Low level control in a semi-autonomous rehabilitation robotic system via a Brain-Computer Interface

Thorsten Lüth, Darko Ojdanić, Ola Friman, Oliver Prenzel, and Axel Gräser

Abstract — In this work, a connection between a semi-autonomous rehabilitation robotic system and Brain-Computer Interfaces (BCI) is described. This paper focuses on a system for user intervention in low-level movement control of an assistive robotic arm. The rehabilitation robotic system allows tetraplegics to control the system with high-level commands (e.g., “grab the bottle”), and then to intervene in the execution of the task, if they see that something is going wrong. In such a case, the user gets the opportunity to control the system with a low-level control of the robot arm. A system for such a control on a low abstraction level by a Brain-Computer Interface based on P300 and steady-state visual evoked potentials (SSVEP) will be described in this work.

I. INTRODUCTION

The FRIEND II rehabilitation robotic system is being developed to assist tetra-plegics in daily life tasks and thus to increase their autonomy. FRIEND II is an electrical wheelchair, which is equipped with a 7 degrees-of-freedom (DOF) manipulator as well as additional sensors and actuators, such as stereo-camera system mounted on a pan-tilt head and a smart tray with tactile surface and integrated scale (see Fig. 1). The control of the system usually takes place on a high abstraction level, e.g., with user commands like “Pour in a drink” or “Take out bottle from fridge” [1]. Subsequent to the user-initiated task selection on the abstract level, the system plans a sequence of operations that are required to solve a certain task. These operations organize the access of the system’s sensors and actuators, but if necessary, also involve the user into task execution, according to the principle of semi-autonomy [2].

During task execution, the manipulator has to be capable of performing various operations autonomously. Motion planning algorithms that smoothly drive the manipulator through the environment without collision with obstacles, are already in place [3]. However, due to the uncertainties present in the data coming from the sensors (cameras, artificial skin etc.), the estimated positions of objects can differ slightly from the true values. This can cause failures in the task execution, e.g., a grasping process. The user can then take over the control of the manipulator and adjust the gripper location. Moreover, the user controls the manipulator in an intuitive manner by using Cartesian commands, but the system controls the redundancy of manipulator in parallel (more details are provided in Section III). In this way it supports the user to utilize all the capabilities of the 7-DOF manipulator for more dexterous manipulation in cluttered environment. After the gripper adjustment by the user, the system can proceed with the execution of the remaining tasks as planned. In this paper, a solution that gives the user direct control of the robot arm with a brain computer interface is described.

Brain-Computer Interfaces provide control of software or hardware applications by interpreting patterns in the brain activity of the user [4]. The brain activity exists in different types that can be used in a Brain-Computer Interface (BCI). In this work two types of brain activity patterns are used: the P300 waveform and steady-state visual evoked potentials (SSVEP). The P300 waveform can be measured in the brain about 300 ms after a stimulus expected by the person, has been presented [5]. The SSVEP signal arises in the visual cortex when a person is looking at a continuously blinking light source [6]. These signals are in this work measured non-invasively with an electroencephalograph (EEG), i.e., with electrodes attached to the subject’s scalp.

The paper is organized as follows. In Section II it will be

Fig. 1. FRIEND II system controlled by a Brain-Computer Interface.
illustrated how the BCI-driven direct control of the manipulator smoothly integrates into the overall control concept. Section III describes the mathematical background for the manipulator direct control and motion planning, as well as how the system utilizes the redundancy in the direct control. The graphical user interface is presented in Section IV, and the P300 as well as the SSVEP is explained in detail in Section V. For experimental evaluation of the presented method a typical support task will be used in this work: Grasping of an object. The experimental results are presented in Section VI.

II. OVERALL CONTROL CONCEPT

The control architecture MASSIVE (Multi-layer architecture for semi-autonomous service-robots with verified task execution [7]) has been developed for service robotic systems like FRIEND II. An important characteristic of MASSIVE with relevance to this paper is its input device abstraction concept [8]. This enables the adaptation of any input media (Speech, Eye-mouse, BCI, etc.). Additional important features of MASSIVE are the operation on the basis of pre-structured task-knowledge as well as the semi-autonomous task execution. These design aspects are justified due to the application area of rehabilitation robotics.

Fig. 2 depicts the activity diagram of task planning and execution in MASSIVE. Initially, the task request from the user is received. According to the task selection, pre-structured task knowledge [7] is retrieved from the database and initialized within a semi-autonomous monitoring process [9]. Subsequently, task relevant information about the target objects to be manipulated as well as about obstacles is available in the system’s world model for further steps of task execution. After the initial task planning, as depicted in Fig. 2, autonomous operations are executed first, and in case of their failure, user interactions are initiated. A continuous planning process in the system’s execution layer controls the activation and deactivation of operations according to the return values of low-level operations [10].

III. MANIPULATOR MOTION

The option to control the manipulator directly with a BCI will be used only for small movements, e.g., for fine adjustments of the gripper location in case the sensors do not provide enough precise position information. In the example shown in Fig. 3, direct robot control is a part of the operation “FineAdjustToObject”. Other movements, e.g., moving the gripper to the proximity of the object, are controlled by motion planning algorithms, i.e., the operation “MoveToObject”.

Authorized licensed use limited to: STAATS U UNIBIBL BREMEN. Downloaded on August 5, 2009 at 04:34 from IEEE Xplore. Restrictions apply.
A. Motion planning

In contrast to the available methods that mainly use the configuration space, leading to large calculation times for robots with many DOFs, the motion planning approach used in the FRIEND II system is based on Cartesian space information. The motion planning algorithm and its implementation is described in [12]. The method operates fast, but since it is a local planner it is not able to avoid local minima (dead-ends) in the robot workspace. In order to solve the problem of local minima, a global planner is currently being developed which will be combined with the local planner and a trajectory smoothing procedure [3].

Although the primary objective of motion planning is to reach the goal location without collision, the redundancy of the robot should be also taken into account. That is, if the manipulator reaches the target location, its configuration should be as far as possible from the joint limits and obstacles. This makes the ensuing movements simpler, regardless of who executes them: the user or the system. Moreover, the object to be grasped often has the shape of a cylinder (bottle, glass, handles, etc.). This introduces additional degree of redundancy since the grasping of the object can be done from several directions.

A method that utilizes this redundancy is integrated into the local planner. First of all, additional grasping frames are calculated by incrementally rotating the given goal frame about the object axis. The frames are determined as depicted in Fig. 4, where the angle \( \varphi \) is a predefined parameter. In this example the axes of the cylinder are parallel to the z-axis of the world coordinate system, meaning that the object is placed vertically on a platform. The corresponding matrix calculation for additional goals is given by the equation

\[
T_{gi} = T_O \cdot Rot(z,i\varphi) \cdot T_O^{-1} \cdot T_g
\]

where \( T_O \) is the object frame, \( T_g \) is the gripper goal frame, and \( i = \pm 1, \pm 2, \ldots, \pm N \). These additional frames are also considered in motion planning. Therefore, if it happens that the given goal frame is not reachable due to the joint limits or obstacles, other frames might be reachable and the manipulator will be able to approach the target object.

B. Direct control

The direct control of the manipulator or robot arm can be performed either in the workspace coordinate systems or the configuration space. The workspace represents the Cartesian space in which the robot moves. The configuration space is the set of all possible configurations of a manipulator, where one configuration represents the set of joint values. Operating the robot in the configuration space can be confusing to the user and for fine movements it is common practice to use the workspace coordinate system for direct control.

1) Workspace commands

There are two different coordinate systems usually used for manipulators: the World and the Gripper coordinate systems. The World coordinate system has a fixed coordinate origin and axes that are usually placed at the base of the manipulator. The Gripper coordinate system is attached to the gripper (see Fig. 7). In some situations the manipulator is easier controlled in the World coordinate system and in other situations the Gripper coordinate system is preferred. In this work, both options are implemented, so that the user can easily switch between them. Determining which coordinate system is the more suitable for control strongly depends on the user’s preferences, as well as on the particular task. A typical example where the use of the Gripper coordinate system is preferred, is the case of grasping an object when the gripper is located directly in front of the object. The user would like to perform a straight-line motion in respect to the gripper, which corresponds to the z-axes of the Gripper coordinate system. On the other hand, the FRIEND II system has the gripper with the fingers slightly directed to the side (imitating the human hand, see Fig. 5). This characteristic can sometimes create a situation where the World coordinate system is preferred.

A 3D workspace is fully described with six coordinates. Three coordinates define the translation (x, y, z) and other three the rotational part of the manipulator. Rotation is typically described with the roll, pitch and yaw angles, representing the rotations about the coordinate axes. As is usual in robotics, the homogenous transformation matrices (4x4) are used for describing the transformation between coordinate...
system [11]. The transformation of the gripper coordinate system with respect to the world coordinate system will here be denoted gripper frame. Suppose that the user controls the translation of the gripper (the dialog is presented in Fig. 7) and he selects x-axes with the step size s, the corresponding transformation matrix can then be written:

$$T_t = \text{Trans}(d \cdot s, 0, 0) = \begin{bmatrix} 1 & 0 & 0 & d \cdot s \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where d defines the selected direction of the motion (±1). If the user operates in a Gripper coordinate system, the next goal frame for the gripper is

$$T_{\text{Gnew}} = T_G \cdot T_t$$

The matrix $T_G$ represents the current gripper frame. In case the user is operating in the World coordinate system, the transformations have to be multiplied from left to right, which results in the different goal frame

$$T_{\text{Gnew}} = T_T \cdot T_G$$

If the user wants to perform a rotation in the Gripper coordinate system, for example about the y-axes for the angular step size $\beta_y$ in the direction d, the corresponding goal frame is calculated as

$$T_{\text{Gnew}} = T_G \cdot T_{\text{Rot}} = T_G \cdot \text{Rot}(y, d \cdot \beta_y) =
\begin{bmatrix}
\cos(d \cdot \beta_y) & 0 & \sin(d \cdot \beta_y) & 0 \\
0 & 1 & 0 & 0 \
-\sin(d \cdot \beta_y) & 0 & \cos(d \cdot \beta_y) & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}$$

The goal frame for the rotation about the other axes is calculated in similar ways, but of course with the appropriate matrix $T_{\text{Rot}}$. The resulting frame for the World coordinate system is calculated analogously to the transformation example above. The velocity of the motion is in this work predefined and constant. During the experiment phase this has been proven appropriate. If further investigations show that different velocities are required, the interaction dialog could be expanded with one more control input.

2) Utilizing redundancy

After the goal frame is defined, the inverse kinematics solution is calculated. In other words, the joint angles that correspond to the desired gripper location are determined. Since the manipulator of the FRIEND II system is redundant there is no unique solution to the inverse kinematics problem. The set of configurations that match a gripper location form a motion of the elbow on the so-called redundancy circle (see Fig. 6). The inverse kinematics solution for the 7 DOF manipulator developed at the Institute of Automation at the University of Bremen [1] can efficiently calculate the resulting configuration for a given angle on the redundancy circle.

The user will use the direct control only for some small movements. Hence, it is reasonable to keep the angle on the redundancy circle at the same value. However, in some particular situations it can happen that for a given angle on the redundancy circle, the resulting configuration lies outside the joint limits. In such a case, the manipulator is not able to reach the desired location. Using a slightly modified redundancy angle, e.g., by adding a small additional movement of the elbow, could produce a valid configuration. This is a typical example illustrating the advantage of using redundant manipulators as the redundancy provides more alternatives for the movement.

![Fig. 6 Movement of the elbow on the redundancy circle. $\alpha$ is the angle on the redundancy circle corresponding to one configuration.](image)

In the direct control of the manipulator using a BCI presented in this work, the system is continuously supporting the user by utilizing the redundancy of the robot arm. When a new configuration is not valid, the joint values are too close to the joint limits, or the minimal distance to obstacles is too small, the system will scan the redundancy circle to find out if there is a better configuration available. This approach enables the user to control the manipulator in an intuitive manner while the system supports him by fully utilizing the capabilities of the manipulator.

IV. GRAPHICAL USER INTERFACE

According to Section II, if an autonomous high level operation fails or the user interrupts ongoing wrong operations, user interactions are initiated (see Fig. 2). In the following, the graphical user interface for direct control of the robot arm is presented, which realizes the manipulative user interaction in case of user interruption of an erroneous autonomous manipulation. How the user selects an option in the interface is explained in Section V.

The main window of the interface, the user is working with, will be the coordinate control. So the first part of the
whole graphical user interface presented to the user is the Manipulation window (Fig. 7). The second window (Settings - Fig. 8) will only be visible, when the user wants to change the settings like the coordinate system or the kind of movement (translation or rotation), or when the user wants to end the direct control.

A. Manipulation window

The main graphical interface presented to the user consists of five areas (numbered in Fig. 7). The “Manipulation” area (1) and the “Step” area (2) are dynamic areas, according to the settings the user can change in the Settings window.

The initial configuration of the Manipulation window is dependent on the task context (see Section III.B.1) and on the user preferences. According to the overall control concept as described in Section II the context for the task is given in the system. For the example of fine adjustments of the manipulator in case of grasping an object (see Fig. 3 – MoveToObject), the initial configuration is the World coordinate system (see Fig. 8) and the “X” coordinate. This configuration is set according to experimental results in this work (see Section VI). An image, with the actual coordinate system drawn, is presented to the user in the upper right corner. By choosing “Plus” (3) or “Minus” (4) the user can add a positive or negative step to the selected “Manipulate” variable in the actual coordinate system.

If the user wants to change the manipulation settings, he has to go to the control settings interface, by selecting “Settings” (5). According to the settings the user made in that dialog (see Fig. 8), the appearance of the Manipulation window changes. If the user chose “Translation” in the Settings window, the Manipulation window contains the “X”, “Y”, and “Z” coordinates and the “Step” in “cm”. On the other hand, if the user chose “Rotation” in the Settings window, the Manipulation window contains “Yaw”, “Pitch”, and “Roll” as well as the “Step” in “degrees”. After finishing the direct control, the user has to go to the Settings window to end the direct control.

B. Settings window

After selecting “Settings” in the Manipulation window, the user gets to the Settings window (Fig. 8). This window is also consisting of five areas, the user can select. The upper part of the interface is for the preparation of the direct robot control. The user can change different settings, i.e. which kind of movement he wants to perform (translation or rotation – 1) and in which coordinate system the manipulation shall be made (Gripper or World coordinate system – 2). The coordinate system the user chose is presented to him as an image in the upper right corner. After these settings, the user can go back to the direct control interface (see Fig. 7) by selecting the “Back to Control” area (5). When the user has finished the direct control, he has to decide if he was able to move the robot in the right way (3) or not (4). On a success the robot is performing the rest of the task autonomously and on a failure the system aborts the task completely.

V. Brain-Computer Interface

A Brain-Computer Interface is realized with EEG equipment and software for analyzing the EEG signals in real time. For the current work, the biosignal amplifier USBamp from g.tec is used to amplify the brain signals acquired using EEG gold electrodes. The acquired signals are digitized with a sampling rate of 128 Hz and further processed on a regular laptop computer. Custom written software for signal analysis and classification has been developed in Visual C++. This software can detect either the P300 signal or the SSVEP response. More information on these brain signal patterns and their detection is given below.
A. P300-based BCI

The P300 response is a post-stimulus waveform occurring in the brain about 300 ms after the presentation of an anticipated stimulus [13],[14]. The temporal shape and spatial distribution of the P300 response are shown in Fig. 9. Spatially, the P300 response is found in the parietal lobe, i.e., at top of the head. That is, to detect the P300 response, the EEG electrodes must be placed over this area. Since the shape of the P300 signals is approximately known it can be detected with a matched filter. In the bottom panel of Fig. 9 (bold line) the matched filter is parameterized as a sum of three Gaussian bumps with different amplitudes and temporal shifts:

\[ h(t) = A_1 e^{\frac{(t-\tau)^2}{\sigma^2}} + A_2 e^{\frac{(t-\tau)^2}{\sigma^2}} + A_3 e^{\frac{(t-\tau)^2}{\sigma^2}} \]  \hspace{1cm} (6)

These parameters can be optimized to match the P300 response to individual users as well as possible. The P300-based BCI now works as follows. The background frames of the 5 different fields of the dialogs in Fig. 7 and Fig. 8 are highlighted in random order. The fields are highlighted by changing the background of the field to a red color for 100 ms. The time between two highlights is 300 ms. The user is focusing the attention on the field containing the command he or she wants to execute. Every time this field is highlighted, a P300 response is elicited. Thus, when the matched filter detection procedure indicates the presence of a P300 response in the EEG signals (by convolving the matched filter with the EEG signals), one can infer that the user wants to execute the command that was highlighted 300 ms prior to the response detection. Since the P300 response is weak and the EEG signals noisy, each field must be highlighted several times (5-10 times depending on the user) before a reliable detection can be made [15]-[18].

B. SSVEP-based BCI

When a person is looking on a light source blinking or flickering with a certain frequency, this frequency is also reflected in the neuronal activity in the visual cortex. This is known as the Steady State Visual Evoked Potential (SSVEP) response, and it can be detected as sharp frequency peaks in the EEG signals [6]. Therefore, an SSVEP-based selection can be realized by having an array of light sources where each light source is flickering with a distinct frequency. Each light source generates different frequency peaks in the EEG signals and one can therefore classify which light source the person is looking at by analyzing the frequency content, which usually is done via a Fast Fourier Transform. In [19], an algorithm for detecting the SSVEP response in multiple EEG signals acquired at different scalp positions was presented. This algorithm is used in this work for detecting the SSVEP signals. To operate the dialogs in Section IV, five different frequencies are needed to give the user the opportunity to choose all of possible selections. A custom made array with Light Emitting Diodes is used for this purpose, see Fig. 10.

The diodes in this array are flickering with the frequencies 13, 14, 15, 16 and 17 Hz. Each diode encodes a certain command in the dialog windows, as described in Table I. That is, when a peak at one of the above frequencies is detected in the EEG signals one can conclude that the user is looking at the diode flickering with this frequency, and the associated command is issued.

<table>
<thead>
<tr>
<th>SSVEP-FREQUENCIES AND CORRESPONDING SELECTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corresponding selection in</td>
</tr>
<tr>
<td>Diode Frequency</td>
</tr>
<tr>
<td>-----------------</td>
</tr>
<tr>
<td>1 13 Hz</td>
</tr>
<tr>
<td>2 14 Hz</td>
</tr>
<tr>
<td>3 15 Hz</td>
</tr>
<tr>
<td>4 16 Hz</td>
</tr>
<tr>
<td>5 17 Hz</td>
</tr>
</tbody>
</table>
VI. RESULTS

The goal in this work is to allow the user to adjust the position of the manipulator and gripper in the FRIEND II system via a BCI. To optimize this procedure, a number of questions need to be investigated. For example, what coordinate system should be used as default (World or Gripper)? What precision is required? How many adjustment steps are needed? These questions are examined below. There are also questions relating to the BCI, for example, how long does it take to make one decision and what is the risk of interpreting the brain patterns erroneously? These questions have been examined in detail elsewhere, see for example [19],[20]. In general, the SSVEP-based BCI is the more robust BCI with a false-positive classification rate usually under 5% and with the ability to issue commands every 2-4 seconds, see [19],[21],[22]. The P300 BCI is slower but on the other hand useful for a larger group of people since no eye-movements are required. The P300 BCI also integrates better in the graphical user interface. Upon acceptance, an on-site demonstration of the BCI and direct control of the FRIEND II system will be given at the ICORR'07 conference.

A. Experimental setup

8 healthy persons (1 female) in the ages between 20 and 35 years were given the task to position the gripper and grab a bottle using the dialog in Fig. 7. Thus, the user had to select along which axis to move the robot and with which step length (4, 2 or 1 cm). The total number of selections required (i.e., number of clicks in the dialog) to place the gripper into the correct position was counted. This task was carried out for 4 different starting positions of the gripper, all about 10 to 15 cm from the target bottle, see Fig. 5. Each task was also performed using both the World coordinate system and the Gripper coordinate system. To focus this experiment on the difficulties in controlling the robot arm directly, the users used a computer mouse for the selections instead of the BCI. The time for making a selection is therefore assumed negligible and the variability that would come from different selection times and errors for different persons with the BCI is avoided. As mentioned above, the selection times using the BCI have been studied in detail elsewhere.

B. Number of selections

In Table II, the number of selections the users made to grab the bottle, averaged over all four starting positions of the gripper, is reported. The selections for the Gripper coordinate system (G) and World coordinate system (W) are reported separately. Most of the subjects had to make more selections in the Gripper coordinate system (statistical significance p=0.025 with a Student's t-test). Since the users had no prior experience with this task, it can be assumed that fewer selections will be required after some practice. An important observation was also that the available step lengths seem to be adequate; a finer resolution than 1 cm was not required to grab the bottle and step lengths of more 4 cm are difficult to handle for the user. Depending on the user, the execution of one selection takes around 2 to 5 seconds with the BCI. Thus, based on the numbers in Table II, the total time for grabbing the bottle can be estimated to between 30 seconds to 2 minutes plus the time to consider the moves and selections (i.e., along which axis the next step should be and how long it should be). In the experiment, it was found that this latter deliberation time generally far exceed the time for actually carrying out the selections with a BCI.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>20.25</td>
<td>17.75</td>
<td>21.50</td>
<td>15.75</td>
<td>17.50</td>
<td>23.00</td>
<td>24.00</td>
<td>20.00</td>
</tr>
<tr>
<td>W</td>
<td>20.00</td>
<td>20.50</td>
<td>16.75</td>
<td>11.75</td>
<td>14.75</td>
<td>16.25</td>
<td>18.50</td>
<td>15.75</td>
</tr>
</tbody>
</table>

TABLE II

MEAN VALUES OF SELECTIONS THE SUBJECTS HAD TO MAKE TO EXECUTE THE TASKS IN GRIPPER (G) AND WORLD (W) COORDINATE SYSTEM

VII. DISCUSSION & CONCLUSION

The presented material aims to bridge two lines of work that hitherto have been developed independently; on the one hand the autonomous robot rehabilitation system FRIEND II and the software structure MASSIVE, on the other hand the Brain-Computer Interface technology. The application chosen for this is direct control of the manipulator in the FRIEND II system using a Brain-Computer Interface. This functionality is now also supported in the MASSIVE software structure. A number of situations where the user can or must operate the gripper directly can be foreseen. For example, a high level gripping task can be designed so that the autonomous motion planning drives the gripper to an approximately correct position and then hands over the fine adjustments to the user. Another option is that the system itself detects that it needs guidance from the user, e.g., if the motion planning fails. Yet another possibility is that the user stops the system and takes over the control if he or she recognizes that something is about to go wrong. The possibility to resolve simple problems is important to the overall goal of providing independence to the handicapped user.

Using a BCI for the direct control of the gripper makes the system potentially useful for people with a wide range of handicaps. However, at the present time, the BCI solution should be seen as an alternative only for people with severe disabilities. For example, users who can operate an SSVEP-based BCI can presumably also operate an eye-tracker. The relative merits of these two alternatives have still to be evaluated. The P300 BCI also proposed in this work does not rely on the user’s ability to move the eyes, and it is therefore in this respect a more general solution. It also integrates more seamlessly into system because the external stimulus is realized as highlights directly in the dialogs shown on the computer monitor. This is not easily done with the repetitive SSVEP stimulus since it then must be synchronized with the update frequency of the monitor. The array of external light
emitting diodes shown in Fig. 10 is for this reason employed in this work.

The optimal integration of the BCI with the FRIEND II system is still being investigated. The results presented in the Results section indicate that the users prefer controlling the robot in the World coordinate system, at least when they are not trained for the gripping task. A precision of 1 cm steps was also found sufficient. There are many more questions that require investigation. For example, how is the computer screen or the LED array placed in relation to the robot workspace so that the user can see both at the same time? Are the dialogs presented in Fig. 7 and Fig. 8 optimal with respect to the uncertainty inherent in BCI, i.e., to the risk of interpreting the brain patterns erroneously and sending the wrong command? Finally, and most importantly, is a disabled user able to use the system? So far it has only been tested on healthy volunteers.

REFERENCES