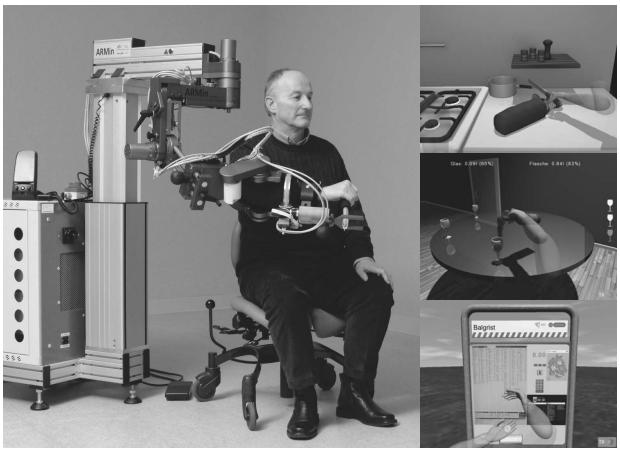


Fig. 1. ARMin III with a subject and virtual activities of daily living

B. Patient-Cooperative Control

In robot-assisted therapy patient-cooperative control strategies are applied to support the patient during the training. Many activities of daily living consist of reach and grasp tasks. Because the variability between subjects and also between movements of one subject is usually high, we should not force the patient to a general admitted reference trajectory. A control strategy where the patient is allowed to move with natural variability is needed. A frequently proposed idea is to build a virtual tunnel around a reference or mean trajectory, where the patient can move freely within the tunnel, but is restricted by the tunnel walls [15]–[19].

This path control strategy was implemented for the ARMin device using the minimal angular jerk algorithm to create a reference trajectory for the tunnel center. Inside the tunnel the patient can be supported by a force flux $\mathbf{F}_{flux}$ pointing along the path towards the target position. The amplitude of the force is adjustable by a gain k_{flux} and then mapped to the tunnel direction $\mathbf{d}_{tunnel}$ to create the force vector applied at the end-effector of the robot.

$$\mathbf{F}_{flux} = k_{flux} \frac{\mathbf{d}_{tunnel}}{|\mathbf{d}_{tunnel}|} \tag{1}$$

To calculate the path direction we use the nearest-neighbour search to find the closest point on the reference trajectory $\mathbf{p}_{ref}$.

$$R : ||\mathbf{p}_{ref}(R) - \mathbf{p}_{act}||^2 := min \tag{2}$$

Here, $\mathbf{p}_{act}$ is the actual end-effector position in space. By deriving the trajectory at this point $\mathbf{p}_{ref}$ we get the direction of the path $\mathbf{d}_{tunnel}$.

$$\mathbf{d}_{tunnel} = \begin{pmatrix} d_{tunnel_x} \\ d_{tunnel_y} \\ d_{tunnel_z} \end{pmatrix} = \frac{d}{dR}\mathbf{p}_{ref}(R) \tag{3}$$

In order to assist the patient only as much as needed, the gain of the flux k_{flux} needs to be adapted according to his impairment. In clinical practice up to now the therapists have adapted this gain, so that the patient can complete the exercise. Additionally, this gain is increased if the patient is slower than a minimal desired movement represented by $\mathbf{p}_{minDes}$.

$$k_{flux} = \begin{cases} k_{base} \cdot (1 + \frac{||\mathbf{p}_{minDes} - \mathbf{p}_{ref}||}{2}) & \text{if } \mathbf{p}_{minDes} > \mathbf{p}_{ref} \\ k_{base} & \text{otherwise.} \end{cases} \tag{4}$$

Due to the broad diversity of range of motion and movement direction during ADL's it is quite obvious that this gain has to be quite variable to assist the patient only as needed. Therefore, a model of the needed arm support would be beneficial in order to adapt this gain with prior knowledge.

C. Patient Model

The impairment space of persons after stroke often varies widely between subjects. Many factors like pathological synergies, spastisticy or muscle weakness can influence their arm dynamics and their ability to perform movements. Therefore, a mathematical representation of the arm dynamics is very difficult to achieve. A possible solution is to learn the model of the required arm support online [12]. This model learns the necessary arm support at different points in space for specific movements. During ADL training very different movements within the workspace are executed. To learn the support needed to perform ADL tasks with the path control strategy the model of Wolbrecht et al. was extended by direction information, which is basically the spatial velocity discretized into positive or negative values. In other words, the model learns for all positions the support needed to move in one of the six directions, positive x-direction, negative x-direction, positive y-direction, negative y-direction, positive z-direction, negative z-direction. A 2-D version of this model is illustrated in Fig. 2.

The gain k_{flux} of the path controller can now be replaced by a gain matrix.

$$\mathbf{F}_{flux} = \begin{pmatrix} F_x & 0 & 0 \\ 0 & F_y & 0 \\ 0 & 0 & F_z \end{pmatrix} \cdot \frac{\mathbf{d}_{tunnel}}{|\mathbf{d}_{tunnel}|} \tag{5}$$

The forces F_i are dependent on the actual end-effector position in space and the direction of the path at this point.

$$F_i = f(x, y, z, d_{tunnel_i}) \tag{6}$$

$$F_i = \begin{cases} F_{i^+}(x,y,z) & \text{if } d_{tunnel_i} > 0 \\ F_{i^-}(x,y,z) & \text{if } d_{tunnel_i} < 0 \end{cases} \tag{7}$$

In the implemented 3-D version the grid points are equally spaced with 10 cm intervals. The total covered workspace is

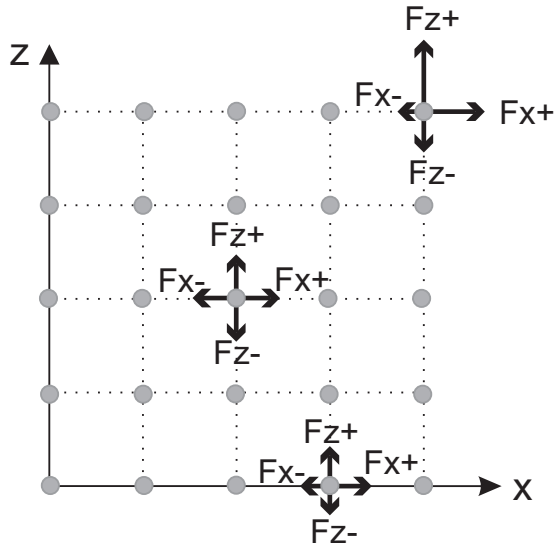


Fig. 2. Simplified model of required arm support in 2-D. The workspace is divided into grid points with four radial basis functions at the center of all grid points. Each function learns the support needed to move into one of the four directions. For three grid points the exemplary values are shown as vectors.

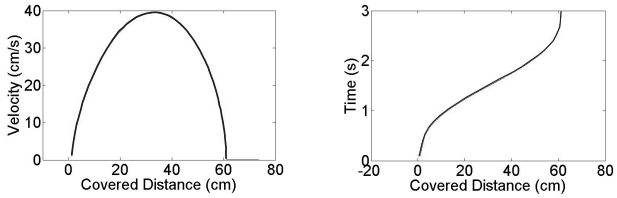
slightly larger than the possible workspace of ARMin to ensure a safe condition at the border.

D. Online Learning

Learning the above described model of required patient support means that criteria for desired behavior need to be defined. Besides the task performance we can also measure kinematic data with the position sensors of ARMin. As we allow some spatial freedom during the movement with the path control strategy, we can not learn from a position error without violating our concept of allowing variability inside the tunnel. Therefore, the kinematic measures velocity and time are used to rate the performance and to update the patient model accordingly. The update process itself is based on the iterative learning concept and underlies the following equation,

$$F_i^k = (1-\alpha)\cdot F_i^{k-1} + \gamma \cdot E \tag{8}$$

where F_i^k is the support needed at iteration k, E denotes the error to minimize, α is called forgetting factor and γ the learning rate. With the forgetting rate we ensure that we always try to challenge the patient by decreasing the support at each iteration. The learning rate defines how fast the model adapts to errors. We defined a physiological bell-shaped trajectory as desired velocity profile [20]. The desired movement time can be estimated, for instance, with Fitt's Law if the distance to the target, the size of the target and individual parameters are known [21]. For simplification we used a fixed movement time of 3 s, which was determined experimentally by analyzing trajectories of subjects training ADL's with ARMin. Using a constant movement time accounts for the fact that shorter movements have a lower average speed than longer movements. Together with the velocity profile the desired kinematics can be calculated as a function of distance along the path *CD* (see Fig. 3).



(a) Desired velocity profile (b) Desired time profile

Fig. 3. Desired velocity as function of covered distance along the path (a) and desired time (b) as function of distance.

With those two measures the kinematic error E can be calculated with,

$$E = \frac{||\mathbf{v}_{des}(CD) - \mathbf{v}_{act}||}{||\mathbf{v}_{desPeak}||} + \eta \cdot \frac{t_{elapsed} - t_{des}(CD)}{t_{elapsed}} \tag{9}$$

where η denotes the relative learning factor to change the impact of the two error terms and $t_{elapsed}$ the elapsed time since start of the movement. Both reference time and reference velocity are scaled to guarantee that the error stays between 0 and 1. Due to the scaling of the time error with the elapsed time, the error can grow fast in the beginning of a movement in the case where the patient can not initiate the movement. At the end, the time error guarantees that the support is increased until the patient reaches the target. On the other hand the velocity error has more influence during the movement and is used to keep up the physiological velocity profile.

Using the radial basis functions to learn the support ensures that grid points are updated regularly and no discontinuity in support occurs. To limit computational time, only the two closest grid points to the end-effector position in each direction are chosen for the update and actual support calculation. Having six spatial directions this leads to $2^6 = 64$ grid points. Considering that the grid-spacing was set to 10 cm, these 64 grid points built a cube of 40 cm side length. The Radial basis function is expressed by,

$$G = \exp\left(-\frac{||\mathbf{p}_{act} - \mathbf{p}_{grid}||^2}{2\sigma^2}\right) \tag{10}$$

where $\mathbf{p}_{act}$ denotes the actual end-effector position in space and $\mathbf{p}_{grid}$ a grid point. The update process of one grid point can now be expressed as

$$F_i^k = (1-\alpha\cdot G)\cdot F_i^{k-1} + G\cdot\gamma_i \cdot \frac{d_{tunnel_i}}{|\mathbf{d}_{tunnel}|}\cdot E. \tag{11}$$

E. Experimental Validation

In an experimental setup the online learning and adaptation of arm support was evaluated with ten healthy subjects (N=10) and two chronic stroke subjects (N=2). As benchmarks for our controller we defined the following. First, the controller should be versatile. This means different movements should be learned by the model. Secondly, the controller should only

TABLE II
MODEL PARAMETERS

Name	Symbol	Value
Forgetting factor	α	$\frac{1}{8}$
Learning rate	$\gamma = \begin{pmatrix} \gamma_x \\ \gamma_y \\ \gamma_z \end{pmatrix}$	$1.6 \cdot 10^{-4} \cdot \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} [N]$
Relative learning rate	η	1.2

assist as much as needed and reduce the amount of support if the subject increases his activity. And thirdly, the controller should allow some error to account for variability in the movement trajectory and timing. Additionally, the new support adaption was compared to the first simple mechanism (see eq. 4), where the support was adapted with only a time constraint, but forgotten once the target was reached.

Two different movements were chosen to test the controllers, one on the horizontal plane from left to right and one diagonally upwards. The movement had to be performed under the conditions active or passive. Active means that the subject tried to do the movement with his own effort. Under the passive condition the subject could relax and let the robot move his arm. The movement back from the target position to the start position was also supported by the assistive controller and was always under the passive condition.

Three different condition sequences where used.

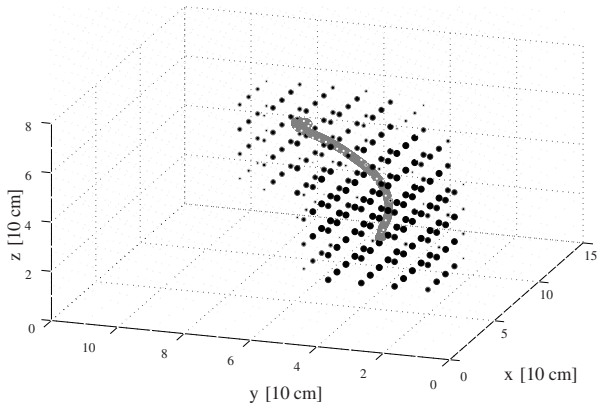
- Four passive movements followed by four active movements (4P4A).
- Active movement with a short pause in the middle of the movement (ASA).
- Active only during the first half of the movement then transition to passive (ASP).

In all experiments the same model parameters were used (see Table II). The parameters were experimental determined with previous tests of the algorithm.

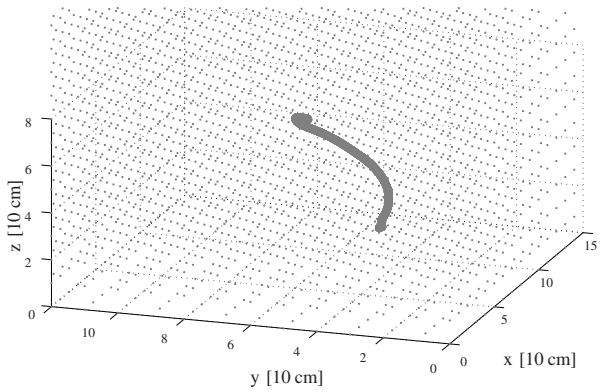
III. RESULTS

A. Healthy subjects

In the experiment conducted the developed algorithm to learn and adapt the arm support was tested. First, we validated how versatile the adaptive controller is in learning the needed arm support in different movement directions. Four passive diagonally up and down movements were analyzed to test if the controller assist both movement the same way. The learned model of the support in z-direction was extracted after the movement and can be seen exemplary for one subject in Fig. 4. As expected lifting the arm needed much more support than lowering the arm back to the start position. Comparing the peak speeds of the passive movements between the learning algorithm and the non-learning algorithm we can see that if we learn the support we achieve same values for peak speed in up and down direction, but if we don't learn the down peak velocity is almost twice as high, because the controller applies too much support.



(a) Support model in positive z-direction



(b) Support model in negative z-direction

Fig. 4. Representation of the arm support model in positive z-axis after one passive movement. The bold gray line represents the trajectory in space. The circles at the grid points show the amount of support needed at this location. The darker and bigger the higher the force learned.

TABLE III
PEAK SPEED OF DIAGONALLY UP AND DOWN MOVEMENTS

direction	Learning algorithm	Non-learning algorithm
up	29.5 (5.1 SD) [cm/s]	24.4 (4.1 SD) [cm/s]
down	30.1(4.7SD) [cm/s]	47.8 (11.2 SD) [cm/s]

The idea of a patient-cooperative controller is to assist only as needed. In the first sequence of conditions four passive movements were followed by four active (4P4A). The learned arm support should decrease rapidly if the subject changes his activity. To compare active and passive behavior the integrated force for the movement was calculated for the four passive and the subsequent active movements (see Fig. 5).

The third requirement was that we want to allow some error. With the path control strategy and the learning criteria we allow already some spatial variability of the movement, but we also want to allow some freedom in timing. To test the behavior of the developed adaptive controller a second and a third movement sequence was used. During the second sequence the subjects had to perform the whole movement actively, but with a short stop at the half distance to the target

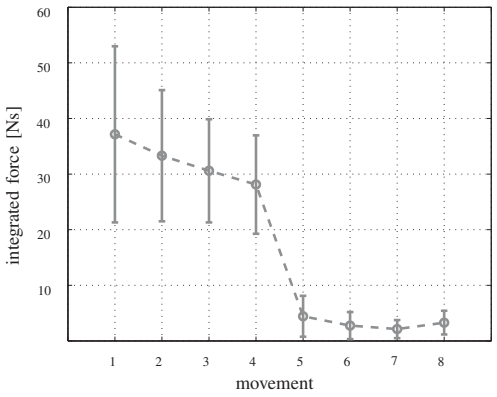
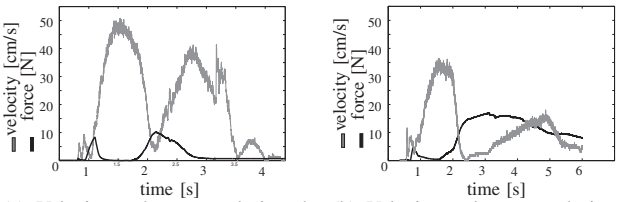


Fig. 5. Mean values of the required integrated force to complete the movement for the four passive movements (1-4) and the four active movements (5-8) of the ten healthy subjects.

(see Fig. 6(a)). The third sequence was similar except that they had to stay passive after the stop in the middle of the trajectory (see Fig. 6(b)).



(a) Velocity and support during the ASA sequence

(b) Velocity and support during the ASP sequence

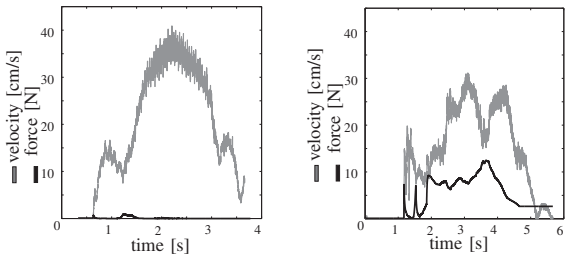
Fig. 6. Behavior of the adaptive controller if the subject stopped in the middle of the movement and then continued to do the movement a) or staying passive after the stop b). When the subjects started to decrease velocity the applied support increased until the subjects started to move again.

All subjects were able to stop and start again without assistance from the controller. Although the controller expected one movement, the subjects could do two sub-movements still showing the desired physiological velocity profile.

B. Stroke subjects

Two chronic stroke subjects were recruited to test the controller and performed the same experiments as the healthy subjects. The learned model allowed the patient to be assisted only as needed. One example can be seen in Fig. 7, where the patient did not need support for the horizontal movement but for the vertical. All movements showed a very physiological velocity profile. The calculated integrated force for the first sequence of four passive movements followed by four active movements (4P4A) can be seen in Fig. 8.

Additionally, the patients had to complete a whole ADL task as in therapy consisting of sixteen different movements. Both subjects were able to finish the ADL task with assistance from the device.



(a) Horizontal movement

(b) Vertical movement

Fig. 7. Support force and velocity profiles of a stroke patient trying to actively perform the movements.

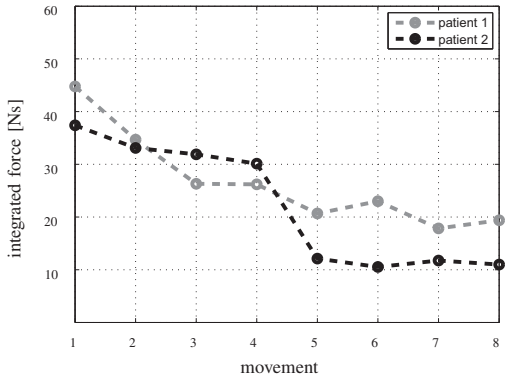


Fig. 8. Mean values of the required integrated force to complete the movement for the four passive movements (1-4) and the four active movements (5-8) of two chronic stroke patients.

IV. DISCUSSION

A new method to learn the necessary arm support online during robot-assisted therapy has been developed. The learned model was used to adapt the assistance provided by a patient-cooperative control strategy. The results showed that the controller is able to adapt the arm support properly. Due to the online learning the controller can react quickly to the activity of the subject. This is important, because the motivation of the patient can decrease fast if he has to wait at a point too long for more assistance to complete the movement. On the other hand, this brings also some drawbacks, because the patient could just wait for the support instead of participating actively. Although the patients did not show this effect, we can use motivating tasks and feedback about the current arm support to reduce this possible drawback. Learning the arm model by errors in spatial velocity and timing allowed the subjects to choose their own movement trajectories within the tunnel without getting ”punished” by the adaptive controller. During the active movements of the healthy subjects the integrated arm support never dropped to zero. One reason is that because only gravity and Coulomb friction of ARMin can be compensated with the position sensors. Inertia, static friction and other effects can still influence the system and are also learned by the arm model. The results from the 4P4A condition showed that the integrated force for moving the subjects arm decreases after the first passive movements.

Because the model of required arm support was initialized with zero at the beginning of the experiment, the controller had to learn the model of the arm to find the optimal support. The experiment with the two chronic stroke subjects showed that the adaptive controller also learns the model of patients. The integrated force required to move their arm in the condition passive was similar to the healthy subjects. When the patient tried to actively perform the movement the model reduced the support. In the active condition both patients required more support to do the movement than healthy subjects. Both patients could finish one complete ADL task consisting of 16 different movements. For this experiment not a single parameter of the controller had to be adapted manually, e.g. by the therapist or experimenter. This brings a great advantage for clinical practice, because the adaptation of parameters can be time consuming. Further experiments with more stroke subjects need to be conducted to evaluate wheter the adaptive controller is capable of learning all kinds of arm impairment.

V. CONCLUSION

This paper proposes an adaptive mechanism to learn the required arm support during robot-assisted therapy. The model learns online only from kinematic errors and can be used without prior knowledge about the patient and does not require manual adjustments. We combined the algorithm with a patient-cooperative control strategy and the ARMin device. The results with healthy subjects and stroke patients showed that the model is able to capture the necessary arm support in an assist-as-needed manner, while still allowing spatial and some temporal freedom.

VI. ACKNOWLEDGMENTS

The authors would like to thank the subjects who participated in the evaluation of this work and the therapists Anja Kollmar and Sabine Schoelch for their help.

REFERENCES

[1] J. Liepert, S. Graef, I. Uhde, O. Leidner, and C. Weiller. Training-induced changes of motor cortex representations in stroke patients. *Acta neurologica scandinavica*, 101(5):321–326, 2000.

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

[3] G. Kwakkel, R.C. Wagenaar, T.W. Koelman, G.J. Lankhorst, and J.C. Koetsier. Effects of intensity of rehabilitation after stroke. a research synthesis. *Stroke*, 28:1550–1556, 1997.

[4] A. Sunderland, D.J. Tinson, E.L. Bradley, D. Fletcher, H.R. Langton, and D.T. Wade. Enhanced physical therapy improves recovery of arm function after stroke. A randomised clinical trial. *Neurol Neursurg Psychiatry*, 55:530–535, 1992.

[5] C. Butefisch, H. Hummelsheim, P. Denzler, and K.H. Mauritz. Repetitive training of isolated movements improves the outcome of motor rehabilitation of the centrally paretic hand. *J Neurol Sci*, 130:59–68, 1995.

[6] N. Hogan, H.I. Krebs, B. Rohrer, J.J. Palazzolo, L. Dipietro, S.E. Fasoli ans J. Stein, R. Hughes, W.R. Frontera, D. Lynch, and B. Volpe. Motions or muscles? Some behavioral factors underlying robotic assistance of motor recovery. *Journal of rehabilitation research and development*, 43(5):605–618, 2006.

[7] G. Nelles, W. Jentzen, M. Jueptner, S. Muller, and HC Diener. Arm training induced brain plasticity in stroke studied with serail position emission tomography. *Neuroimage*, 13(6):1146–1154, 1999.

[8] M. Roosink and I. Zijdewind. Corticospinal excitability during observation and imagery of simple and complex hand tasks: Implications for motor rehabilitation. *Behavioural brain research*, 213(1):35–41, 2010.

[9] L. Marchal-Crespo and DJ Reinkensmeyer. Review of control strategies for robotic movement training after neurologic injury. *J. of NeuroEngineering and Rehabilitation*, 6(20), 2009.

[10] J. L. Emken, J. E. Bobrow, and D. J. Reinkensmeyer. Robotic movement training as an optimization problem: designing a controller that assists only as needed. *In Proc. IEEE 9th Int. Conf. Rehabil. Robot*, pages 307–312, 2005.

[11] D. J. Reinkensmeyer, D. Aoyagi, J. L. Emken, J. A. Galvez, W. Ichinose, G. Kerdanyan, S. Maneekobkunwong, K. Minakata, J. A. Nessler, R. Weber, R. R. Roy, R. de Leon, J. E. Bobrow, S. J. Harkema, and V. R. Edgerton. Tools for understanding and optimizing robotic gait training. *J. Rehabil. Res. Dev*, 43(5):657670, 2006.

[12] E.T. Wolbrecht, V. Chan, D.J. Reinkensmeyer, and J.E. Bobrow. Optimizing compliant, model-based robotic assistance to promote neurorehabilitation. *Transactions on neural systems and rehabilitation engineering*, 16(3):286–297, 2008.

[13] T. Nef, M. Mihelj, and R. Riener. ARMin: a robot for patient-cooperative arm therapy. *Medical and Biological Engineering and Computing*, 45(9):887–900, 2007.

[14] T. Nef, M. Guidali, and R. Riener. ARMin III - arm therapy exoskeleton with an ergonomic shoulder actuation. *Applied Bionics and Biomechanics*, 6(2):127–142, 2009.

[15] P. Lum, C.G. Burgar, P.C. Shor, M. Majmundar, and M. van der Loos. Evidence for improved muscle activation patterns after retraining of reaching movements with the mime robotic system in subjects with post-stroke hemiparesis. *IEEE Trans. Neural Systems and Rehabilitation*, 12(2), 2004.

[16] A. Duschau-Wicke, J. v. Zitzewitz, A. Caprez, L. Lnenberger, and R. Riener. Path control: A method for patient-cooperative robot-aided gait rehabilitation. *Trans. on Neural Systems and Rehabilitation Engineering*, 18(1):38–48, 2009.

[17] L. L. Cai, A. J. Fong, C. K. Otoshi, Y. Liang, J. W. Burdick, R. R. Roy, and V. R. Edgerton. Implications of assist-as-needed robotic step training after a complete spinal cord injury on intrinsic strategies of motor learning. *J. Neuroscience*, 26(41):564568, 2006.

[18] H. Krebs, J. J. Palazzolo, L. Dipietro, M. Ferraro, J. Krol, K. Rannekleiv, Volpe, and N. Hogan. Rehabilitation robotics: Performance-based progressive robot-assisted therapy. *Autonomous Robots*, 15(1):720, 2003.

[19] S. K. Banala, S. K. Agrawal, , and J. P. Scholz. 'active leg exoskeleton (alex) for gait rehabilitation of motor-impaired patients. *in Proc.IEEE 10th Int. Conf. Rehabil. Robot*, page 401407, 2007.

[20] K. Yamanaka, Y. Wada, and M. Kawato. Quantitative examinations for human arm trajectory planning in three-dimensional space. *Wiley Periodicals, Inc. Syst Comp Jpn*, 34(7):43–54, 2003.

[21] Atsuo Murata and Hirokazu Iwase. Extending fitts' law to a three-dimensional pointing task. *Human Movement Science*, 20(6):791 – 805, 2001.