Abstract—An increasingly large number of people in industrialized countries suffers from a stroke every year. The rehabilitation robot ARMin has been developed for the arm rehabilitation of stroke patients and other neurological patients. In this work, a control strategy for training Activities of Daily Living (ADL) with ARMin is presented. The training of ADLs motivates the subject and facilitates relearning of functional movements. A combination of impedance and admittance control is implemented and a control strategy to support the subject in a patient-cooperative way during the training is introduced. The control strategy is based on the minimum intervention principle thus maximizing the subject’s own contribution to the task. First results with healthy subjects are presented. The algorithms are described as a general concept and can also be applied to other rehabilitation robots.

I. INTRODUCTION

Today, stroke is the leading cause of disabilities in industrialized countries. Many stroke patients are paralyzed on one side of the body or suffer from a one-sided weakness. The rehabilitation process can be significantly improved by means of early intervention using intensive arm therapy [1], [2], [3]. There is evidence that robot-aided therapy can be particularly effective in progressing the treatment [4]. New control methodologies have been introduced that allow interaction between subject and robot thus enabling new possibilities for rehabilitation and guaranteeing safety for the subject [5], [6], [7]. In order to intensify and support the rehabilitation process of the upper extremities, the arm rehabilitation robot ARMin was developed [8], [9].

A. Arm rehabilitation robot ARMin

The control strategy presented in this work is implemented on the arm rehabilitation robot ARMin. The robot is realized as a multi-modal display stimulating the three most important sensory channels of the subject: haptic (human/robot) interaction, visual (graphical display), and acoustic (sound system). This work focuses on the control of the haptic display. Having six DOF, the exoskeleton can mimic human arm movements in a natural way. The mechanical structure of ARMin is shown in Fig. 1.

B. Rationale for Activities of Daily Living

Since the therapeutic outcome highly depends on training intensity and duration, motivation is a key success factor of the rehabilitation. The training tasks should be intuitively understandable for the subjects. Furthermore, it is very important that the subjects relearn functional movements and the coordination of their movements [10], [11], [12].

These needs can be fulfilled with the implementation of ADLs. An ADL like eating or combing hair is meaningful to the patient and can therefore be expected to increase motivation. Furthermore, virtual environments known from daily living support intuitive understanding of the task. ADLs consist of coordinated movements, where both position and orientation of the human hand need to be controlled. Finally, with the training of ADLs, the subject trains functional movements. This is expected to improve the transfer of learning effects to real life.

The problem of implementing an ADL on ARMin can be summarized as follows: A support for the training needs to be defined in such a way that the subject’s own contribution is maximized. Still, the subject should be guided when not able to complete the task by himself (minimum intervention principle).

C. Requirements for the control structure

In order to implement an ADL as defined above, the control strategy has to fulfill the following requirements:
• **Patient-cooperative support**: The subject should be supported only as much as necessary according to the minimum intervention principle thus maximizing his own contribution.

• **Variety**: The control strategy should be applicable to a large variety of ADLs. A general solution is therefore preferred.

• **Safety**: All implementations need to comply with the high safety standards in rehabilitation engineering.

## II. CONTROL STRUCTURE

Haptic displays can generally be classified into two categories, those which measure motion and display force and those which measure force and display motion [13]. The former are referred to as **impedance displays** and the latter as **admittance displays**. Impedance displays typically have low inertia and are highly backdriveable, while admittance displays often have high inertia, are non-backdriveable, fitted with force sensors, and are driven by a position and velocity control loop. An overview of impedance and admittance controllers can be found in [14]. Furthermore, it is desirable, that the subject has a realistic impression of the virtual environment and does not detect the dynamics of the haptic device. Transparency has therefore always been regarded as an important performance measure of haptic displays [15], [16].

The inner velocity loop of the admittance controller has high dynamics and can rapidly compensate for disturbances. Since some axes of ARMin have high friction and high inertia, admittance control is generally preferred for ARMin, because it ensures better performance. However, in order to use admittance control, a force/torque sensor is needed. Since there are only signals from the force/torque sensor available for Axes 1 and 2, only these axes can be admittance controlled. Axes 3 to 6 have to be impedance controlled.

The control structure used for ARMin is depicted in Fig. 2. The admittance of the haptic device is given in joint space as $J^T Z_c^{-1}$. The inputs are the joint torques and the outputs are the velocities $\dot{q}$ in the joint space. These are mapped to the velocity of the hand $v_h$ in Cartesian space by the Jacobian $J$. This velocity is subtracted from the reference velocity $v_r$, and the resulting difference is input to the virtual environment $Z_c$. The virtual environment is modelled as an impedance. This means that it takes positions or velocities as inputs and returns the force vector $f_s$. The forces $f_s$ are then mapped with the transposed Jacobian $J^T$ to the torques $\tau_c$ in the joint space.

Axes 3 to 6 are impedance controlled. Thus, the torques $\tau_{c3-6}$ are directly fed back to the haptic device. For Axes 1 and 2, the signals from the force/torque sensor $\tau_s$ are subtracted from the torques $\tau_c$. This difference is input to the controller admittance $J^T Y_c$. The outputs of the admittance filter are the velocities $\dot{q}_c$ in the joint space. The velocities $\dot{q}$ in the joint space are subtracted from the reference velocities $\dot{q}_c$. The difference is input to a PI controller. The outputs are the torques $\tau_{c1,2}$, which are fed to the haptic device.

To judge on transparency of the system, the impedance $Z_{CL}$ felt by the subject when coupled to the robot and operating in the virtual environment with the overall impedance $Z_c = f_s/v_h$ will be estimated for the two cases of impedance and admittance control.

### 1) Impedance control: From Fig. 2 it can be derived

\[
\begin{align*}
\dot{v}_h &= J^T Z_c^{-1} [\dot{Z}_r J^{-1} v_h + J^T f_s - J^T F_h] \quad \text{(1)} \\
\tau_s &= Z_c (v_r - v_h) \quad \text{(2)}
\end{align*}
\]

where $F_h$ is the force applied by the subject. Defining the open-loop impedance of the haptic device $Z_r \equiv J^{-T} Z_c J^{-1}$ gives

\[
\dot{v}_h = Z_r^{-1} [\dot{Z}_r J^{-1} v_h + Z_c (v_r - v_h) - F_h] \quad \text{(3)}
\]

Without loss of generality, $v_r$ can be set equal to zero. This results in the closed-loop impedance felt by the subject

\[
Z_{CL} = \frac{-F_h}{v_h} = Z_c + Z_r - \hat{Z}_r, \quad \text{(4)}
\]

with $\hat{Z}_r \equiv J^{-T} \hat{Z}_r J^{-1}$ representing the dynamic model of the haptic device in Cartesian space. For $\hat{Z}_r = Z_c$, the last two terms cancel out and the impedance felt by the subject $Z_{CL}$ approaches the impedance of the virtual environment $Z_c$, meaning a good transparency is achieved.

### 2) Admittance control: From Fig. 2 it can be derived

\[
\begin{align*}
\dot{v}_h &= J^T Z_c^{-1} [\dot{Z}_r J^{-1} v_h + \tau_c - J^T F_h] \quad \text{(5)}
\end{align*}
\]

Using the relationships $Z_r \equiv J^{-T} Z_c J^{-1}$ and $\hat{Z}_r \equiv J^{-1} \hat{Z}_r J^{-1}$, this yields

\[
\dot{v}_h = Z_r^{-1} [\dot{Z}_r v_h + J^{-T} \tau_c - F_h] \quad \text{(6)}
\]

With (2) and setting $v_r = 0$ the torque $\tau_c$ is

\[
\tau_c = D [Y_c (-J^T Z_c v_h - J^T F_h) - J^{-1} v_h], \quad \text{(7)}
\]
where
\[ D = J^{-T} \dot{D} J^{-1} \]  
(8)
is the robot joint space controller transfer function expressed in Cartesian space, and
\[ Y_c = J^T Y_c J \]  
(9)
is the joint space admittance transfer function of the robot controller expressed in Cartesian space. Using simple algebraic manipulation of (6) and (7), it is possible to estimate the closed loop impedance felt by the subject as
\[ Z_{CLA} = \frac{-F_h}{v_h} = (DY_c + I)^{-1}(Z_e - \dot{Z}_e + D(Y_c Z_e + I)) \]
(10)
where I is the identity matrix. With increasing gains of the PI controller, $\dot{D} \rightarrow \infty$, the closed-loop impedance simplifies to
\[ Z_{CLA} \approx Z_e + Y_c^{-1}. \]
(11)
If the admittance of the robot controller $Y_c$ is high, then the closed-loop impedance $Z_{CLA}$ approaches the impedance of the virtual environment $Z_e$.

**III. CONTROL STRATEGIES FOR THE VIRTUAL ENVIRONMENT**

In the overall control structure as explained above, the virtual environment with the ADL is implemented as an impedance. Therefore, the supportive force has to be defined based on the velocity or position error. The control strategy supporting the ADLs is explained in two steps. First, a patient-cooperative support for point-to-point movements is explained. From this, a strategy for the support of arbitrary ADL movements (including curved ones) is derived. The control strategy is first described only for controlling the position of the hand and then extended to its orientation.

**A. Patient-cooperative support for point-to-point movements**

In order to define support, an ideal reference trajectory is needed. There are generally several ways to model unconstrained human point-to-point movements in three-dimensional space [17], [18]. Hogan [19] claims that humans do smooth movements and these are characterized by the fact that they minimize jerk, which is the third time derivative of position
\[ \ddot{x}(t) = \frac{d^3x(t)}{dt^3}. \]
(12)
The smoothness of a particular trajectory $x_p(t)$ can be measured by calculating the jerk cost
\[ \int_{t_i}^{t_f} \ddot{x}_p(t)^2 dt. \]
(13)
To find the minimum jerk trajectory, a jerk cost needs to be assigned to each possible trajectory and the trajectory with the smallest cost has to be found. This is equivalent to minimizing the functional
\[ H(x(t)) = \frac{1}{2} \int_{t_i}^{t_f} \ddot{x}^2 dt. \]
(14)
This minimization problem is solved by calculus of variations. This yields
\[ \frac{dH(x + \eta)}{d\eta}\bigg|_{\eta=0} = -\int_{t_i}^{t_f} \eta \ddot{x}^{(6)} dt = 0, \]
(15)
where $\eta$ is the function representing the variation and $c$ is an arbitrary parameter. For further explanation on calculus of variations and a detailed derivation see [20]. This must hold for any $\eta(t)$, therefore
\[ \ddot{x}^{(6)} = 0. \]
(16)
This means that some function $x(t)$ that has its sixth derivative equal to zero will minimize jerk. The differential equation $\ddot{x}^{(6)} = 0$ has the general solution
\[ x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5, \]
(17)
where $a_0...a_5$ are the polynomial parameters. The derivatives are
\[ \dot{x}(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \]
(18)
\[ \ddot{x}(t) = 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3. \]
(19)
To find the polynomial parameters, the equations are evaluated at the boundaries $x(t_i), \dot{x}(t_i), \ddot{x}(t_i), \dddot{x}(t_i), \ddddot{x}(t_i), \dddddot{x}(t_i)$, where $t_i$ is the initial time at the beginning of the movement and $t_f$ is the final time at the end of the movement. The six unknowns can be determined using these six equations.

The minimum jerk trajectory can also be formulated as a feedback controller [21]. For this, time is normalized to
\[ \tau = \frac{t - t_0}{D}, \]
(20)
so that $\tau \in [0, 1]$, where $t$ is the current time and $t_0$ is the initial time and
\[ D = t_f - t_0. \]
(21)
Furthermore, $t_f$ is the final time and
\[ \frac{d\tau}{dt} = \frac{1}{D} \]
(22)
This yields
\[ x(t) = a_0 + a_1 \tau + a_2 \tau^2 + a_3 \tau^3 + a_4 \tau^4 + a_5 \tau^5 \]
(23)
\[ \dot{x}(t) = \frac{a_1}{D} + \frac{2a_2}{D} \tau + \frac{3a_3}{D} \tau^2 + \frac{4a_4}{D} \tau^3 + \frac{5a_5}{D} \tau^4 \]
(24)
\[ \ddot{x}(t) = \frac{2a_2}{D^2} + \frac{6a_3}{D^2} \tau + \frac{12a_4}{D^2} \tau^2 + \frac{20a_5}{D^2} \tau^3. \]
(25)
The initial conditions are specified as
\[ x(t_0) = x_i, \quad \dot{x}(t_0) = v_i, \quad \ddot{x}(t_0) = p_i. \]
(26)
When $t = t_0$, this yields $\tau = 0$ and therefore
\[ a_0 = x_i, \quad a_1 = Dv_i, \quad a_2 = \frac{Dp_i}{2}. \]
(27)
The conditions for the end of the movement are
\[ x(t_f) = x_f \quad \dot{x}(t_f) = 0 \quad x(t_f) = 0. \quad (28) \]
When \( t = t_f \), this yields \( \tau = 1 \) and therefore
\[
\begin{align*}
a_3 &= \frac{-3D^2}{2} p_i - 6Dv_i + 10(x_f - x_i) \quad (29) \\
a_4 &= \frac{3D^2}{2} p_i + 8Dv_i - 15(x_f - x_i) \quad (30) \\
a_5 &= \frac{-D^2}{2} p_i - 3Dv_i + 6(x_f - x_i). \quad (31)
\end{align*}
\]
At any time into the movement, the current time can be labelled as \( t = t_0 \) and the current state can be assumed to be
\[
\chi(t_0) = [x_i \quad v_i \quad p_i]^T. \quad (32)
\]
Then the jerk is
\[
\ddot{x}(t) = \frac{6a_{x3}}{D^3} + \frac{24a_{x4}}{D^3} \tau + \frac{60a_{x5}}{D^3} \tau^2. \quad (33)
\]
At \( t = t_0 \), \( \tau = 0 \) and \( D = t_f - t \). This yields
\[
\ddot{x}(t_0) = \frac{6a_{x3}}{D^3} = \frac{60}{D^3}(x_f - x(t_0)) - \frac{36}{D^2} \ddot{x}(t_0) - \frac{9}{D} \dddot{x}(t_0), \quad (34)
\]
which can be written as
\[
\dot{\chi} = \begin{bmatrix} \dot{x} \\ \dot{\ddot{x}} \\ \dot{\dddot{x}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\frac{60}{D^3} & -\frac{36}{D^2} & -\frac{9}{D} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{60}{D^3} \end{bmatrix} x_f. \quad (35)
\]
This minimum jerk trajectory can now be used to calculate a supportive force, which is applied in a patient-cooperative way. A reference point \( p_r \) is defined for each instant in time that is characterized by the optimal distance to the target according to the minimum jerk trajectory. Then the distance from the reference point \( p_r \) to the goal \( g \) is given by
\[
d_{ref} = \|g - p_r\|. \quad (36)
\]
With this, the subject has a certain time constraint given by the minimum jerk trajectory. At each simulation step, the distance \( d_{act} \) from the actual position of the hand \( p_h \) to the goal \( g \) is calculated, which is given by
\[
d_{act} = \|g - p_h\|. \quad (37)
\]
The situation is shown in Fig. 3. Now, a supportive force \( f_s \) is applied in the direction of the goal, if the movement of the subject resulted in a point that is too far away to reach the goal in the time defined by the minimum jerk trajectory.

The unit vector \( h \) pointing from the current hand position towards the goal is calculated as
\[
h = \frac{g - p_h}{\|g - p_h\|}. \quad (38)
\]
With \( k_p \) as a controller gain, the supporting force is then given as
\[
f_s = \begin{cases} k_p \cdot |d_{ref} - d_{act}| \cdot h & \text{if } d_{act} > d_{ref} \\ 0 & \text{if } d_{act} \leq d_{ref}. \end{cases} \quad (39)
\]
This approach allows guiding the subject from a start point towards a goal giving him freedom to move on his own trajectory. If he does not approach the goal rapidly enough, he will be supported depending on his actual position. It is therefore guaranteed that the subject reaches the goal while giving him freedom to choose his own trajectory and applying support according to the subject’s performance. A patient-cooperative support is thus realized.

To make the controller more robust, damping is added in all three directions. A local coordinate frame is defined for that purpose with axis \( \phi_z \) pointing from the current hand position towards the goal
\[
\phi_z = \frac{g - p_h}{\|g - p_h\|}. \quad (40)
\]
Axes \( \phi_x \) and \( \phi_y \) can be arbitrarily selected. Here, they are defined as
\[
\phi_x = \frac{p_h \times \phi_z}{\|p_h \times \phi_z\|} \quad (41)
\]
and
\[
\phi_y = \phi_z \times \phi_x. \quad (42)
\]
The transformation matrix of the local coordinate frame \( \Phi \) with respect to the task coordinate frame is then given as
\[
\Phi = [\phi_x \quad \phi_y \quad \phi_z]. \quad (43)
\]
The hand velocity is used for damping. The hand velocity with respect to the local frame \( \Phi \) is
\[
\phi_v_h = \Phi^T v_h. \quad (44)
\]
The supportive force in the local coordinate frame is computed as
\[
\Phi_s = \begin{bmatrix} -k_v \cdot \vec{v}_h_z \\ -k_v \cdot \vec{v}_h_y \\ -k_v \cdot \vec{v}_h_x + \begin{cases} k_p \cdot |d_{ref} - d_{act}| & \text{if } d_{act} > d_{ref} \\ 0 & \text{if } d_{act} \leq d_{ref} \end{cases} \end{bmatrix}
\]

In the task frame, this force is given by
\[
f_s = \Phi \cdot \Phi_s.
\]

### B. Patient-cooperative support for arbitrary ADLs

Supporting arbitrary ADLs rather than just point-to-point movements is more complex since they can consist of curved movements. Still, in order to have a large variety of supported ADLs, it is compulsory to define a support for arbitrary ADL movements. Once again, an ideal trajectory for the movement needs to be defined first.

The ideal trajectory should again mimic natural human behaviour while allowing curved movements. This can be achieved by using the feedback controller as presented in the previous paragraph but having the goal of the point-to-point movement not static but itself moving. Using the minimum jerk trajectory ensures natural movements, while the moving goal realizes curved trajectories. This is due to the fact that the moving goal position is continuously updated in the feedback controller and the minimum jerk trajectory is thus continuously updated, too. To realize curves, the goal has to move along a curved trajectory. Then, the reference point is also moving along a similar trajectory based on the minimum jerk principle. The situation is depicted in Fig. 4.

![Fig. 4. Calculation of the supportive force for arbitrary ADL movements. The goal is moving on a predefined trajectory and the reference point is following according to the minimum jerk trajectory. The force field indicates the supportive forces for each hand position in space. Inside the sphere, the supportive force is equal to zero.](image)

The trajectory for the goal movement can be defined in different ways. Here, it was done by predefining via points and then using interpolation of linear segments with parabolic blends. For each instant in time during the movement, the current goal position is thus given, which provides an input to the feedback controller and to the calculation of the reference point position. These positions can now be used to determine a supportive force based on (45) and (46).

1) **Adaptable freedom of motion:** The freedom of motion depends on the distance \(d_{ref}\) between the reference point and the goal. Since the subject only gets support if the distance of the current hand position to the goal \(d_{act}\) is larger than the distance between reference point and goal, \(d_{ref}\) defines the radius of a sphere with its center aligned with the goal position. If the current hand position is inside this sphere, the subject is not supported. If the hand position is outside the sphere, a force is applied to the subject’s arm. The situation is shown in Fig. 4.

The subject can be restricted to a larger or smaller degree in his movement by changing \(d_{ref}\). If \(d_{ref}\) approaches zero, then the movement of the subject is position-controlled. In general, it is arguably valid that the subject should be more restricted to the trajectory when passing around obstacles, while he should have more freedom when moving in free space. Therefore, \(d_{ref}\) should depend on the current position of the goal in the virtual environment. In free space, movements will generally follow a straight line. Curved movements are only used when moving around obstacles. The curvature of the goal trajectory can therefore be used as an indication how large \(d_{ref}\) should be. If the radius of curvature is large, \(d_{ref}\) can be large giving the subject a lot of freedom in his movement. If the radius of curvature is small, the subject probably needs to pass an obstacle and should therefore be more restricted to the trajectory. That is why three auxiliary points \(h_1, h_2, h_3\) are moving ahead of the goal. Their positions are given by the goal trajectory at the time \(t + t_{h1}, t + t_{h2}, t + t_{h3}\), where \(t_{h1}, t_{h2}, t_{h3}\) constitute the time offsets of the three help points to the time of the goal \(t_g\). These three help points are used to calculate two vectors and the angle between them is determined as depicted in Fig. 5.

![Fig. 5. Adaptive distance. The reference distance and with it the freedom of motion of the subject is changed according to the curvature of the trajectory.](image)

The two vectors are calculated as
\[
\zeta_1 = h_2 - h_1 \quad \text{and} \quad \zeta_2 = h_3 - h_2.
\]
The angle between \(\zeta_1\) and \(\zeta_2\) is determined as
\[
\delta = \arccos \left( \frac{\zeta_1 \cdot \zeta_2}{|\zeta_1| \cdot |\zeta_2|} \right).
\]
The angle δ is used to calculate a new value for D in (35). This means that the calculation of the minimum jerk trajectory is influenced by the angle delta and thus the distance of the reference point \( p_r \) to the goal \( g \) is adapted.

2) Moving ahead of the goal: It should also be possible for well-performing subjects to do the movement faster than suggested by the trajectory. If the subject is moving ahead, the velocity of the goal should be increased, ensuring that the subject is not slowed down by the goal velocity. This can be done if the position of the goal \( g \), the current hand position \( p_h \), and the reference position \( p_r \) are taken into account as illustrated in Fig. 6.

![Fig. 6. Acceleration of the goal. The goal is accelerated to prevent fast moving subjects from passing the goal.](image)

In order to control the velocity of the goal, two vectors \( \eta_h \) and \( \eta_r \) pointing from the goal towards the current hand position \( p_h \) and to the reference point \( p_r \) respectively are calculated. The vectors are given by

\[
\eta_h = p_h - g \quad \text{and} \quad \eta_r = p_r - g. \quad (49)
\]

The angle between the vectors \( \gamma \) is calculated by

\[
\gamma = \arccos \left( \frac{\eta_h \cdot \eta_r}{|\eta_h| \cdot |\eta_r|} \right). \quad (50)
\]

The velocity of the goal \( v_g \) should only be increased if \( \gamma \) is greater than 45° or 0.7854 rad. The velocity of the goal is calculated as

\[
v_g = \begin{cases} v_{g0} + k_t \cdot (\gamma - 0.7854) & \text{if } \gamma > 0.7854 \\ v_{g0} & \text{if } \gamma \leq 0.7854. \end{cases} \quad (51)
\]

This means that the velocity of the goal \( v_{g0} \) is equal to the predefined velocity \( v_{g0} \) if the subject is moving inside the cone behind the goal. If the subject is moving out of the cone, the goal is accelerated by the difference of the angle \( \gamma \) and 45° multiplied with the controller gain \( k_t \).

C. Orientation subtask

Similarly to the supporting force, which was calculated in the position subtask, a supportive torque for the orientation subtask can be calculated. This torque should guide the orientation of the hand to the goal orientation. An instantaneous axis of rotation \( \pi_z \) is introduced for that purpose. This axis is characterized by allowing the shortest rotation between the actual orientation of the hand \( o_h \) and the goal orientation \( o_g \).

In order to calculate the instantaneous axis of rotation \( \pi_z \), an orientation error matrix is computed from the orientation matrix of the goal \( R_g \) and the orientation matrix of the hand \( R_h \) as

\[
E = R_g R_h^T. \quad (52)
\]

The rotation angle \( \kappa \) between the hand and the goal orientation can be calculated from this as

\[
\kappa = \arccos \frac{\text{Tr}(E) - 1}{2}. \quad (53)
\]

This gives

\[
\pi_z = \frac{1}{2 \sin(\kappa)} \begin{bmatrix} e_{32} - e_{23} \\ e_{13} - e_{31} \\ e_{21} - e_{12} \end{bmatrix}, \quad (54)
\]

where \( e_{ij} \) are the elements of the matrix \( E \). Again, a local coordinate frame is defined, where the axis \( \pi_z \) is given by (54) and \( \pi_x \) and \( \pi_y \) can be chosen arbitrarily. Here, \( \pi_x \) is calculated as

\[
\pi_x = \frac{(g - p_h) \times \pi_z}{\|(g - p_h) \times \pi_z\|} \quad (55)
\]

and

\[
\pi_y = \pi_z \times \pi_x. \quad (56)
\]

The orientation matrix of the local coordinate frame with respect to the base frame is then given as

\[
\Pi = \begin{bmatrix} \pi_x & \pi_y & \pi_z \end{bmatrix}. \quad (57)
\]

Using this result, the minimum jerk principle with the feedback controller as derived in (35) is applied to the orientation similarly as for the position subtask. A reference angle to the target \( o_r \) is calculated. The supporting torque is then calculated based on the error between the reference angle \( o_r \) and the actual angle of the hand to the target \( o_h \). The rotation perpendicular to the axis of instantaneous rotation \( \pi_z \) is damped with a torque proportional to the hand angular velocity. This is expressed in the local coordinate frame as

\[
\Pi \sigma_h = \Pi^T \sigma_h. \quad (58)
\]

The torque expressed in the local coordinate frame is

\[
\Pi \rho = \begin{bmatrix} \kappa_x \rho \Pi \sigma_h \xi \\ -\kappa_y \rho \Pi \sigma_h \eta \\ k_{\pi_z} \sigma_h - \kappa \sigma_h \end{bmatrix} - \begin{bmatrix} k_{\pi_z} (\kappa - \kappa_h) \| \sigma_h \| \rho \Pi \sigma_h \xi \\ 0 \end{bmatrix} \quad (59)
\]

In the task frame, the torque is given by

\[
\rho = \Pi^T \Pi \rho. \quad (60)
\]

With this strategy, a supportive force or torque can be defined that is both position- and time-dependent. This ensures a patient-cooperative support motivating the subject and giving him freedom to move on his own trajectory.
IV. RESULTS

An object transporting task as depicted in Fig. 7 was chosen as ADL to be implemented in Matlab/Simulink. The program runs on a Matlab/Simulink xPC-target with 1 ms loop time. In the beginning, all objects are located on the shelf. The objects need to be moved in a defined order. Whenever an object is grasped, a red marker appears indicating the location the object should be taken to. When the object is transported to that marker, the object rests in its place and a new object has to be grasped. During the movements, the other objects as well as the shelf have to be avoided by moving around them. Additionally, the subject gets acoustic feedback when an object is grasped or set aside. A small fanfare is played when the subject succeeds in putting all objects in the right place on the table.

![Robotic setup and graphical display.](image)

Fig. 7. **Robotic setup and graphical display.** The ADL scenery is presented on the screen, the subject has to fulfil a task and gets patient-cooperative support.

The strategy has been tested on an experimental setup with five healthy subjects. A simple trajectory for the goal was calculated for that purpose by defining three via points. These points are

- Via point 1: \(x = x_{\text{init}}; \ y = y_{\text{init}}; \ z = z_{\text{init}}\)
- Via point 2: \(x = 0.5; \ y = 0; \ z = 0.2\)
- Via point 3: \(x = 0.3; \ y = 0; \ z = 0\),

where \(x_{\text{init}}, y_{\text{init}}\) and \(z_{\text{init}}\) define the position of the hand at the beginning. For the measurements described in this section, the start position was chosen to be \(x = 0.3, \ y = 0.3\) and \(z = 0\). This results in a movement with one curve in the middle. This movement was then executed with a passive and an active subject. The passive subject was instructed to relax as much as possible. The active subject was informed about the trajectory and instructed to perform the movement by himself.

The supportive forces for the passive and the active subject are depicted in Fig. 8. One can see that the force is larger for the passive subject compared to the active subject. At the beginning, the force is zero with the active subject. Still, with the active subject, there also is a force. There are two possible explanations for this. Either the active subject did the movement not very exactly and significantly deviated from the predefined trajectory or he moved very fast and was held back by the supportive force even with acceleration of the goal. Since he only needed half of the time for his movement compared to the passive subject, it is very likely, that the second reason applies here. Still, the supportive force is a lot smaller for the active subject than for the passive one.

![Supportive force](image)

Fig. 8. **Supportive force.** The supportive force is larger for a passive subject (a) compared to an active subject (b).

The additional velocity is depicted in Fig. 9. This is the velocity which is added to the time of the goal in order to accelerate it. For the passive subject shown in Fig. 9(a) the additional time is zero throughout the movement. The reason for this is that the passive subject does not try to pass the goal and therefore he does not have to be accelerated. However, for the active subject, which is depicted in 9(b), the additional time is very important. At the very beginning, the additional time is zero, but then consistently is greater than zero thus accelerating the goal. This is because the active subject permanently tries to pass the goal.

![Additional time](image)

Fig. 9. **Additional time.** The goal is not accelerated for the passive subject (a), but for the active subject it is almost permanently accelerated (b).

In order to assess the performance of the subjects, a quantitative measurement should be given. Since the supportive force and its calculation is the central issue in the strategy, a support index is used as a performance indicator. The norms

![Norm of supportive force](image)

Fig. 10. **Norm of supportive force.** The norm of the supportive force is a lot bigger for the passive subject (a) than for the active subject (b).
are calculated from the force vectors in Fig. 8 and plotted in Fig. 10.

To calculate the support index, the integral of the norm of the supportive force \(\bar{F}_s\) from the initial time \(t_i\) to the final time \(t_f\) is taken.

\[
\text{Support index} = \int_{t_i}^{t_f} ||\bar{F}_s|| \, dt. \tag{61}
\]

The results for the Support index with the passive and active subject are given in Table I.

**TABLE I**

<table>
<thead>
<tr>
<th>Subject</th>
<th>Support index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passive</td>
<td>91.49</td>
</tr>
<tr>
<td>Active</td>
<td>8.92</td>
</tr>
</tbody>
</table>

From these support indices it can be seen that the passive subject is supported over ten times more than the active subject. This support index can be taken as a performance indicator for the therapy. The evolution of this support index should be recorded in the course of a subject’s therapy to point out changes in ability.

V. DISCUSSION AND CONCLUSIONS

First results with healthy subjects show that subjects can be effectively supported in the training of coordination of hand position and orientation. A combination of impedance and admittance control was successfully implemented on ARMin and ensures good performance. Being based on the minimum intervention principle, the control strategy maximizes the subject’s own contribution to the task. The implementation of ADLs has great potential to increase motivation and to improve the transfer of learning effects to daily living.

VI. ACKNOWLEDGMENTS

We thank Prof. Dr. med. Volker Dietz of the University Hospital Balgrist, Zürich, as well as Gabriela Kiefer and Christina Perndl for their contributions to this work.

**REFERENCES**


