Navigating a Smart Wheelchair with a Brain-Computer Interface
Interpreting Steady-State Visual Evoked Potentials

Christian Mandel, Thorsten Lüth, Tim Laue, Thomas Röfer, Axel Gräser, and Bernd Krieg-Brückner

Abstract—In order to allow severely disabled people who cannot move their arms and legs to steer an automated wheelchair, this work proposes the combination of a non-invasive EEG-based human-robot interface and an autonomous navigation system that safely executes the issued commands. The robust classification of steady-state visual evoked potentials in brain activity allows for the seamless projection of qualitative directional navigation commands onto a frequently updated route graph representation of the environment. The deduced metrical target locations are navigated to by the application of an extended version of the well-established Nearness Diagram Navigation method. The applicability of the system proposed is demonstrated by a real-world pilot study in which eight out of nine untrained subjects successfully navigated an automated wheelchair, requiring only some ten minutes of preparation.

I. INTRODUCTION

People who are severely disabled, quadriplegics in particular, may not be able to comfortably control an electric wheelchair and are thus confined to a push-chair, relying on external help. Here, the research area of human-robot interaction (HRI) can help by providing sophisticated interface techniques. General literature overviews [1], [2] list at least 45 research projects that aim at the development of smart wheelchairs. Many of these projects support specialized input methods that allow the paraplegic to control his or her vehicle without a standard joystick. Common approaches are Natural Language Communication [3], [4], Head Pose Interpretation [5], [6], and recently the application of Brain-Computer Interfaces (BCI).

A BCI system analyzes specific patterns in the user’s brain activity and translates them into commands to control soft- or hardware devices [7]. The principal goal of BCI research is to provide severely disabled people a form of communication which varies from spelling applications [8], [9] to complex rehabilitation systems [10], or prostheses [11]. Navigating a wheelchair with the help of a BCI is a recently grown area in the BCI community. Starting from the idea of manipulating smart environments [12], a person can move a wheelchair to a predefined goal position by using a BCI [13]. But also a low-level navigation of a wheelchair is possible with a limited number of commands [14]. The connection of an SSVEP (steady-state visual evoked potential) based BCI that supports many more commands (13 commands reported in [15]) to a wheelchair has not yet been demonstrated. This work describes the control of the Bremen Autonomous Wheelchair Rolland using the SSVEP-based BremenBCI.

The remainder of the paper is structured as follows: Section II gives an overview of the system’s components, including the hardware modules and the software modules involved. Section III continues with the physiological background of steady-state visual evoked potentials and the EEG signal processing structure to handle nuisance components in EEG data and to classify the desired frequency. Section IV follows with a presentation of the Voronoi graph as the fundamental spatial representation for describing the environment and its embedded navigable paths. Afterwards, Sections V and VI illustrate the path selection scheme applied as well as the extended version of the Nearness Diagram Navigation approach employed. Finally, Section VII presents the results and a discussion of an experimental pilot study that supports the essential ideas of this work.

II. SYSTEM OVERVIEW

With an electrical wheelchair that comprises autonomous navigation capabilities, and an EEG-based HRI that interprets
SSVEPs, this work introduces a complete navigation solution for the paraplegic.

**A. The Wheelchair**

Serving as the experimental platform, the Bremen Autonomous Wheelchair Rolland (cf. Fig. 1) is based on the battery-powered wheelchair Meyra Champ 1.594. Rolland has a differential drive in the back and passive castor wheels in the front. It is equipped with two laser range finders that sense nearby obstacles in a height of about 12cm. The system also provides two incremental encoders which measure the rotational velocities of the two independently actuated rear wheels, allowing for odometric pose estimations. The software architecture used to control Rolland is based on the framework of the GermanTeam [16]. In its version for Rolland, the system embeds all navigational software modules necessary, ranging from the acquisition and maintenance of local and global spatial representations, to high-level path selection, and low-level obstacle avoidance.

**B. The Safety Layer**

Since the navigation method employed (cf. Section VI) is heuristic in nature, it cannot guarantee collision-free motion. Hence, the wheelchair is equipped with a safety layer that ensures that the vehicle will stop in time before a collision can occur. 30 times per second, the safety layer makes a binary decision. Either the current driving command is safe, and it can be sent to the wheelchair, or it is not, and the wheelchair has to stop instead. “Safe” means that if a stop command would be initiated in the next processing cycle (i.e. 33 ms later), the wheelchair would still be able to stop without a collision. Otherwise, it has to be stopped in this cycle, because in the next cycle it would be too late. Whether the wheelchair can stop in time depends on the actual speeds of the two drive wheels and the current drive command, because it will influence the current speeds in the future, the shape of the wheelchair, and its current surroundings. The surroundings are measured using the laser scanners and a model of the environment is maintained in a local obstacle map (cf. Section IV), treating stationary and moving obstacles as equal. Based on the current speeds and the commanded speeds, a safety area is searched for obstacles in the map. If the safety area is free of obstacles, the current driving command is safe. Since the shape of such a safety area is rather complex, a large number of safety areas were pre-computed and stored in a lookup table. Two such safety areas are shown in Fig. 3 (the wheelchair turns to the left) and in Fig. 4(b) (the wheelchair drives straight ahead).

**C. The BCI System**

The stimulus for the SSVEP-based BremenBCI is implemented as an LED-panel of four different diodes oscillating with different frequencies. These diodes relate to the commands chosen to navigate the wheelchair (13 Hz = left, 14 Hz = right, 15 Hz = front and 16 Hz = back). A given command is not interpreted as an ongoing movement waiting for a stop command, i.e. the wheelchair stops itself after reaching a nearby target position. Therefore and because of the safety layer, an explicit stop command is not required. The software structure of the BCI corresponds to the usual signal processing structure of data acquisition, preprocessing, feature extraction, classification, and application interface. For data acquisition, an electroencephalography (EEG) cap with non-invasive electrodes placed on the scalp is used. The signal is amplified with a gUSBamp biosignal amplifier from g.tec already equipped with an analog-to-digital converter. The TCP/IP protocol is used as the application interface.

**III. STEADY-STATE VISUAL EVOKED POTENTIALS**

Electrical potential changes in brain activity due to an external stimulus are known as evoked potentials (EP) and can be observed in the sensor cortex of the brain. Steady-state visual evoked potentials (SSVEP) are periodic components of the same frequency as a continuously blinking visual stimulus (higher than 4 Hz), as well as a number of harmonic frequencies that can be obtained in brain activity in the visual cortex, when a person is focusing on that stimulus [17], [18]. The SSVEP response can be detected quite robustly, because its characteristic varies from spontaneous brain activity. The strongest response is measurable for stimulation frequencies in the range of 5 - 20 Hz [19]. Figure 2 shows a typical SSVEP response in the visual cortex for a test person focusing on a flickering stimulus of 9 Hz. Peaks at the fundamental frequency as well as two harmonics are observable.

**A. Combining Electrode Signals into Channel Signals**

We consider visual stimulation with a flicker frequency of $fHz$. If a person is focusing attention on that stimulus, the SSVEP response in the EEG signal measured as the voltage between a reference electrode and the electrode number $i$ can be estimated as:

$$
y_i(t) = \sum_{k=1}^{k=N_h} (a_{i,k}\sin(2\pi ft + \Phi_{i,k})) + b(t) \quad (1)
$$

where $0 \leq t < TS$, $b$ describes the noise, $TS$ is the time segment, $N_h$ is a number of harmonics. Each sinusoid on each electrode has its own amplitude ($a_{i,k}$) and phase ($\Phi_{i,k}$).

The nuisance signals $b$ can have several origins: the environment and its effect on the subject, a natural physical
disturbance such as other brain processes, and noise of each electrode on the cap. Therefore, one goal is to magnify the SSVEP response and to decrease the noise to improve the detection of the desired frequency.

A channel $s$ is used as a linear combination of the signals measured by the $N_y$ electrodes. With $0 \leq l < N_x$ and $N_x$ is the number of channels, $s$ is defined by:

$$ s_l(t) = \sum_{i=1}^{N_y} w_{i,l} y_i(t) \quad (2) $$

For a channel, the information from the electrodes is resumed in a single scalar at a time $t$. For the EEG signal processing, the first goal is to find an optimal set $w_{i,l}, 1 \leq i < N_y$.

For the creation of a single or several channel signals, the minimum energy combination [20] is used in this paper. This method allows the combination not only of pairs or groups of electrodes but also of an arbitrary number of electrodes that cancel the noise as much as possible. The advantage of this method is that the number of electrodes does not need to be chosen beforehand. Its good performance was validated in different applications [21], [22].

B. Feature Extraction and Frequency Classification

For the detection of the stimulus frequencies in the acquired data, the total power $P$ at the SSVEP frequency is estimated. SSVEP stimulation frequencies and its harmonics do not always coincide with the Discrete Fourier Transform (DFT). Therefore, slightly different to the squared DFT magnitude, a more general formula that can estimate the power in the $k$th SSVEP harmonic frequency in the $l$th channel signal $s_l$. With $X_k$ is an SSVEP model (excluding the noise) for each harmonic frequency according to equation 1, the power is estimated to:

$$ \hat{P}_{k,l} = \| X_k^T s_l \|^2 \quad (3) $$

After the power of each stimulus frequency is calculated in the acquired brain activity, we use a linear classifier to classify the frequency the subject is focusing on. To consider a stimulus frequency as the desired one and therefore to generate a new command, the corresponding power of that frequency has to exceed a threshold. If more than one power exceeds the threshold, the frequency with the highest power is classified.

IV. REPRESENTING SPATIAL ENVIRONMENTS: ROUTE GRAPHS

The basic representation of the environment, the so-called evidence grid [23], is a two-dimensional array of cells each of which stores the evidence that the corresponding location in the environment is occupied by an obstacle. The current implementation maintains a $7.5 \times 7.5m^2$ grid out of $300 \times 300$ cells resulting in a spatial resolution of $2.5 \times 2.5cm^2$. In definition (4) $EGC(x,y) = 0$ denotes a surely unoccupied, and $EGC(x,y) = 1$ a surely occupied cell respectively.

$$ EGC : \mathbb{N} \times \mathbb{N} \rightarrow \{0...1\} \quad (4) $$

The so-called distance grid is derived from the evidence grid and contains the distance to the closest obstacle for each cell. It is calculated by a fast double sweep-line algorithm [24] that computes for each free cell the metric distance to the next occupied cell. Formally it consists of cells as defined in (5), where $c$ is the resolution of the grid, i.e. 2.5cm.

$$ DGC : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{R} $$

$$ DGC(x,y) = \min_{x'\,y' : EGC(x',y') \geq 0.5} c \left| \left( x-x' \right) \left| y-y' \right| \right| \quad (5) $$

The final stride in the line of spatial representations is an instantiation of the route graph concept [25], [26]. A route graph is a multi-layered and graph-structured representation of the environment in which each graph layer describes the workspace on a different level of abstraction. For this work the route graph comprises a single graph layer, the so-called Voronoi graph $V\mathcal{G}$ (7). Its construction is directly based on the distance grid. In a first step, the algorithm computes for every $DGC(x,y)$ whether the distance between two of its generating points, e.g. the occupied cells $EGC(x',y')$ and $EGC(x'',y'')$ that gave $DGC(x,y)$ its value, is greater than a given threshold $\epsilon$. In the formal definition of the resulting Voronoi diagram $VD$ (6), the constant value $\epsilon$ determines the minimal free space that is required to mark a region as navigable. We use $\epsilon = 70cm$, i.e. the wheelchair’s maximal width plus 6cm. Note that the evidence grid, the distance grid, and the voronoi diagram are updated at the same frequency as the safety layer, i.e. every 33ms.

$$ VD \models \begin{cases} x, y \in \mathbb{N}, x', y', x'', y'' \in \mathbb{N} : \\
EGC(x',y') > 0.5 \land \\
EGC(x'',y'') > 0.5 \land \\
d_{x',y'}^{x'',y''} > \epsilon \land \\
\frac{d_{x',y'}^{x'',y''}}{d_{x',y'}^{x',y''}} = DGC(x,y) \end{cases} \quad (6) $$

where $d_{x',y'}^{x'',y''} = c \left( \left| x - x' \right| \left| y - y' \right| \right)$. The second step searches the Voronoi diagram $VD$ for elements that hold one or more than two neighbors in $VD$. These cells correspond to terminating or branching nodes respectively, and they are inserted into the Voronoi graph’s set of nodes $V\mathcal{G}, N$. The Voronoi graph’s set of edges comprises pairs of references to elements in $V\mathcal{G}, N$ that are connected by points out of $VD$.

$$ N \subset \{(x,y) | x, y \in \mathbb{R}\} $$

$$ E \subset \{(n^x, n^y) | n^x, n^y \in N, n^x \neq n^y\} \quad (7) $$

V. INTERPRETING QUALITATIVE NAVIGATION COMMANDS ON ROUTE GRAPHS

The interpretation of a qualitative driving command such as go left basically asks for an adequate projection of the given direction onto the spatial knowledge of the robot. As a primary source of information, this work applies the Voronoi graph $V\mathcal{G}$ for the extraction of the set of navigable paths $NP(8)$. In a second step, each path $np \in NP$ is evaluated against the command given, resulting in the best matching path to be forwarded to the local navigation module.
Fig. 3. Assuming a given command $BCI_{cmd} = \text{right}$, represented as the vector from on to cmd, the algorithm for the interpretation of qualitative navigation commands (cf. Section V for details) basically evaluates the angle between the vectors $\vec{cmd} - \vec{on}$ and $\vec{np}_j - \vec{on}$.

$$
NP \subset \left\{ np = (e^1, \ldots, e^i) \mid i \in \mathbb{N}, \ e^1, \ldots, e^i \in E, \ e^i.n^g = e^{i+1}.n^s \land e^i.n^g \neq e^j.n^s \forall i \geq j \right\} 
$$

In order to derive the set of navigable paths $NP$ from a given Voronoi graph $VG$, its set of nodes $VG.N$ is first enriched by the node $on$, representing the current odometry position. Furthermore, $VG.E$ is augmented by edges that connect $on$ with all nodes lying in a given circumference of $on$.

The computation of $NP$ continues by applying an A*-algorithm [27] that searches $VG$ for all paths that connect $on$ to all other nodes in $VG.N$, i.e., all possible target positions. The resulting set of paths $NP$ is then filtered for paths the goal of which is not included as an interim node of any other path in $NP$, with the only exception of paths ending in a branching node with more than two incoming edges. This process leaves $NP$ holding only paths that lead to spatially emphasized target nodes, however covering all in-between goals by the way.

Evaluating each of the navigable paths in $NP$ against a qualitative directional command is a special case of the interpretation of common spatial relations against a given route graph. First described in the context of interpreting coarse verbal route descriptions [4], a qualitative directional command can be given in four different levels of granularity. Due to the limited set of available channels coming from the BCI, i.e., four, we apply a four-valued directional system holding the commands $BCI_{cmd} \in \{ \text{front}, \text{right}, \text{back}, \text{left} \}$.

$$
\alpha = \arctan2 \left( \frac{np_j.e_i.n^g.y - on.y}{np_j.e_i.n^g.x - on.x} \right) \mid j \in \{1..|NP|\} 
$$

$$
\beta = \begin{cases} 
0 & : BCI_{cmd} = \text{front} \\
\frac{\pi}{2} & : BCI_{cmd} = \text{right} \\
\pi & : BCI_{cmd} = \text{back} \\
\frac{3\pi}{2} & : BCI_{cmd} = \text{left} 
\end{cases} 
$$

$$
\text{score}(np_j) = e^{-\frac{1}{2}(\frac{\alpha - \beta}{\Delta\alpha})^2} 
$$

The final computation of a single path’s score is done in two steps. We start by computing the angle $\alpha$ between i) the vector that connects the odometry node $on$ with the goal node $np_j.e_i.n^g$, and ii) the vector that is based in $on$ and aligned to the current heading $\theta$ of the wheelchair. Taking the most compatible angle $\beta$ for the given command $BCI_{cmd}$, which is $\text{right} = 3/2\pi$ in the illustrative example in Fig. 3, and the normalizing constant $c$, we can now compute $\text{score}(np_j)$ as can be seen in (9). With the straightforward maximization of $\text{score}(np_j)$ over all $j \in \{1..|NP|\}$ the algorithm outputs the best matching path, the final node of which is to be forwarded to the local navigation module.

An alternative path selection scheme has been used in [28]. The principle idea is to iterate over the ordered sequence of each navigable path’s edges, and to assess the angles between all pairs of consecutive sections. In the case of evaluating a $BCI_{cmd} = \text{front}$, a single path’s score is formulated as the product over all intermediate scores that state how good two consecutive edges are aligned to each other. The treatment of commands that introduce a bending maneuver, e.g., $BCI_{cmd} = \text{right}$, is different because of the additional necessity to choose the best suiting node of a given path for the triggering directional command. The actual approach iterates over all nodes of a path to be evaluated, and calculates for each possible branching node the path’s score as the product of scores that arise from the evaluation of $BCI_{cmd} = \text{front}$ between all pairs of consecutive edges. The only exception is given by the two sections connected to the selected branching node, where the score’s factor is determined by the assessment of the angle between the two edges w.r.t. the given command.

**VI. LOCAL NAVIGATION APPROACH: NEARNESS DIAGRAM NAVIGATION**

Within this work we employ the Nearness Diagram Navigation (NDN) approach by Minguez et al. [29], in order to transfer Rolland from its current position to a nearby target position, while avoiding static and dynamic obstacles. The NDN-approach describes the sensor measurements of the visible part of the environment along with the desired target position $p_q = (x_g, y_g)$ as a unique element of an exclusive and complete set of situations. Each of the five situations originally defined is associated with a specific control law that determines the translational and rotational speed to be applied, as well as the desired direction of movement. In order to define a situation, the workspace is divided into sectors centered at the wheelchair’s origin. By maintaining a list of distances to the closest obstacles\(^1\) in each of the typically $2^5$ wide sectors, i.e., the nearness diagram (cf. Fig. 4(a)), the system is able to compute free regions in-between two consecutive gaps of the nearness diagram. Finally, a navigable region closest to the goal location is selected.

\(^1\)Actually the NDN-approach not only maintains a list of distances between the surrounding obstacles and the robot’s bounding-polygon, but also a list of distances between obstacles and the so-called safety-zone. The safety-zone itself is defined by the bounding-polygon plus the constant safety margin $D_s$. 

1121
This so-called free walking area along with its determining obstacles define a specific situation.

In the following we will informally survey necessary conditions for the five basic situations (cf. [30] for a broad analysis). Abbreviations, constant- and variable declarations, as well as the resulting control laws can be found in Table I.

<table>
<thead>
<tr>
<th>Situation</th>
<th>Direction Of Movement ( s_g )</th>
<th>Sheer Out Extensions ( s' )</th>
<th>Translational Speed ( v )</th>
<th>Rotational Speed ( \omega )</th>
</tr>
</thead>
<tbody>
<tr>
<td>HSGR</td>
<td>( s_{goal} )</td>
<td>( s_{goal} \pm s_{max} )</td>
<td>( v_{max} )</td>
<td>( \frac{\pi - 2</td>
</tr>
<tr>
<td>HSWR</td>
<td>( s_{goal} \pm \frac{s_{max}}{2} )</td>
<td>( s_{goal} \pm \frac{s_{max}}{2} )</td>
<td>( - )</td>
<td>( \omega_{max} )</td>
</tr>
<tr>
<td>HSNR</td>
<td>( s_{goal} \pm \frac{s_{max}}{2} )</td>
<td>( s_{goal} \pm \frac{s_{max}}{2} )</td>
<td>( - )</td>
<td>( \omega_{max} )</td>
</tr>
<tr>
<td>LS1</td>
<td>( s_{goal} \pm \frac{s_{max}}{2} )</td>
<td>( s_{goal} \pm \frac{s_{max}}{2} )</td>
<td>( - )</td>
<td>( \omega_{max} )</td>
</tr>
<tr>
<td>LS2</td>
<td>( s_{goal} \pm \frac{s_{max}}{2} )</td>
<td>( s_{goal} \pm \frac{s_{max}}{2} )</td>
<td>( - )</td>
<td>( \omega_{max} )</td>
</tr>
</tbody>
</table>

\( s_g \) = ND-sector containing the desired direction of movement. 
\( s_{goal} \) = ND-sector containing the target \( p_g \). 
\( s_{max} \) = Minimal width of a wide region in number of ND-sectors. 
\( s_{rd} \) = ND-sector with rising discontinuity, i.e. border of the free walking area containing an obstacle that violates the safety-zone. 
\( s_{od} \) = ND-sector containing the other border of the free walking area. 
\( D_s \) = Safety margin between wheelchair’s bounding-polygon and safety-zone. 
\( D_{od} \) = Distance to obstacle in ND-sector \( s_{od} \) that triggered LS1/LS2-situation.

This table I is for the basic nearness diagram control laws and sheer out extensions.

\( s_{ml} \) = ND-sector containing next obstacle left/right of \( s_{rd} \). 
\( s_{closergap} \) = ND-sector containing the closer border of the free walking area, i.e. \( s_{rd} \) or \( s_{od} \). 
\( s_{safedriveby} \) = ND-sector maintaining the minimal tolerable distance to the next obstacle while describing the maximal sheer out direction. 
\( n \) = Total number of ND-sectors. 
\( \theta \) = Direction of movement (\( \theta = 2\pi + \frac{2\pi}{n} \)).

VII. EXPERIMENTAL EVALUATION

A. Subjects and Procedure

Nine healthy subjects, between 19 and 38 years old (mean = 26.67), participated in this study. The subjects had different levels of experience in BCI studies. Some of them took already part in earlier BCI studies, while some of them were naive. But none of the subjects had experience in controlling the wheelchair Rolland with or without the BremenBCI. All subjects completed a consent form and a screening questionnaire and would have been rejected if they ever had a seizure, epilepsy, mental or physical disorders or skin contact allergies. Table II shows the subjects in terms of age, gender, vision, and experience level in BCI studies.

The SSVEP response in the subject’s brain activity was non-invasively acquired with an EEG-cap. The electrodes for data acquisition were placed at sites \( P_{O3}, PO_{4}, O_{9}, O_{10}, O_{2} \), and \( P_{z} \) using the extended 10-20 system of electrode placement [32] with \( C_z \) as reference and \( AF_z \) as ground. The data was sampled with a sampling frequency \( F_s = 128Hz \).

After placing the electrodes, the subject obtained the task to randomly focus attention on each stimulus frequency of the LED panel described in Section II. During this initial task, the threshold for classification of the desired frequency was set individually for each subject. The wheelchair was not moving during the calibration step. Afterwards, a short training phase started. The subject had up to 10 minutes to train the stimulus frequencies and their corresponding commands to navigate the wheelchair, and to become comfort-
able with the behavior of the wheelchair when a command was classified. After training, the subject had to navigate the wheelchair Rolland through a course of obstacles. The navigation was expected to be performed in a figure of eight as shown in Fig. 5(a) - 5(b). Starting at position \( S = (0, 0) \), the subject had to navigate the wheelchair clockwise around the right obstacle first and then counter-clockwise around the left obstacle. The run ended when the wheelchair reached a point nearby the start position, after navigating through the entire course. Each session consists of up to four runs (depending on the subject).

B. Experimental Setup

In order to compare and evaluate the results of the different experimental runs, a common frame of reference is needed. In our case, this is a global map of the environment, and the ability to localize the wheelchair at any point in time within the map. For creating the map (shown in Fig. 5(a) - 5(b)), we used the GMapping implementation from OpenSLAM [33]. It is a highly efficient Rao-Blackwellized particle filter that learns grid maps from raw laser range and odometry data [34]. The grid map used for our experiments has a cell size of \( 2 \times 2 \text{cm}^2 \) and a total size \( 14 \times 10 \text{m}^2 \). Real-time localization during the experimental runs has been realized by a modified version of the GMapping implementation. While leaving the map data untouched, a small number of particles keep track

---

(a) Nearness diagram representation of the environment (cf. Section VI) describing the distances to the closest obstacles.

(b) Evidence grid representation of the environment (cf. Section IV) along with a centered and upward facing bounding-polygon of the wheelchair. The target \( p_{tg} \) is depicted as a circle.

Fig. 4. Illustration of Nearness Diagram Navigation within a simulated environment. \( S_{od} \) and \( S_{rd} \) denote the left and right border of the free walking area (cf. Section VI), while \( S_{goal} \) and \( \theta \) describe the direction to the local target \( p_{tg} \) and the current orientation of the wheelchair respectively.

C. Results

Table III shows the results of the experiment for all nine subjects. The classification accuracy of the BCI during the training phase is given as well as the average number of commands and the average classification accuracy during the runs. Also, the best time for navigating Rolland through the course is provided. Eight of nine subjects were able to navigate the wheelchair in a figure of eight.

The results were achieved using a segment length of \( 2 \text{s} \) of acquired EEG-data to extract the SSVEP features and to classify the desired frequency. The feature extraction and classification step of the BCI was done every \( 0.1 \text{s} \). To avoid that the same frequency is classified over and over again, because of the short time between two classifications, an idle period of \( 2 \text{s} \) is considered during the signal processing of the EEG-data, i.e., after classification of a frequency, the earliest next classification of a frequency can be done after the idle period. Thus the subject got enough time to stop focusing attention to that stimulus. For calculation of the classification accuracy, only appropriate commands to fulfill the given task were considered as true positive classifications. To minimize the chance of false positive classifications, the behavior of the power in the corresponding frequency was observed in a range of \( 0.5 \text{s} \) before and \( 0.5 \text{s} \) after the classification of a command. If the power showed a significant behavior (i.e. continuous increase and decrease) over that period, that

---

<table>
<thead>
<tr>
<th>TABLE II</th>
<th>SUBJECT INFORMATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>Gender</td>
</tr>
<tr>
<td>S1</td>
<td>female</td>
</tr>
<tr>
<td>S2</td>
<td>male</td>
</tr>
<tr>
<td>S3</td>
<td>female</td>
</tr>
<tr>
<td>S4</td>
<td>female</td>
</tr>
<tr>
<td>S5</td>
<td>male</td>
</tr>
<tr>
<td>S6</td>
<td>male</td>
</tr>
<tr>
<td>S7</td>
<td>male</td>
</tr>
<tr>
<td>S8</td>
<td>male</td>
</tr>
<tr>
<td>S9</td>
<td>female</td>
</tr>
</tbody>
</table>

Experience level:
0 = Naive in BCI studies
1 = Participated in a few earlier BCI studies (up to 3)
2 = Participated in several earlier BCI studies (more than 3)

<table>
<thead>
<tr>
<th>TABLE III</th>
<th>BCI ACCURACY DURING TRAINING PHASE AND RESULTS OF RUN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>Training Accur. [%]</td>
</tr>
<tr>
<td>S1</td>
<td>100.00</td>
</tr>
<tr>
<td>S2</td>
<td>96.30</td>
</tr>
<tr>
<td>S3</td>
<td>100.00</td>
</tr>
<tr>
<td>S4</td>
<td>94.45</td>
</tr>
<tr>
<td>S5</td>
<td>98.81</td>
</tr>
<tr>
<td>S6</td>
<td>15.18</td>
</tr>
<tr>
<td>S7</td>
<td>93.61</td>
</tr>
<tr>
<td>S8</td>
<td>100.00</td>
</tr>
<tr>
<td>S9</td>
<td>95.46</td>
</tr>
</tbody>
</table>

of the wheelchair’s position.
command is considered as a true positive classification.

D. Discussion & Conclusion

During the training phase, eight of nine subjects showed good classification results of the desired frequency. The minimum classification accuracy was 93.61% (subject S7) during wheelchair movement. It seems that a stressful situation (subject is sitting in a moving wheelchair) does not affect the SSVEP response of that subject significantly. The range of the classification accuracy during the navigation task is quite similar to that during the training phase. For eight subjects, classification errors were mainly characterized as too many classifications of the same frequency the subject is attending to. An explanation is that some subjects showed a very high SSVEP response for the stimulus frequencies. If the signal processing analyzes the recent 2s of the acquired data every 0.1s, the power calculated is even quite high for some classification steps after the subject stopped focusing attention to the stimulus, because of the high SSVEP response. In future experiments, this can be avoided by using individual idle periods for subjects with high power in the stimulus frequencies over a long time, or by using shorter segment lengths (but this typically affects the classification accuracy more significantly). Only for a single subject (subject S6), the BCI was not able to classify the desired frequency accurately. But this corresponds to the percentage of BCI illiteracy reported in [35]. Another important issue is that all subjects, for whom an SSVEP response could be obtained accurately, were able to navigate the wheelchair quite well after a minimal training phase (not more than 10 minutes) independent of their level of experience in BCI experiments. Those subjects were even able to correct wrong interpretations of a given command.

From the perspective of navigation, a major issue is the desired correspondence between the calculated, and the user-intended target position. Although general results indicate good performance of the algorithms involved, there exist situations in which the approach selected returns inadequate solutions. Considering the estimated positions of some commands given at locations that are marked by a solid circle in Fig. 5(a) - 5(b), it becomes obvious that the interpretation of the corresponding commands is not intuitive here. This behavior is due to the leading path-selection mechanism described in Section V that favors paths the final nodes of which fit closely to the commanded direction. When the Voronoi graph now includes paths that lead to currently unobservable but still mapped regions, targets may be chosen that do not correspond to the user’s anticipation. With the path-selection scheme described in second place in Section V, the described problems vanish. The critical drawback that leaves this approach to work only under a perfect world assumption is that it is highly prone to topological changes in the Voronoi graph, as induced by sensor noise.

A second situation in which the navigational approach applied did not answer the expectations of the volunteers is exemplarily highlighted in Fig. 5(b) by a dashed circle around the place of a given BCI-command. The depicted case is designated by a target position that is located at the border to a region that was unobserved at the time of computation. While approaching the selected target, new sensor measurements revealed it to be located completely on the border to an occupied region. A following stop maneuver that was initiated by the downstream NDN module irritated the participant of the study, i.e. he/she had to give a new command shortly after the aborted one. Although this behavior does not meet everyday usability requirements, our current approach of canceling a target request by the NDN module, because of an appearing obstacle that blocks the goal location, seems justified. As a reason for this we refer
to the reduced complexity of interactions between the path selecting and executing modules.

VIII. ACKNOWLEDGMENTS

The authors would like to thank B. Gersdorf for the technical setup of the smart wheelchair platform Rolland. Further acknowledgements go to U. Frese for his support with the implementation and extension of the NDN-approach.

REFERENCES


