Assist-as-needed path control for the PASCAL rehabilitation robot

Urs Keller[1,2,3], Georg Rauter[1] and Robert Riener[1,2]

***Abstract*—Adults and children with neurological disorders often require rehabilitation therapy to improve their arm motor functions. Complementary to conventional therapy, robotic therapy can be applied. Such robots should support arm movements while assisting only as much as needed to ensure an active participation of the patient. Different control strategies are known to provide arm support to the patient. The path controller is a strategy that helps the patient's arm to stay close to a given path while allowing for temporal and spatial freedom. In this paper, an assist-as-needed path controller is presented that is implemented in the end-effector-based robot PASCAL, which was designed for children with cerebral palsy. The new control approach is a combination of an existing path controller with additional speed restrictions to support, when the arm speed is too slow, and to resist, when the speed is too fast. Furthermore, a target position gain scheduling is introduced in order to reach a target position with a predefined precision as well as an adaptable direction-dependent supportive flux that supports along the path. These path control features were preliminarily tested with a healthy adult volunteer in different conditions. The presented controller covers the range from a completely passive user, who needs full support to an actively performed movement that needs no assistance. In close future, the controller is planned to be used to enable reaching in children as well as in adults and help to increase the intensity of the rehabilitation therapy by assisting the hand movement and by provoking an active participation.**

I. INTRODUCTION

After a stroke, a lesion of the spinal cord or another neurological disorder, people are often restricted in the use of their affected arm during daily life. These patients usually undergo an intensive, task-specific and repetitive rehabilitation training in order to regain or acquire motor functions of the arm. During this training, an active participation is highly relevant to trigger neuroplasticity and to significantly improve the therapy outcome in adults [1] as well as in children [2].
Complementary to conventional interventions, robots are more and more commonly used to support frequent and long training. They can compensate for the arm weight and support the arm during movements in space. In order to motivate the patient's active contribution to these movements, it is important that the robot is not completely guiding the arm but cooperatively assisting the arm as much as needed. An assist-as-needed (AAN) controller can help the patient to complete a movement and allows for position and timing errors, since making errors is known to drive motor learning [3], while fixed guidance rather hampers the learning process [4]. As a further enhancement, game-like scenarios with audiovisual feedback are used to instruct and challenge the patient and to motivate the patient to remain active during the training.
Different support strategies have been described in the literature to assist the patient. A detailed review on haptic control strategies, especially for training of complex movements, can be found in Sigrist et al. 2012 [5]. First, impedance control approaches aim to increase the compliance of the robot in order to allow the patient to deviate from a predefined trajectory [6]. But to promote an active contribution of the patient to the arm movement, controllers were introduced that try to minimize the robot support and allow for spatial freedom while providing sufficient support that the patient can fulfill the desired task, e.g. controllers following an AAN approach [7]. However, these control strategies do usually not allow for temporal freedom. A solution to this problem is the use of a virtual tunnel which is rendered around a given path in space, rather than using a predefined trajectory [8], [9]. This allows freedom in terms of timing and possible deviation from the path. This kind of controller is often referred to as path controller and has been used for supporting periodic movements in task space [10], [11] and in joint space [12], [13] as well as for point-to-point movements of the arm [8], [14].
The robot used in this paper is PASCAL (Pediatric Arm Support robot for Combined Arm and Leg training, [15]) (Fig. 1). PASCAL is an end-effector-based robot to support arm movements and can be combined with the commercial gait orthosis Lokomat® (Hocoma, Switzerland) [16] to train combined movements of arm and legs. In this paper, however, PASCAL is used as a decoupled system for (re-)habilitation therapy and with focus on the upper extremity to assist children with cerebral palsy (CP) or other neurological lesions and with mild spasticity. PASCAL is primarily used to support arm reaching movements in space. For this, the robot is interfaced with a display that shows the PASCAL end-effector position

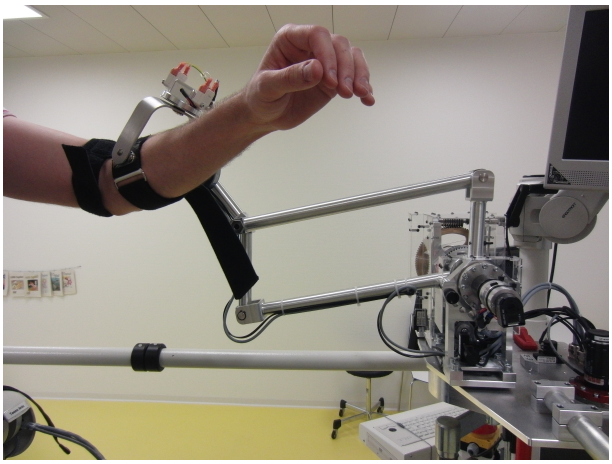

Fig. 1. PASCAL robot fixed to the combined center of mass of the arm.

This work was supported by the Fondation Gaydoul and the Mäxi Foundation to the Rehabilitation Center for Children and Juveniles, Affoltern a. Albis, Switzerland.

[1]Sensory-Motor Systems Lab, ETH Zurich, Switzerland
[2]Medical Faculty, University Hospital Balgrist, University of Zurich, Switzerland
[3]Rehabilitation Center, University Children's Hospital Zurich, Affoltern a. A., Switzerland
urs.keller@hest.ethz.ch
georg.rauter@hest.ethz.ch
robert.riener@hest.ethz.ch

978-1-4673-6024-1/13/$31.00 ©2013 IEEE

as well as the target position that has to be reached. The task for the subject is to successively reach for the different targets. During these point-to-point movements, the desired path to a given target position is precalculated according to the minimum jerk paradigm as described in the literature [17]. To allow for spatial and temporal freedom, the path control concept was chosen and adapted to the PASCAL.
In this paper, we describe how the path controller approach can be combined with minimum and maximum speed restrictions and a scheduled control gain to reach a target with desired precision. Furthermore, a direction-dependent assisting force, called flux, along the path is provided that is continuously adapted to the movement direction to support the patient arm during the exercise. This AAN path controller was implemented in PASCAL and tested with a healthy adult volunteer for preliminary evaluation of the path controller and its extended features.

II. Method

A. PASCAL robot

PASCAL is an end-effector-based robot with three active degrees of freedom (DoF) (Fig. 2) [15]. The robot is fixed to the combined center of mass (CCM) of the human arm (as suggested by [18]) by means of a cuff (Fig. 1). However, for the tests performed in this paper the robot end effector was attached to the hand by replacing the cuff with an appropriate handle.

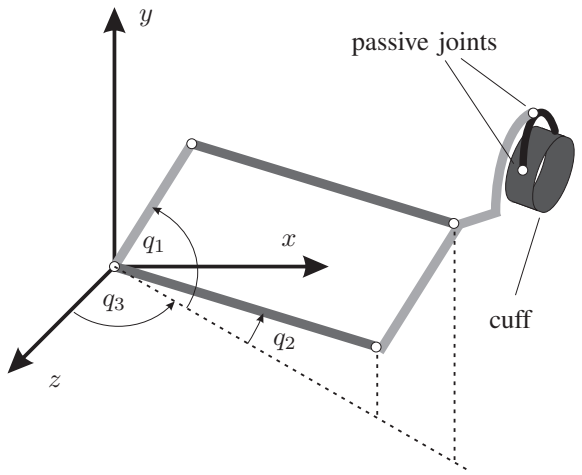

Fig. 2. Kinematic structure of the PASCAL robot. The angles q_1, q_2 and q_3 represent actuated joints.

A parallel kinematic structure and a motor rotating it around the vertical axis allows movement of the end effector in three translational degrees of freedom. The cuff itself contains two more passive DoF to ensure that only interaction forces are present and no torques are applied to the arm. As the robot is used in close interaction with humans, different mechanical, software and electrical safety measures guarantee the overall safety of the patient.

B. Support features of the AAN path controller

The PASCAL robot is used to support arm movements along a path in space. However, the patient may not be able to move actively along the given path. Therefore, the controller presented in this paper has four main features to help the patient's arm to move along the path:

- Virtual tunnel to stay close to the desired path
- Minimum and maximum speed restrictions
- Direction-dependent supportive flux along the path
- Gain-scheduling control to ensure that the target position is reached

These features will be discussed in the following subsections.

C. Virtual tunnel

In order to help the patient to stay close to a prescribed path in space, a virtual tunnel is rendered from the given start position $\boldsymbol{p}_{start}$ to the desired target position $\boldsymbol{p}_{target}$ of the path (Fig. 3)[1] [13]. If the patient deviates from the given path, the tunnel wall will apply a force in the direction of the path and, therefore, push the patient's arm to stay close to the path. In close vicinity to the path, there is a dead band with constant radius R_w, where no tunnel forces are applied. This allows the patient to choose his individual trajectory without being assisted (patients with little or no remaining functions can be supported by the additional strategies introduced later). If the actual position $\boldsymbol{p}_{act}$ is outside this radius, a tunnel force $\boldsymbol{f}_{t,1}$ is applied to the end effector at position $\boldsymbol{p}_{act,1}$ (Fig. 3) that points in the direction of the shortest distance to the path (Euclidean distance). Therefore, the patient's arm is forced to move towards the path.

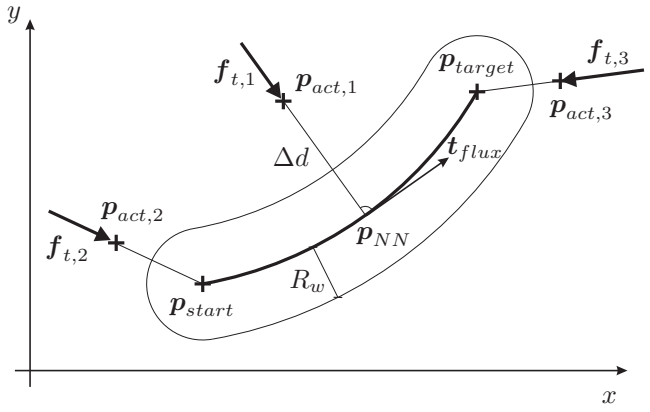

Fig. 3. Concept of the virtual tunnel used in the path controller. Simplified to a planar case for better understanding.

The resulting force field for the tunnel can either be calculated by real-time differentiation of a potential field calculated beforehand (e.g. by using radial basis functions to build up a potential field [19]) or by directly calculating the force field in a given position. The approach presented in this paper requires that the force field can be changed during a movement (as described later by the changing path using minimum and maximum speed boundaries) and, therefore, the tunnel force

[1]The notation that is used throughout this paper, is $\boldsymbol{p} = (p_x, p_y, p_z)^T$ for the end-effector position and $\boldsymbol{q} = (q_1, q_2, q_3)^T$ for angles of the three robot axis and $\boldsymbol{f} = (f_x, f_y, f_z)^T$ being the force applied at the end effector.

applied to the arm is calculated online by using the following formula [14]:

$$\boldsymbol{f}_t = \begin{cases} \mathbf{K} \cdot (\Delta d - R_w) \cdot \frac{\boldsymbol{p}_{act} - \boldsymbol{p}_{NN}}{\Delta d}, & \text{if} \quad \Delta d > R_w \\ 0, & \text{else} \end{cases} \tag{1}$$

where $\Delta d = |\boldsymbol{p}_{act} - \boldsymbol{p}_{NN}|$, $\boldsymbol{p}_{NN}$ is the nearest neighbor position on the path and $\mathbf{K}$ is a diagonal matrix with the stiffness values for the tunnel wall. Both, $\mathbf{K}$ and R_w, can be adapted during the therapy to adjust the controller to the needs of the patient. It can be seen from (1) that the tunnel force increases linearly with the distance to $\boldsymbol{p}_{act}$ from the path. When the actual position is lying rearward to the start position ($\boldsymbol{p}_{act,2}$, Fig. 3), $\boldsymbol{p}_{NN}$ is set to coincide with the start position. As a consequence, the tunnel forces $\boldsymbol{f}_{t,2}$ point to the start position. In the case that the actual position is ahead of the target position ($\boldsymbol{p}_{act,3}$, Fig. 3), $\boldsymbol{p}_{NN}$ is set to coincide with the target position and the tunnel forces point to the target.
The force calculation according to (1) will only produce a smooth force field over the whole workspace for end-effector positions $\boldsymbol{p}_{act}$ that have only one single nearest neighbor. This is only true if the local radius of the path (r_{path}) is longer than the distance between the actual position and its nearest neighbor position and, therefore, the necessary condition for the curvature κ of the path is:

$$\kappa < \frac{1}{|\boldsymbol{p}_{act} - \boldsymbol{p}_{NN}|} \quad \forall \ \boldsymbol{p}_{act} \in W,$$

where W is the set of the points that lie within the reachable workspace. As a consequence, it has to be ensured, that only reference paths are chosen that fulfill this condition for all possible end-effector positions within the workspace.

D. Minimum and maximum speed restrictions

A patient may not be able to move along the tunnel trajectory on his own. Here, the robot should adapt its support and start assisting along the path. For this purpose, a minimum speed reference trajectory is defined from the start to the target position. In case that the patient is slower than this minimum speed reference an additional force will help to move along the path. The idea to have a minimum speed was already introduced in 2003 by Krebs and his group [8] and was applied in the ARMin robot [14]. Based on this, a minimum speed trajectory $\boldsymbol{p}_{min}(t)$ is defined using a minimum jerk profile. However, for very slow movements, the speed trajectory should be changed to a constant velocity profile. This minimum speed trajectory can be considered as a moving wall in the back of the virtual tunnel, i.e. if the patient is not moving or his speed is too slow, the back of the tunnel will push the end effector towards the target position.
Another issue is that a patient can also move too fast, e.g. if a start position is above the target position. In this case the arm may fall down inside the tunnel due to the acting gravity, unless there is a braking force. In our project framework, this problem was solved by introducing an additional maximum speed trajectory $\boldsymbol{p}_{max}(t)$ analogous to the minimum speed trajectory. This maximum speed trajectory defines the momentary end of the virtual tunnel, i.e., when the arm is moving too fast, the moving front end of the tunnel constrains the arm speed according to the maximum speed profile. Both speed boundaries superimposed lead to a virtual tunnel which is lengthening and contracting while the patient moves from the start to the target position. This is illustrated in Fig. 4.

E. Direction-dependent supportive flux

The minimum speed trajectory was introduced to help patients that cannot move their arm along the path. This approach can even be used for patients with severely affected arm functions or a completely passive arm. But patients with mildly affected arms may only need a little support to perform the movement on their own. Therefore, the concept of flux is introduced. The flux is an adaptable force that supports the patient along the virtual tunnel [14], [13], [20].
This supportive flux $\boldsymbol{f}_{flux}$ can be calculated by the following formula:

$$\boldsymbol{f}_{flux} = k_{flux} \cdot \boldsymbol{t}_{flux} \tag{2}$$

where k_{flux} denotes the flux gain and can be adapted by the therapist and $\boldsymbol{t}_{flux}$ is the direction of the flux which is parallel to the tangent of the path (Fig. 3). Generally, the total support applied by the path controller at the end effector is the

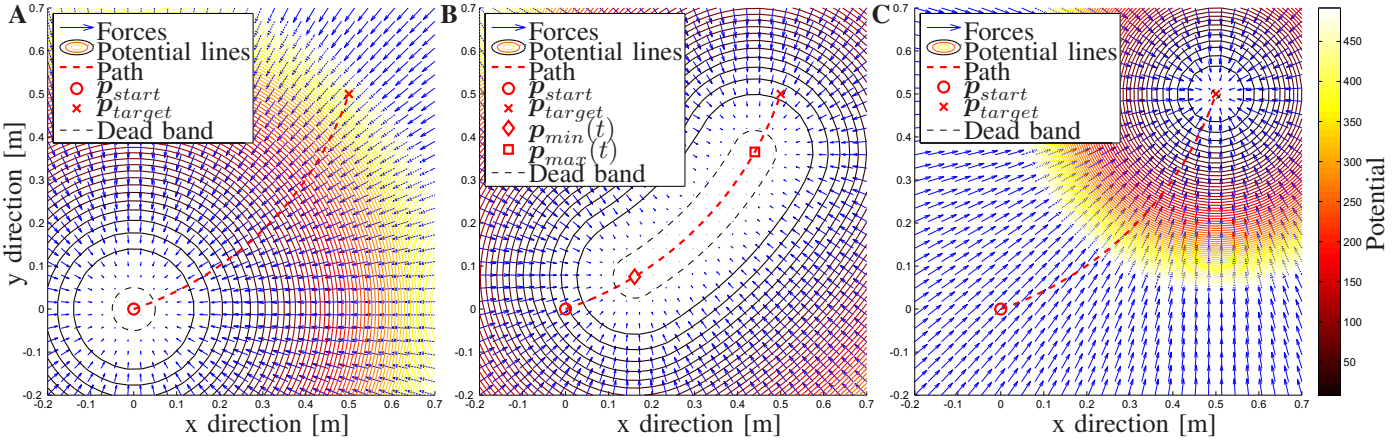

Fig. 4. Development of the virtual tunnel towards the target position. The arrows are the calculated forces that point in the direction of the nearest neighbor on the path and the potential lines are shown for better understanding of the tunnel shape. Furthermore, the dead band is indicated with a dashed line. A: Initial situation, an impedance controller keeps the hand at the start position; B: As soon as the target position is shown, the minimum and maximum speed trajectories start spanning the virtual tunnel in which the patient can freely move; C: An impedance controller attracts the end-effector to the target position.

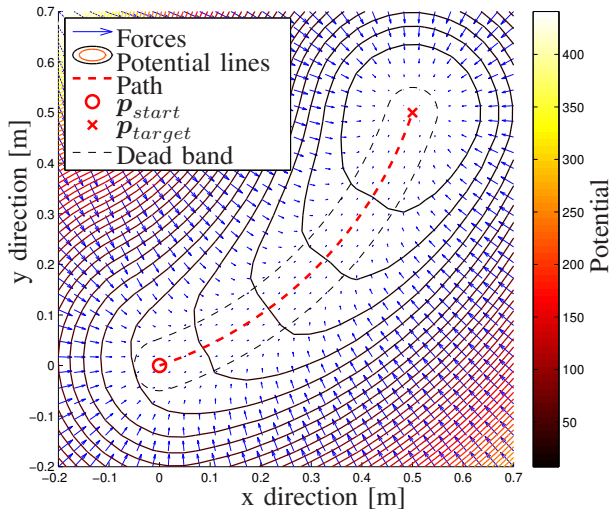

Fig. 5. Tunnel- and flux force field applied to the end effector and according potential lines for better visualization of the tunnel and the flux.

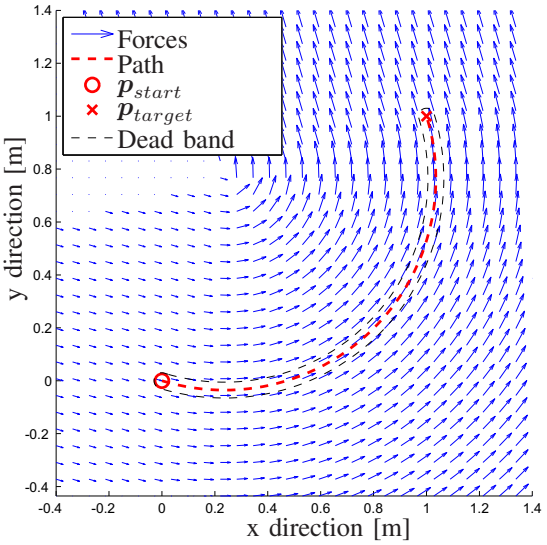

Fig. 6. Force field of the direction-dependent flux to support the end-effector movement along the path. The force field is the highest for the upward direction. The tunnel force is set to zero for this visualization.

superposition of the flux and the tunnel forces (As shown in Fig. 5):

$$\boldsymbol{f}_{pc} = \boldsymbol{f}_{flux} + \boldsymbol{f}_{t} \tag{3}$$

In order to account for different directions, the flux should be higher, when the inclination of the path from the horizontal is increased. The chosen solution to account for this is a direction-dependent flux factor k_{dd} that is used to calculate the flux:

$$\boldsymbol{f}_{flux} = k_{dd}(\boldsymbol{t}_{flux}) \cdot k_{flux} \cdot \boldsymbol{t}_{flux} \tag{4}$$

where $k_{dd}(\boldsymbol{t}_{flux})$ is calculated by means of the scalar product of the tangent vector $\boldsymbol{t}_{flux}$ and the unit vector in the vertical direction $\boldsymbol{e}_y$:

$$k_{dd}(\boldsymbol{t}_{flux}) = \frac{1}{2}(\boldsymbol{t}_{flux}^{\mathrm{T}} \cdot \boldsymbol{e}_y + 1) \tag{5}$$

With the new flux factor, the complete flux force is applied, when the path is pointing in the upright direction, half the flux, when the path is horizontal and no flux, when the arm is moving top down. An example of the calculated direction-dependent flux force field is shown in Fig. 6.

F. Gain scheduling

Depending on the stiffness of the tunnel wall, it may not be possible to reach the target, e.g., if the patient is completely passive or has a mild spasticity, the minimum speed wall will push the patient close to the target position and the impedance controller will keep the patient there, but the hand will deviate from the target position because of the arm's weight pushing into the wall of the impedance controller. In order to bring the hand closer to the target position, the stiffness of the impedance controller is continuously increased and the dead band radius R_w is decreased, until the hand is in the desired target region, or the predefined maximum stiffness is reached. This is indicated in Fig. 4 in panel C with the steeper potential around the target position (compared to panel A). In the currently implemented version, the algorithm first waits 2 seconds after the minimum speed wall reached the target and then starts increasing the stiffness. This allows the patient to first try to reach the target on his own.

G. Implementation of the path controller

The adopted AAN path control strategy is implemented in MATLAB/Simulink, and real-time code is executed on a desktop PC running xPC target (Mathworks). The sampling rate is 1 kHz. A simplified version of the control chart can be found in Fig. 7. On the right side is the PASCAL robot in interaction with the patient's arm ($\boldsymbol{\tau}_{int}$). Position sensors measure the actual angles $\boldsymbol{q}_{act}$ in the kinematics structure. The forward kinematics of the robot is used to calculate the actual end-effector position $\boldsymbol{p}_{act}$, which is fed into the visual interface implemented in the Unity game engine (Unity Technologies). The interface gives visual feedback about the end-effector position as well as the start position $\boldsymbol{p}_{start}$ and target position $\boldsymbol{p}_{target}$ that define the current path. This information is used to calculate the nearest neighbor point $\boldsymbol{p}_{NN}$ on the path and the tangential vector $\boldsymbol{t}_{flux}$ by means of the end-effector position. Additionally, the minimum and maximum speed trajectories ($\boldsymbol{p}_{min}(t)$,$\boldsymbol{p}_{max}(t)$) are calculated using the minimal and maximal speed values defined by the therapist in the settings of the visual interface. The speed trajectories, $\boldsymbol{p}_{NN}$, $\boldsymbol{t}_{flux}$ and the current position are used to calculate the path control forces $\boldsymbol{f}_{pc}$ according to (3). The transposed Jacobian $\mathbf{J}^T$ is used to calculate the corresponding joint torques $\boldsymbol{\tau}_{pc}$ that are used together with the torques from the compensation model $\boldsymbol{\tau}_{comp}$ (consisting of friction, gravity and spring compensation) as a reference for the torque-controlled motor drives (Maxon motor, Switzerland).

H. Test setup

A first test was performed with the AAN path controller by using a simple interface, where the end-effector position of PASCAL was represented as a sphere on the screen. The task was to move to one of eight different targets located at a distance of 10 cm around a starting position (Fig. 8). In order to avoid having gravity always act in the same direction with respect to the path, and for better understanding of the test results, the targets and the starting position were placed

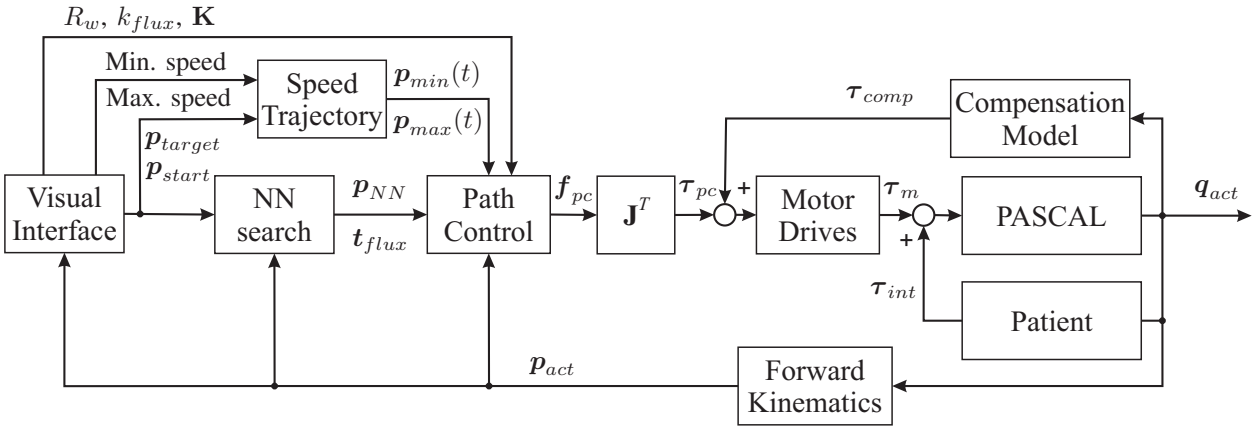

Fig. 7. Control chart of the AAN path controller implemented in the PASCAL robot.

in the vertical x-y plane (frontal plane) at a distance of 0.4 m from the subjects shoulder. Each of the eight targets had to be reached two times and the calculated reference path was a straight line from start to the target position. The robot safety features correspond to the criteria of the ethics committee of the Kanton of Zurich.

I. Evaluation of the AAN path controller

The controller was tested for two extremal conditions, i.e., for a completely passive user, who needs full support and an active user who needs no assistance. The performance of a patient is expected to be in between these extremal conditions. However, a few children with CP and a spastic arm may need full assistance from the different support features to reach a target. Furthermore, the passive condition can be used for arm mobilization or to instruct movement tasks. In order to test the full support condition, a subject was instructed to stay passive while the robot moves the left hand. Three passive arm conditions were used with different control parameter sets to show that the robot is able to perform the movement without any active subject contribution and to investigate the influence of the different control parameters.

The first condition tested whether the target can be reached when choosing a low tunnel wall stiffness and without flux. In this condition, the wall stiffness was 500 N/m, the minimum speed 0.03 m/s, the maximum speed 0.2 m/s, the maximum stiffness of the impedance controller at the target position 5000 N/m and the radius R_w was 0.01 m. In the second condition, the wall stiffness was increased to 3000 N/m and R_w decreased to 0.005 m, while the other parameters were unchanged, to test the influence of the tunnel wall stiffness and the minimum speed wall on the tunnel forces and the distance to the target. The third condition was used to show the influence of the flux on the forces that are applied by the virtual tunnel and on the movement timing. A flux gain k_{flux} of 12 N was used.

In a fourth condition, the subject was instructed to actively move to the target as straight as possible, in order to show, that the controller is only assisting if needed. Here, the flux gain was set to 0 N, while the other parameters were unchanged.

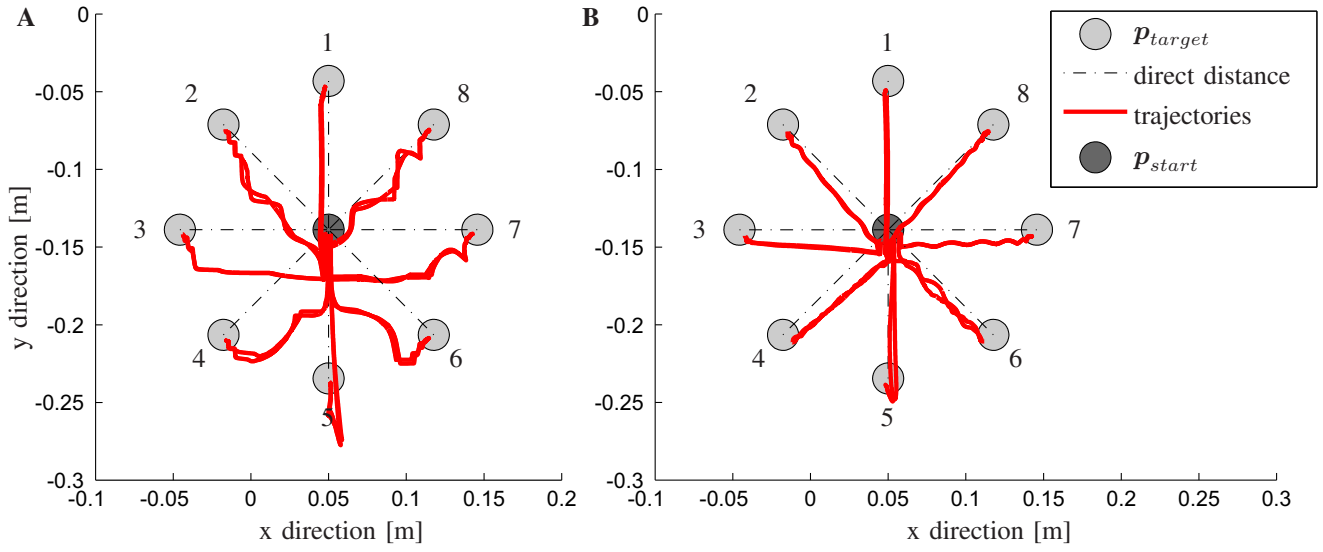

Fig. 8. End-effector trajectories from the start position (center) to the 8 different target positions with the subject staying passive (movement back to start position not shown). The y direction corresponds to the vertical direction; A: Condition 1, tunnel walls with low stiffness lead to high deviation from the direct path; B: Condition 2, increasing the tunnel wall stiffness and decreasing R_w brings the arm trajectories closer to the reference path.

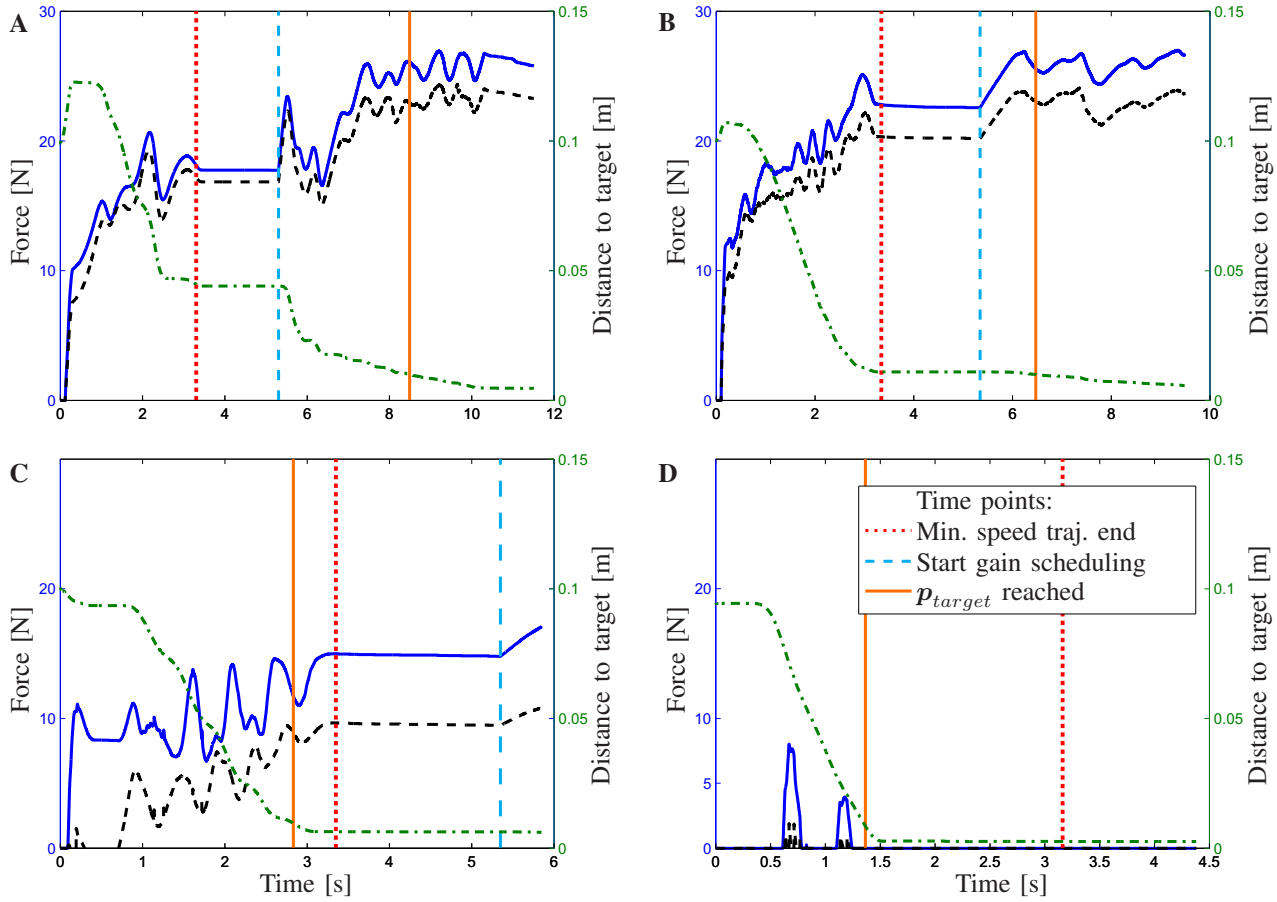

Fig. 9. Detailed analysis of the arm trajectory to target number 8; A: Condition 1, passive arm, low wall stiffness and no flux lead to high deviations from the direct path; B: Condition 2, passive arm, high wall stiffness and no flux lead to decreased deviation from the path compared to A, tunnel forces similar to A; C: Condition 3, passive arm, high wall stiffness and with flux lead to decreased tunnel forces and decreased tangential forces compared to A and B, target reached earlier; D: Condition 4, active arm, high wall stiffness and no flux lead to nearly no tunnel forces supporting the movement; At time point 0 the target is shown. The solid line is the norm of the tunnel force $\boldsymbol{f}_t$. The dash-dotted line shows the distance to the target. The dashed line displays the tangential contribution to the tunnel force. The dotted vertical line shows when the minimum speed trajectory reached the target position. The dashed vertical line indicates the onset of the gain scheduling, which starts 2 s after minimum speed trajectory reached the target. The solid vertical line shows when the target was reached, i.e. $\boldsymbol{p}_{act}$ is closer than 0.01 m to the target position.

III. Results

The resulting end-effector trajectories for the first two conditions can be seen in Fig. 8. All targets were successfully reached twice, while the deviation from the direct path was higher for the first condition (Fig. 8, A) due to the low wall stiffness. In the second condition (Fig. 8, B) the arm deviated less from the direct path. The deviation in position was highest for the horizontal movements where the weight of the arm is perpendicular to the path, which led to a deeper penetration into the wall.

For a more detailed analysis, the movements to target 8 (Fig. 8) were considered in more detail. Panel A in Fig. 9 shows the movement to target 8 for the first test condition described before (Fig. 8, A). As the arm was passive, the minimum speed wall started moving the arm towards the target position. This is indicated by the increasing tunnel forces and the decreasing distance to the target position. Once the minimum speed trajectory reached the target, the forces and position stayed constant. When the gain scheduling started, the tunnel forces increased and helped to reduce the distance to the target and finally the target position was reached. Panel B in Fig. 9 shows the second condition. Compared to the first condition, the increased stiffness kept the end-effector position closer to the path during the movement. Once the minimum speed trajectory reached the target, the distance to the target position is smaller than in condition 1. As a consequence, the target was reached quicker, after the gain scheduling started. In the third condition (Fig. 9, C) with the flux, the tunnel force was decreased, and the tangential part of the tunnel force was much smaller compared to the previous conditions. The remaining tangential tunnel force resulted from the minimum speed wall. In this condition, the target position was reached slightly before the minimum speed trajectory reached the target position.

In the last condition (Fig. 9, D), with active participation, the target was reached in a bit more than a second and, therefore, in half the time that the minimum speed trajectory needed to reach the target. As long as the subject moved in the dead band, there was no supporting force. At two times the arm left the dead band and touched the virtual wall, producing a small supporting force.

IV. Discussion

The results of the four different conditions show that the AAN path controller can be used to support the upper extremity during a reaching task. To adjust the assistance according to the patient's needs, the therapist can activate and change four different supporting features independent of each other. The influence of these features was tested in different conditions. Condition 1: The virtual tunnel helps the arm to stay in the vicinity of the desired path and the minimum speed trajectory was shown to bring the arm close to the target in a predefined time. Combined with the target gain scheduling the target could always be reached. Condition 2: Increasing the wall stiffness constrains the arm movement to closely follow the desired path. Condition 3: The direction-dependent flux force helps to move along the tunnel, which is indicated by the decreased tunnel forces and tangential forces. This support is mainly important for an arm movement with a speed higher than the minimum speed, i.e. for patients that actively contribute to the movement, but are not strong enough to complete the task without support. The presented direction-dependent flux factor presented here can be adapted to the desired amount by the therapist, i.e. patients may need more support to the left than to the right direction, or even resistance in a certain direction. Condition 4: Here we showed, when the subject is actively participating, there is only a small or no force that is helping along the path and there is no force from the minimum speed wall or the gain scheduling, i.e. the subject can freely move to the target position.

V. Conclusion and Outlook

The new AAN path controller was tested and applied to a healthy subject. It could be shown that the controller acts as an assist-as-needed controller and covers the support range from a completely passive arm that needs full assistance to a completely active movement, where the support is unnecessary. The AAN path controller was yet only used for healthy subjects. In a next step, the controller will have to be used in patients for further evaluation of the new control strategy. We assume that the AAN path control algorithm assists the patients enough such that they can actively finish a reaching movement while staying challenged throughout the therapy.

ACKNOWLEDGMENT

The authors would like to thank Maria Rozou, who worked on this project during her semester thesis, Volker Bartenbach for improving the safety of the robot, Aniket Nagle for implementing the game interface for the robot as well as Joachim v. Zitzewitz and Hubertus v. Hedel for valuable discussions and feedback.

References

[1] N. Hogan, H. Krebs, B. Rohrer, J. Palazzolo, L. Dipietro, S. Fasoli, J. Stein, R. Hughes, W. Frontera, D. Lynch *et al.*, "Motions or muscles? some behavioral factors underlying robotic assistance of motor recovery," *Journal of rehabilitation research and development*, vol. 43, no. 5, p. 605, 2006.

[2] D. Damiano, "Activity, activity, activity: rethinking our physical therapy approach to cerebral palsy," *Physical therapy*, vol. 86, no. 11, pp. 1534–1540, 2006.

[3] J. Patton, M. Stoykov, M. Kovic, and F. Mussa-Ivaldi, "Evaluation of robotic training forces that either enhance or reduce error in chronic hemiparetic stroke survivors," *Experimental Brain Research*, vol. 168, no. 3, pp. 368–383, 2006.

[4] R. Schmidt, "Frequent augmented feedback can degrade learning: Evidence and interpretations," *Tutorials in motor neuroscience*, pp. 59–75, 1991.

[5] R. Sigrist, G. Rauter, R. Riener, and P. Wolf, "Augmented visual, auditory, haptic, and multimodal feedback in motor learning: A review," *Psychonomic Bulletin & Review*, pp. 1–33, 2012.

[6] N. Hogan, "Impedance control: An approach to manipulation," in *American Control Conference, 1984*. IEEE, 1984, pp. 304–313.

[7] J. Emken, J. Bobrow, and D. Reinkensmeyer, "Robotic movement training as an optimization problem: designing a controller that assists only as needed," in *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on*. IEEE, 2005, pp. 307–312.

[8] H. Krebs, J. Palazzolo, L. Dipietro, M. Ferraro, J. Krol, K. Rannekleiv, B. Volpe, and N. Hogan, "Rehabilitation robotics: Performance-based progressive robot-assisted therapy," *Autonomous Robots*, vol. 15, no. 1, pp. 7–20, 2003.

[9] R. Riener, L. Lunenburger, S. Jezernik, M. Anderschitz, G. Colombo, and V. Dietz, "Patient-cooperative strategies for robot-aided treadmill training: first experimental results," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 13, no. 3, pp. 380–394, 2005.

[10] L. Cai, A. Fong, C. Otoshi, Y. Liang, J. Burdick, R. Roy, and V. Edgerton, "Implications of assist-as-needed robotic step training after a complete spinal cord injury on intrinsic strategies of motor learning," *The Journal of neuroscience*, vol. 26, no. 41, pp. 10 564–10 568, 2006.

[11] S. Banala, S. Agrawal, and J. Scholz, "Active Leg Exoskeleton (ALEX) for gait rehabilitation of motor-impaired patients," in *Rehabilitation Robotics, 2007. ICORR 2007. IEEE 10th International Conference on*. IEEE, 2007, pp. 401–407.

[12] G. Rauter, R. Sigrist, L. Marchal-Crespo, H. Vallery, R. Riener, and P. Wolf, "Assistance or challenge? Filling a gap in user-cooperative control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 3068–3073.

[13] A. Duschau-Wicke, J. von Zitzewitz, A. Caprez, L. Lunenburger, and R. Riener, "Path control: A method for patient-cooperative robot-aided gait rehabilitation," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 18, no. 1, pp. 38–48, 2010.

[14] M. Guidali, A. Duschau-Wicke, S. Broggi, V. Klamroth-Marganska, T. Nef, and R. Riener, "A robotic system to train activities of daily living in a virtual environment," *Medical and Biological Engineering and Computing*, pp. 1–11, 2011.

[15] A. Koenig, U. Keller, K. Pfluger, A. Meyer-Heim, and R. Riener, "PASCAL: Pediatric arm support robot for combined arm and leg training," in *4th International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 2012, pp. 1862–1868.

[16] R. Riener, L. Lünenburger, I. Maier, G. Colombo, and V. Dietz, "Locomotor training in subjects with sensori-motor deficits: an overview of the robotic gait orthosis lokomat," *Journal of Healthcare Engineering*, vol. 1, no. 2, pp. 197–216, 2010.

[17] L. Marchal-Crespo and D. Reinkensmeyer, "Review of control strategies for robotic movement training after neurologic injury," *Journal of neuroengineering and rehabilitation*, vol. 6, no. 1, p. 20, 2009.

[18] J. Herder, N. Vrijlandt, T. Antonides, M. Cloosterman, P. Mastenbroek *et al.*, "Principle and design of a mobile arm support for people with muscular weakness," *Journal of rehabilitation research and development*, vol. 43, no. 5, p. 591, 2006.

[19] H. Vallery, A. Duschau-Wicke, and R. Riener, "Generalized elasticities improve patient-cooperative control of rehabilitation robots," in *International Conference on Rehabilitation Robotics (ICORR)*. IEEE, 2009, pp. 535–541.

[20] L. Marchal-Crespo, G. Rauter, D. Wyss, J. von Zitzewitz, and R. Riener, "Synthesis and control of an assistive robotic tennis trainer," in *4th International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 2012, pp. 355–360.